Programa Oficial de Doctorado en Tecnologías de la Información y la Comunicación

Departamento de Teoría de la Señal, Telemática y Comunicaciones

Universidad de Granada

TESIS DOCTORAL

Supervivencia en redes ad hoc. Mecanismos de tolerancia y reacción frente amenazas de seguridad

Realizada por:

D. Roberto Magán Carrión

Dirigida por:

Prof. Dr. D. Pedro García Teodoro

Dr. D. José Camacho Páez

Editor: Universidad de Granada. Tesis Doctorales
Autor: Roberto Magán Carrión
ISBN: 978-84-9125-913-8
URI: http://hdl.handle.net/10481/43857

Official Ph.D Program in Information and Communication Technologies

Department of Signal Theory, Telematics and Communications

University of Granada

Ph.D. THESIS

Survivability of ad hoc networks. Tolerance & response mechanisms against security threats

Author:

Mr. Roberto Magán Carrión

Advisors:

Prof. Dr. Pedro García Teodoro

Dr. José Camacho Páez

El doctorando D. Roberto Magán Carrión y los directores de la tesis Dr. D. Pedro García Teodoro y Dr. D. José Camacho Páez, 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 18 de marzo de 2016

Directores de la Tesis

Prof. Dr. D. Pedro García Teodoro

Dr. D. José Camacho Páez

Doctorando

D. Roberto Magán Carrión

El presente trabajo de tesis ha sido financiado fundamentalmente por el programa FPU 6A del plan propio de la Universidad de Granada. Adicionalmente, parte del mismo ha sido sufragado a través del proyecto SuMA (TEC2011-22579) del MCINN (Ministerio de Ciencia e Innovación) (actual MINECO (Ministerio de Economía y Competitividad)), Gobierno de España.

Agradecimientos

Sin duda, estos cuatro últimos años de mi vida han dado para mucho. Si tuviera que destacar algo por encima de todo, si me plantease de nuevo empezar este trabajo, sería por la oportunidad que se me ofreció de conocer a tanta gente espléndida. Uno podría preguntarse, ¿pero qué hay de tu trabajo, tu investigación, tu futuro sin las personas a las que quieres? Creedme cuando os digo que todo eso no tiene sentido sin ellas. Así, cuando os propongáis a pasar otro fin de semana más trabajando, parad, pensad y ¡marchad a disfrutar con vuestra gente!

Aunque suene a tópico repetitivo interpuesto dentro de un párrafo sin más sentido que el de escribirlo, es cierto que sois muchos a los que he de dar las gracias. Cada uno de vosotros habéis aportado algo para la consecución de este trabajo. Empezando por mis padres, Eleazar y Cecilia, ya que sin ellos nada de esto estaría pasando. Siguiendo por mis hermanos Carlos, Esperanza y Beatriz, cada uno particularmente especial y a los que siempre añoro. Mis entrañables sobrinos Cristina, Alma, Izan y Marco, recién nacido. A mis suegros y cuñados, sin duda únicos. A la "pichuchada" al completo, por esas reuniones inolvidables. A Alberto, ¿por qué?, ¡porque eres grande, primo! A aquellos que pasean orgullosos el apellido Magán.

Qué increíblemente bueno es tener amigos. Agradecer a mis amigos de toda la vida Luis, Andrés, Romano y Fernando, los miles de momentos, chorradas y barbaridades que hemos hecho juntos. A José Fernando, porque su corazón no tiene parangón. Todavía recuerdo nuestro primer "paseillo" por Linares. A Rafita, porque es una de las personas con las que más he aprendido y de las que más quiero aprender. A Leo, porque es un toca narices y se hace querer. Al "Pableras" por ser el mejor del barrio del Realejo. A "Alextoni" *dubsmash maker*. A Irene por su constancia y amabilidad. A Juanga, per ser valencià. A "Juanjer" el magnífico. A "Gabri" por ser poco menos que el primo del creador de Matrix. A Marta porque me empieza a caer bien. A "Antoñete" porque es mi amigo. Many thanks to Gianni and Edu for their kind support in Switzerland. A todos, ¡GRACIAS!

–*Esta es TU tesis Roberto.* – me dicen Pedro y Pepe. *Lo siento*, *pero no es solo mi tesis*, *es nuestra tesis*, afirmo yo. Hay cosas que no tengo claras (me pasa mucho). Sin embargo hay otras que sí que las tengo. Una de ellas es que este trabajo de tesis no hubiese visto la luz sin la inapreciable ayuda de Pedro y Pepe. Sois geniales, de verdad. Tanto lo sois que a veces dais hasta miedo. ¡GRACIAS!

Hubo bastantes momentos en los que el lado oscuro de la tesis hizo acto de presencia. Por su paciencia, amabilidad, comprensión, apoyo y cariño desinteresado, no quiero terminar sin antes dar las gracias a una de las personas más especiales de mi vida. Ella es Vanesa, mi mujer. GRACIAS por todo. Parte de este trabajo es tuyo. ¡Te quiero mucho "lupilla"! ... Tu "lupillo".

A mis padres.

A Vanesa.

Resumen

Tecnológicamente hablando, nos encontramos dentro de un escenario de cambios continuos y acelerados. La constante innovación y apuesta por nuevas tecnologías ha llevado a la interconexión de gran cantidad de dispositivos de diversa naturaleza y propósito. Personas y "cosas" (*things*) están hoy casi inevitablemente unidos. Teléfonos móviles, tabletas, artilugios portátiles para llevar puestos (*wearable devices*), automóviles, etc., se conectan e interactúan entre sí, cambiando en cierto modo los hábitos de las personas gracias a la aparición de nuevos servicios y aplicaciones relacionados. Esta variopinta amalgama de dispositivos y servicios ofertados demanda la necesaria infraestructura de comunicación que los soporte. En este marco, las redes ad hoc son especialmente idóneas para tal fin. Dependiendo del contexto de uso nos podemos encontrar con redes ad hoc móviles (MANET (*Mobile Ad hoc NETwork*)), redes de sensores (WSN (*Wireless Sensor Network*)) o redes vehiculares (VANET (*Vehicular Ad hoc NETwork*)) o incluso aquellas en las que los artefactos que hacen las veces de nodos, poseen la capacidad de volar (FANET (*Flying Ad hoc NETwork*)).

Desde este prisma, podríamos decir que las redes ad hoc son poco menos que la panacea de las comunicaciones, interconectadas a su vez, mediante Internet. Sin embargo, desde el punto de vista de la seguridad, son también sus especiales e inherentes características las que las hacen particularmente vulnerables. Así, adolecen de vulnerabilidades en el canal de comunicaciones, dada su naturaleza abierta; vulnerabilidades en los propios nodos, debidas a las propias restricciones y limitaciones de estos; o vulnerabilidades ante actuaciones maliciosas que intenten sacar partido de la falta de infraestructura y gestión centralizada para interrumpir el proceso normal de operación de la red; entre otras vulnerabilidades.

Con objeto de avanzar hacia el desarrollo de redes más resistentes y robustas frente a las amenazas y ataques a la seguridad, es usual la propuesta de soluciones basadas en la prevención (resistencia), la detección (reconocimiento) o la reacción/-tolerancia (recuperación). Una vez que las medidas preventivas son eludidas, son los sistemas de detección quienes han de alertar de la presencia de un ataque para que, por último, se tomen las medidas adecuadas cuyo fin es contrarrestar los efectos producidos por el mismo. Este tipo de soluciones abogan por la continuidad de la red y de los servicios que ofrece ante la presencia de amenazas o ataques, donde aspectos como la eficiencia energética, capacidad de adaptación o auto-gestión han de ser tenidos en cuenta para contribuir al mismo fin. En su conjunto, todas estas cualidades o características forman parte de un objetivo más ambicioso como es añadir la capacidad de supervivencia a la red o sistema ante ataques, fallos o accidentes.

Motivado por las razones anteriores y ante la notable falta de soluciones que aboguen por la adicción de capacidad de supervivencia al sistema o red en donde se

aplican, el objetivo principal del presente trabajo de tesis es el desarrollo y puesta en marcha de *esquemas de respuesta y tolerancia* que contribuyan a la continuidad de la red y de los servicios que esta ofrece, más allá de la actuación específica sobre determinados ataques. A su vez, y no menos importante, es objetivo también construir sistemas de seguridad que habiliten *la integración e interacción de diferentes líneas de defensa*, como parte de un esquema completo frente amenazas de seguridad. En este sentido, llevaremos a cabo el diseño de los mecanismos necesarios para el despliegue de soluciones integrales de seguridad, como pilar fundamental hacia la consecución de sistemas y redes con capacidad de supervivencia.

Abstract

Technology is continuously evolving, so that a huge amount of heterogeneous devices are connected with different purposes. People and things are necessarily in touch. Smartphones, tablets, wearable devices and vehicles interact with each other by means of diverse communication means to offer new services and applications that change people's daily life. To support such a variety of services and the underlying communications, network environments like ad hoc networks are especially recommended. Depending on the context of use we can find several types of ad hoc networks: mobile ad hoc networks (MANETs), sensor networks (WSNs), vehicular ad hoc networks (VANETs) or those where nodes have flying capabilities (FANETs).

We could think that such a kind of networks are the panacea of communications. However, their unique and inherent characteristics constitute at the same time their main weaknesses from the point of view of security. In fact, they suffer from channel vulnerabilities, due to the usual wireless communications involved; node vulnerabilities, due to usual constraints and limitations on their nodes; or malicious acts that take advantage of the absence of infrastructure for these networks in order to disrupt the normal network operation; among several other vulnerabilities.

In this context, a number of proposals can be found in the literature to get more robust and resilient networks against security threats or attacks. They all belong to one among three traditional and collaborative defense lines: prevention (resistance), detection (recognition) or response/tolerant (recovery). This way, once the preventive line is eluded by a potential attack, the detection module is in charge of determining the presence of the attack and triggering an alarm. After that, the response/tolerant module will operate to mitigate the attack effects. Through such in-depth defense lines we try to protect and maintain the network and services offered over time. Aspects like energy efficiency, adaptability or self-management are of main relevance to achieve that overall aim. All together support a more ambitious goal: addition of survivability capabilities in the presence of attacks, faults or accidents in the network environment.

Mainly due to the notable lack of current solutions intended to provide survivability capabilities, the principal objective of the present thesis is *to design and implement new response and tolerant schemes*. These proposals are aimed at maintaining the network and services offered over time in a wider sense than that of merely fighting against specific attacks. Moreover, we also develop security proposals aimed at enabling *the integration and interaction of different defense lines*, as part of a unique and global multi-line defense scheme. For that, we carry out the design of the necessary mechanisms to deploy integral security solutions as an essential tool for obtaining survivable systems and networks.

Contenido

Lista de Figuras

Lista de Tablas

Lista de Abreviaturas y Acrónimos

2CRNDC	*2-Connected Relay Node Double Cover*
AA	*Action Agent*
ADR	*Attack on Direct Routing*
ADRR	*Attack on DRR*
ALR	*Attack on LEACH Routing*
AMR	*Attack on MCFA Routing*
AODV	*Ad hoc On-demand Distance Vector*
AP	*Attraction Point*
AR	*Attraction Rate*
ARAN	*Authenticated Routing for Ad hoc Networks*
ARR	*Attack on RR*
ASR	*Atacck on SR*
ATT	*ATTack test dataset*
BA	*Bat-inspired Algorithm*
CAL	*CALibration dataset*
CBR	*Constant Bit Rate*
CH	*Cluster Head*
CU	*Central Unit*

CN	*Central Node*
CONFIDANT	*Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks*
CRADS	*Cross-layer Routing Attack Detection System*
CREP	*Confirmation REPly*
CREQ	*Confirmation REQuest*
CRNSC	*Connected Relay Node Single Cover*
CSP	*Constraint Satisfaction Problem*
CTS	*Clear To Send*
DCRNPP	*Delay Constrained Relay Node Placement Problem*
DoS	*Denial of Service*
DDoS	*Distributed DoS*
DPCA	*Dynamic PCA*
DPRAODV	*Detection, Prevention and Reactive AODV*
DR	*Dropping Ratio*
DRNS	*Dynamical Relay Node placement Solution*
DRR	*Differential Random Routing*
DSR	*Dynamic Source Routing*
DTMS	*Distributed Trust Management System*
DTN	*Delay Tolerant Network*
dynRNP	*dynamic Router Node Placement*
E2ED	*End-to-End Delay*
EFSA	*Extended Finite State Automaton*
EPFL	*École Polytechnique Fédérale de Lausanne*
E-SRPM	*Enhanced SRPM*
FANET	*Flying Ad hoc NETwork*
FAP	*Flying Aerial Platform*

FDA	*Fisher Discriminant Analysis*
FIR	*FIRe dataset*
GPRS	*General Packet Radio Service*
HMM	*Hidden Markov Model*
IA	*Immune Agent*
IDAD	*Intrusion Detection based on Anomaly Detection*
IDS	*Intrusion Detection System*
IDSIA	*Institute Dalle Molle for Artificial Intelligence*
IEEE	*Institute of Electrical and Electronics Engineers*
ILP	*Integer Linear Programming*
INET	*INET Framework*
IP	*Internet Protocol*
IPS	*Intrusion Prevention System*
IRS	*Intrusion Response System*
LCM	*Lightweight Communications and Marchalling*
LDMR	*Least Distance Movement Recovery*
LEACH	*Low Energy Adaptive Clustering Hierarchy*
LOO	*Leave-One-Out*
LUCE	*Laussanne Urban Canopy Experiment*
LV	*Latent Variable*
MAC	*Medium Access Control*
MANET	*Mobile Ad hoc NETwork*
MCFA	*Minimum Cost Forwarding Algorithm*
MOA	*MOnitoring Agent*
MPC	*Model Predictive Control*
MSE	*Mean Squared Error*

MSPC	*Multivariate Statistical Process Control*
MST	*Minimum Spanning Tree*
NED	*Network Description*
NeSSi	*Network Security Simulator*
NETA	*NETwork Attacks*
NS-2	*Network Simulator 2*
NFJ	*Null Frequency Jamming*
OMH	*One More Hop*
PC	*Principal Components*
PCA	*Principal Component Analysis*
PDR	*Packet Delivery Ratio*
PLS	*Partial Least Squares*
PRESS	*Prediction Error Sum of Squares*
PSO	*Particle Swarm Optimization*
OMNeT++	*Objective Modular Network Test-bed in C++*
OPNET	*Optimized Network Engineering Tools*
QoS	*Quality of Service*
RIM	*Recovery through Inward Motion*
RN	*Relay Node*
ROA	*ROuting Agent*
RPGM	*Reference Point Group Mobility*
RR	*Random Routing*
RREP	*Route REPly*
RREQ	*Route REQuest*
RTS	*Request To Send*
RWP	*Random Way Point*

SA	*Simulated Annealing*
SAODV	*Secure AODV*
SORI	*Secure and Objective Reputation-based Incentive*
SR	*Switching-based Routing*
SRPM	*Secure Routing Protocol for wireless Mesh*
SRREP	*Secure RREP*
SRREQ	*Secure RREQ*
SVM	*Support Vector Machine*
TESRP	*Trust and Energy aware Secure Routing Protocol*
TCP	*Transfer Control Protocol*
TM	*Trust Management*
TFT	*Tit-For-Tat*
TSR	*Trimmed Scores Regression*
UAV	*Unmaned Aerial Vehicles*
UDP	*User Datagram Protocol*
URL	*Uniform Resource Locator*
UN	*User Node*
VANET	*Vehicular Ad hoc NETwork*
WMN	*Wireless Mesh Network*
WSN	*Wireless Sensor Network*

Capítulo 1

Introducción

Contenido

Hoy en día, principalmente gracias a la aparición de nuevas tecnologías, prácticamente cualquier dispositivo electrónico existente está dotado con capacidades de comunicación. Personas y "cosas" (*things*) permanecen en contacto a través del intercambio de información sobre medios y redes heterogéneas. El variopinto e incesante crecimiento en el número de servicios y aplicaciones ofertadas hace que sea necesaria la provisión de las adecuadas vías de comunicación e infraestructura de red que las soporte. En este marco, son las redes ad hoc[1] y sus especiales características las que las hacen idóneas para cubrir la mayoría de las necesidades actuales de comunicación [2].

Las redes ad hoc se caracterizan principalmente por no disponer de una topología o infraestructura fija, de manera que cada nodo se ubica de acuerdo a objetivos específicos. Adicionalmente, su topología flexible y descentralizada implica la necesaria habilidad de los nodos para comunicarse de modo cooperativo formando comunicaciones multi-salto o *multi-hop* [3]. Estrictamente hablando, el término "ad hoc" no lleva asociado implícitamente el uso de conexiones inalámbricas, aunque este tipo de enlaces mejoran sin duda la escalabilidad y versatilidad de este tipo de redes. Estas son especialmente recomendadas en aquellos escenarios en los que las

[1] El término "ad hoc" proviene del latín y literalmente significa "para esto" [1].

redes convencionales basadas en una infraestructura fija no están disponibles o no es viable su utilización [4].

Dependiendo del contexto de uso o de la adopción de características especiales, existen varios tipos de redes ad hoc. Por ejemplo, los nodos de una red MANET (*Mobile Ad hoc NETwork*) son capaces de moverse por el área en donde se despliegan. Esta cualidad hace que su uso esté especialmente recomendado en escenarios tales como operaciones de rescate o recuperación de comunicaciones ante desastres naturales. Otra clase especial de redes ad hoc son las WSN (*Wireless Sensor Network*), cuyo principal objetivo es la monitorización y recogida de información de una determinada área. La información obtenida por los nodos (sensores en este caso) puede ser usada después para la detección y respuesta ante determinados eventos o anomalías. Un ejemplo de uso es la detección y lucha contra incendios en zonas forestales. Las DTN (*Delay Tolerant Network*), también llamadas *oportunistic networks*, son otro tipo de redes ad hoc. Estas se despliegan normalmente en entornos que varían de forma rápida y continua, siendo su principal característica la tolerancia a los retardos en las comunicaciones. En escenarios similares, las VANET (*Vehicular Ad hoc NETwork*) o las FANET (*Flying Ad hoc NETwork*) se presentan como otros tipos especiales de redes ad hoc cuyo fin principal es el control de carreteras, tráfico y vehículos, o de dispositivos con capacidad de vuelo (UAV (*Unmaned Aerial Vehicles*) o *drones*), respectivamente. En su caso, las WMN (*Wireless Mesh Network*) se diseñan principalmente para servir de redes intermedias en la provisión de acceso a Internet a otras redes. En la Figura 1.1 se ilustran gráficamente algunos de los tipos de redes ad hoc anteriores, así como un hipotético pero factible esquema de interconexión y comunicación entre ellas.

A pesar de las grandes posibilidades que ofrecen las redes ad hoc, estas adolecen de inconvenientes relevantes en lo que respecta a la seguridad. Entre otros, sufren de vulnerabilidades en el canal de comunicaciones, debido a la naturaleza abierta de los enlaces inalámbricos que hacen posible, por ejemplo, ataques de escuchas no autorizadas o *eavesdropping*; vulnerabilidades en los propios nodos, ya que es posible que alguien con malas intenciones pudiera acceder al nodo para dañarlo o modificar la información que este gestiona, comprometiendo así la integridad de la información transmitida; o vulnerabilidades ante actuaciones maliciosas producidas por ataques como el de *dropping* o *sinkhole*, o por comportamientos egoístas (*selfish*) [5][6], que intenten sacar partido de la falta de infraestructura y gestión centralizada para interrumpir el proceso normal de operación de la red. Por las anteriores razones, es muy importante añadir sistemas o esquemas adicionales para la provisión de servicios esenciales de seguridad, garantizando así la disponibilidad, integridad, confidencialidad, privacidad, autenticación y no repudio en las comunicaciones [7]. En este sentido, es habitual el uso de tres líneas de defensa. Estas son: la prevención, la detección y la respuesta/tolerancia tradicionalmente implementadas a través sistemas IPS (*Intrusion Prevention System*), IDS (*Intrusion Detection System*) e IRS (*Intrusion Response System*), respectivamente [8]. Aunque existe una gran variedad de

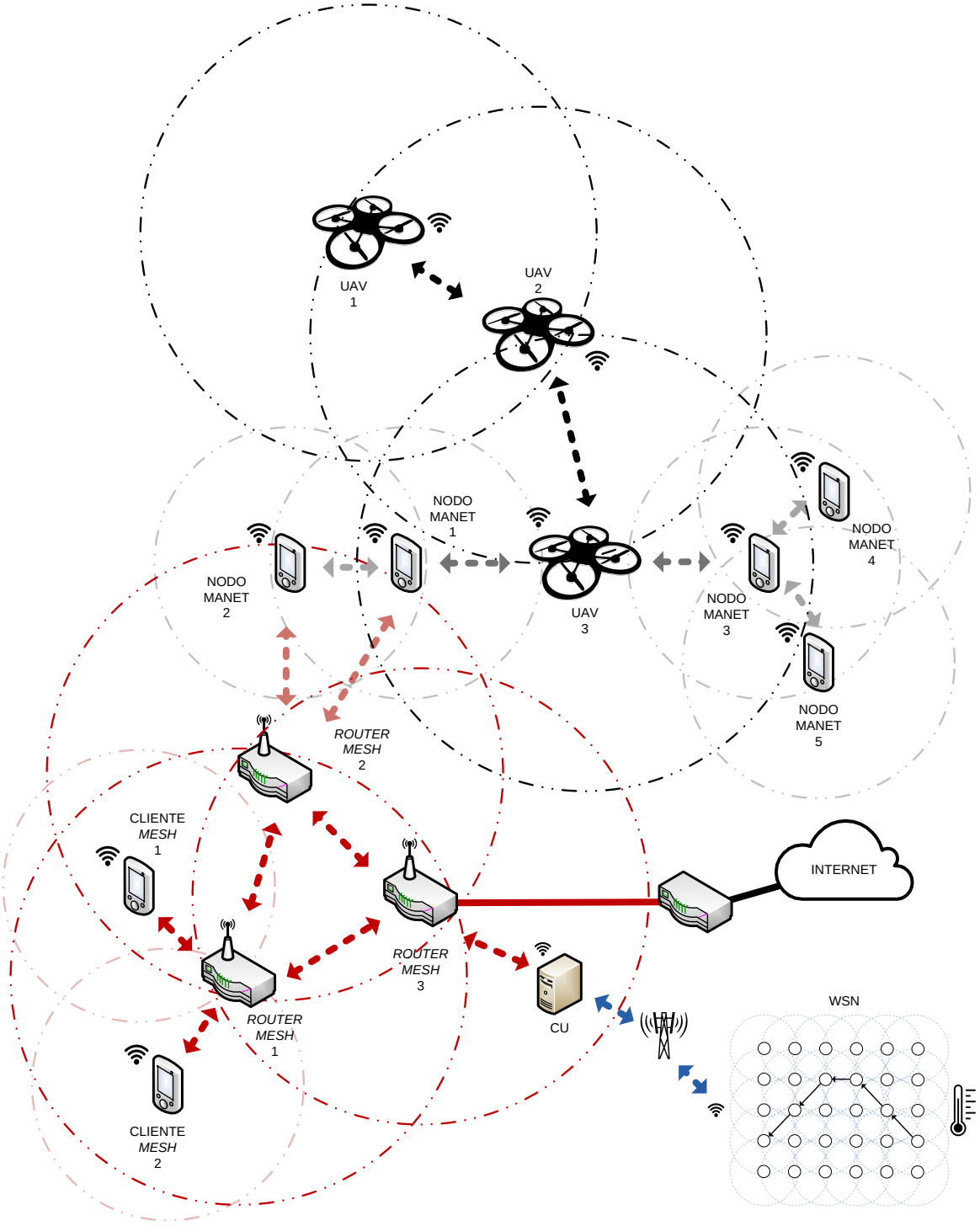


Figura 1.1: Diferentes tipos de redes ad hoc y sus posibilidades de interconexión.

propuestas en la literatura que abordan aspectos de seguridad en redes ad hoc [9], la mayoría de ellas se centra en soluciones preventivas principalmente basadas en el uso de técnicas criptográficas. En ocasiones, sin embargo, este tipo de soluciones son eludidas especialmente cuando nos enfrentamos a atacantes que forman parte de la red (ataques internos). Es en este momento cuando los esquemas de detección deberían actuar para determinar la presencia de actuaciones maliciosas dentro de la red. Por sí mismos, los procedimientos de detección no tratan de mitigar el ataque, de modo que, una vez detectado este, es necesario llevar a cabo la subsecuente respuesta que suavice o erradique el efecto producido por el ataque sobre los servicios y el rendimiento de la red.

Las anteriores líneas de defensa hacen que la red y los servicios que esta ofrece sean más robustos ante amenazas o ataques a la seguridad. De manera más general, podríamos aventurarnos a decir que no solo es necesaria la aportación de propuestas seguras, sino también una apuesta por sistemas que añadan capacidades adicionales como la eficiencia energética, la adaptabilidad al entorno, la auto-gestión, etc. En suma, que en su conjunto aboguen por la supervivencia de la red. Pero, ¿qué es la supervivencia? Este paradigma está presente en muchos aspectos de la naturaleza. Por ejemplo, en el mundo animal, cuando solo aquellos especímenes más fuertes, capaces de combatir diferentes tipos de amenazas y adaptarse a cambios en su entorno, son los que perduran. En el ámbito que nos concierne, un sistema con capacidad de supervivencia es aquel que posee "*la habilidad para cumplir con sus objetivos (ofrecer sus servicios) en el tiempo y ante la presencia de ataques, fallos o accidentes [10]*". Dicho así, el término supervivencia se presenta como algo difícil de medir o cuantificar, siendo más bien un concepto subjetivo y meramente cualitativo de cómo un sistema es capaz de cumplir con una serie de propiedades y requisitos relacionados con la supervivencia [11]. Sin embargo, sí podemos concluir y en cierto modo garantizar que la obtención de esta cualidad pasa por el diseño de sistemas o metodologías que necesariamente cumplan con unos requerimientos básicos, como son: (*i*) la habilidad de resistir (prevenir), (*ii*) de reconocer (detectar), (*iii*) de recuperarse (reaccionar) y (*iv*) de adaptarse (tolerar) ante fallos, ataques o accidentes que dañen su capacidad para seguir ofreciendo las funciones o servicios para lo cuales se diseñaron [10]. A través de la Figura 1.2 se observa la necesaria interacción entre estas habilidades, así como su estrecha relación con las principales líneas de defensa de seguridad.

Es notable la falta de propuestas encaminadas a conseguir sistemas con capacidad de supervivencia donde se consideren los aspectos anteriores de forma global [10]. Ello está en parte motivado por la también notable dificultad implícita que el desarrollo de este tipo de sistemas conlleva, ya que estaríamos hablando de soluciones capaces de evitar o contrarrestar casi cualquier ataque, fallo o accidente que se produzca. Quizá debido a esta generalidad, muchas de las propuestas que se encuentran en la literatura especializada se centran en amenazas o ataques específicos. Es más, normalmente solo consideran una línea de defensa y/o se enfocan en una

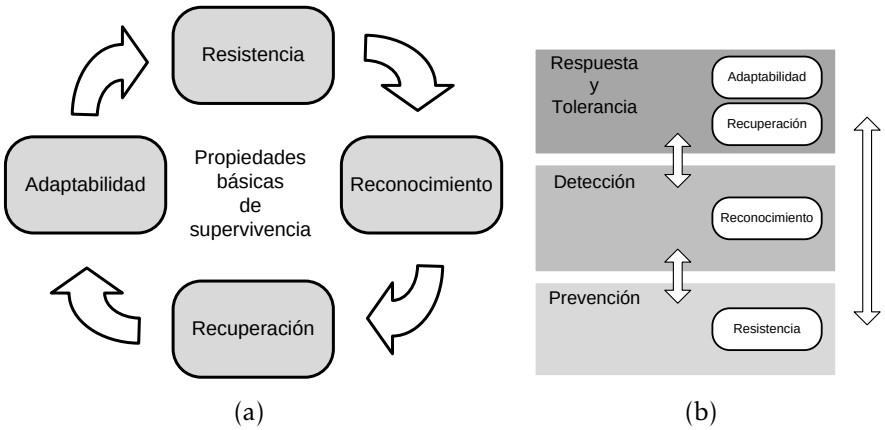


Figura 1.2: Propiedades básicas de supervivencia en redes ad hoc (a), y su relación existente con las principales líneas de defensa (b).

determinada capa del modelo de red [12]. Este tipo de soluciones son especialmente efectivas para el ataque para el cual han sido diseñadas, pero resultan poco útiles ante cambios en el escenario de aplicación, ataque o amenaza. En la Figura 1.3 se expone el ámbito de actuación común de las soluciones de seguridad propuestas por parte de la comunidad investigadora, así como el correspondiente a aquellas soluciones orientadas a conseguir sistemas con capacidad de supervivencia. Se observa claramente que las soluciones de seguridad cubren una parte mínima dentro del marco de la supervivencia del sistema, ya que normalmente se centran en una línea de defensa específica, una única capa dentro del modelo de red y/o escasos requerimientos adicionales. En el ejemplo de la Figura 1.3 se expone un posible sistema de detección (reconocimiento) cuyo ámbito se limita a la actuación sobre una única capa de red y que engloba todos los requerimientos adicionales de supervivencia (en el mejor de los casos). Es decisión de la persona o personas que idean la solución de seguridad, el abarcar más o menos aspectos dentro de este marco. Indiscutiblemente, cuanto mayor sea el volumen ocupado por la solución dentro del sistema de coordenadas de supervivencia, más robusta, resistente y fiable será la propuesta planteada ante amenazas o ataques a la seguridad.

Dentro de este contexto, en el que existe una clara necesidad de provisión de soluciones de seguridad orientadas a la supervivencia, el objetivo principal del presente trabajo de tesis es el desarrollo y puesta en marcha de esquemas de respuesta y tolerancia que aboguen por la conservación o recuperación, en su caso, de los servicios ofrecidos por este tipo de redes. A su vez, y no menos importante, es objetivo también construir sistemas de seguridad que habiliten la integración e interacción de diferentes líneas de defensa, como parte de un esquema completo de defensa frente amenazas de seguridad. En este sentido, llevaremos a cabo el diseño de los mecanismos necesarios para el despliegue de soluciones integrales de

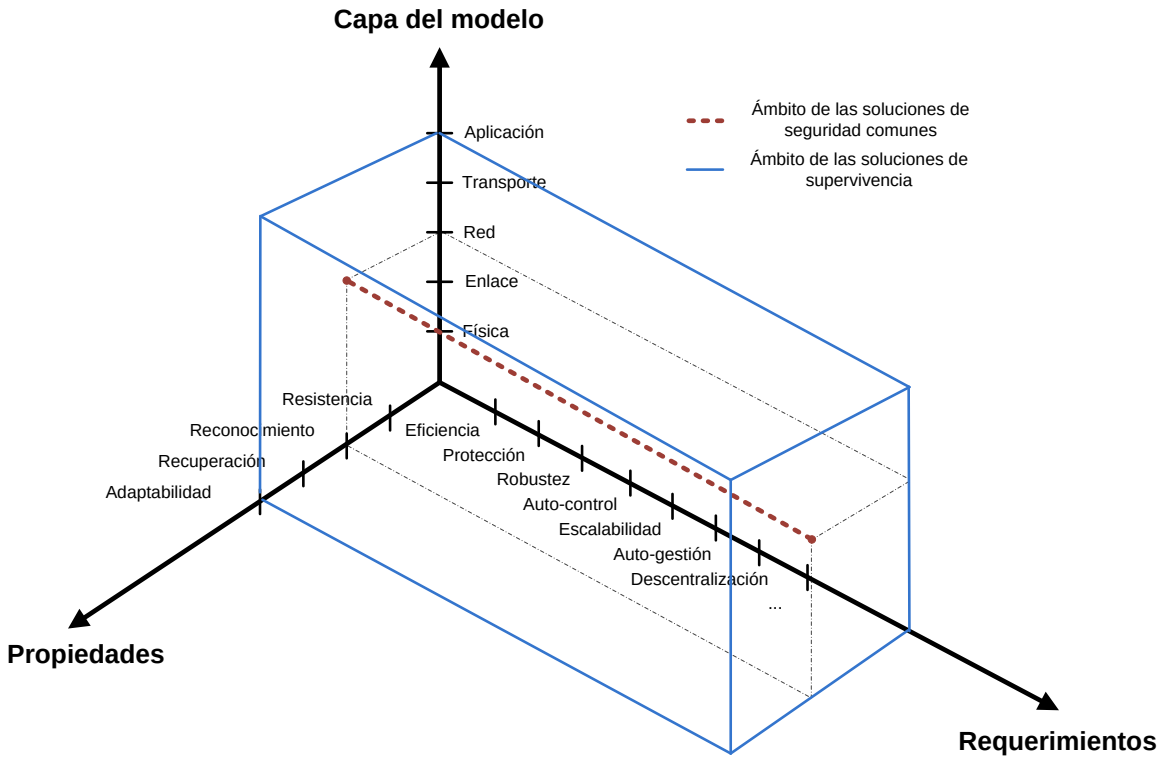


Figura 1.3: Dimensiones de supervivencia.

seguridad, como pilar fundamental hacia la consecución de sistemas con capacidad de supervivencia.

1.1. Objetivos y metodologías

Previamente comentado, el principal objetivo del presente trabajo es robustecer las redes ad hoc frente a las vulnerabilidades de seguridad de las que adolecen, mediante la adición de capacidades de supervivencia como parte del camino hacia la consecución de sistemas integrales y globales de defensa. Para ello, diseñaremos, implementaremos y desplegaremos nuevas soluciones de seguridad basadas en la reacción y la tolerancia ante la constatación de ataques. Tales soluciones añaden cualidades de supervivencia al sistema como la auto-curación o la auto-adaptación, entre otras, a través de las cuales se pretende mitigar o contrarrestar los efectos producidos por actuaciones maliciosas, fallos o accidentes.

Guiados por el anterior fin, el primer paso que daremos será la realización del estudio de la literatura especializada orientado a la búsqueda de: (*i*) las amenazas actuales y relevantes de seguridad que afectan a las redes ad hoc, y (*ii*) las soluciones

de seguridad basadas en el empleo de técnicas de reacción o tolerancia cuya meta es conformar redes con capacidad de supervivencia.

A continuación nos centraremos en el diseño y desarrollo de técnicas útiles y prácticas de reacción/tolerancia, que cubran algunas de las deficiencias encontradas en la literatura especializada y robustezcan este tipo de redes frente a algunas de las vulnerabilidades de seguridad que presentan. Debido a la variedad en el tipo de redes ad hoc, nos centraremos aquí en dos de las más utilizadas: las WSN y las MANET. Para cada una de ellas se proponen diferentes aproximaciones que se evalúan a través de extensivos experimentos tanto en entornos simulados como reales. Los resultados obtenidos muestran el buen rendimiento de las propuestas, su viabilidad y utilidad práctica.

Aunque las soluciones implementadas aquí contribuyen a reforzar la red y su capacidad de supervivencia frente algunas de sus principales debilidades, es necesaria la propuesta de soluciones más ambiciosas que apuesten por la integración global de esquemas de seguridad. Con este propósito, también se propondrá un *framework* que considera de forma conjunta la unión de soluciones relativas a diferentes líneas de defensa. Para hacerlo efectivo, se diseñará e implementará la infraestructura necesaria que permita, por un lado, la interacción y comunicación entre dichas líneas de defensa y, por otro, la implementación y ejecución de ataques que posibiliten la realización de las pruebas necesarias de operación y rendimiento del sistema desde el punto de vista de la seguridad.

En lo que sigue, exponemos las tareas que se llevarán a cabo a lo largo del trabajo:

I. *Estudio de la supervivencia y la seguridad en redes ad hoc, orientado a*

   a) Establecer las dependencias y relaciones existentes entre la seguridad y la supervivencia, mostrando la relevancia de la provisión de soluciones seguras desde este punto de vista.

   b) Estudiar las características y vulnerabilidades de seguridad en redes ad hoc.

   c) Realizar un detallado análisis de las soluciones disponibles en la literatura especializada enfocadas a la respuesta y tolerancia ante las anteriores amenazas.

II. *Propuesta de soluciones de seguridad para la respuesta y tolerancia en redes ad hoc, de manera que*

   a) Se ideen y desarrollen nuevos esquemas reactivos/tolerantes para la lucha contra las amenazas de seguridad más relevantes en redes ad hoc, con especial enfoque en escenarios de red WSN y MANET.

   b) Se evalúe el correcto comportamiento de las propuestas mediante:

- El uso de entornos simulados, que nos llevan a corroborar su viabilidad en primera instancia.
- El uso de entornos reales. Este hecho confirmará su aplicación práctica, algo que normalmente no se considera en la mayoría de los trabajos existentes en la literatura.

III. *Integración de soluciones de seguridad con objeto de conseguir redes con capacidad de supervivencia. Esto es*

   a) Idear y diseñar una arquitectura factible para la implementación y pruebas de ataques en redes ad hoc.

   b) Abordar el problema de cómo diferentes líneas de defensa deberían interaccionar entre ellas, apoyándonos en el estudio previo de las dispares soluciones encontradas en la literatura.

   c) Idear, diseñar e implementar un *framework* capaz de incorporar varias líneas de defensa, así como proveer los mecanismos necesarios para permitir su integración e interacción.

   d) Probar y evaluar el sistema completo a través de la implementación y despliegue de ciertos ataques.

1.2. Contribuciones principales

A partir de los objetivos anteriores, las contribuciones principales del presente trabajo de tesis se resumen como sigue:

1. Estudio de las principales amenazas de seguridad que afectan a las redes ad hoc, así como de las soluciones propuestas por la comunidad investigadora para su defensa, prestando especial atención a las relativas a proveer mecanismos de respuesta o tolerancia ante la presencia de ataques. Tras este estudio, se proporciona una novedosa clasificación para aquellas soluciones reactivas y tolerantes, cuyo fin es clarificar y ordenar la amalgama de soluciones encontradas al respecto.

2. Desarrollo y evaluación de dos novedosas soluciones de respuesta/tolerancia para la lucha frente amenazas de seguridad en redes ad hoc. La primera aproximación está basada en el empleo de técnicas de análisis multivariante para la imputación de datos faltantes a partir de las mediciones recibidas por los sensores de una WSN. Actuando directamente sobre esta información, somos capaces mitigar los efectos de ataques o actuaciones maliciosas cuyo origen es distinto. Por ejemplo, nos centraremos en solventar ataques a la integridad de los datos a nivel físico (*data tampering*) siendo este esquema perfectamente

aplicable a ataques que tratan de interrumpir la disponibilidad de la red y los servicios que esta ofrece, como es el caso de los ataques de *packet dropping* o comportamientos *selfish*.

3. La segunda propuesta ofrece un esquema de optimización para el posicionamiento de nodos *relay* cuyo objetivo es mantener, recuperar o incluso mejorar la conectividad y el *throughput* en MANET. Actuando sobre la maximización de métricas de rendimiento de la red (son algunos los trabajos en los que dichas métricas se utilizan como mediciones de la capacidad de supervivencia de un sistema [11]) se pretende conseguir sistemas más globales de respuesta/tolerancia que contemplen aquellas amenazas o ataques cuyo impacto en la red, directo o indirecto, afecte a dichas métricas en su deseo de actuar, principalmente, contra la disponibilidad de la red. A modo de ejemplo, se contrarrestan los efectos producidos en el rendimiento de la red por ataques *blackhole*, aunque sería perfectamente aplicable para la lucha contra ataques como el de *sinkhole* o *wormhole* o frente a comportamientos *selfish* y otros similares.

4. Desarrollo de un novedoso *framework* para la integración e interacción de soluciones de seguridad heterogéneas pertenecientes a diferentes líneas de defensa, que se soporta en una arquitectura flexible, escalable y versátil. Sin duda, la integración de esquemas de defensa, proporciona sistemas y arquitecturas que contribuyen a la adicción de capacidades de supervivencia en aquellas redes donde se despliegan.

1.2.1. Publicaciones

A continuación se indica el conjunto de publicaciones derivadas de este periodo de tesis doctoral y relacionadas con el tema principal de la presente tesis:

Revistas internacionales

1. **R. Magán-Carrión**, J. Camacho, P. García-Teodoro, E. F. Flushing y Gianni A. Di Caro. "Dynamical Relay Node placement Solution (DRNS) for MANETs," Enviado a *Ad Hoc Networks (Elsevier)*, 39 páginas, 2016.

2. J. Camacho, **R. Magán-Carrión**, P. García-Teodoro, J. J. Treinen. "Networkmetrics: Multivariate Big Data Analysis in the Context of the Internet," Enviado a *J. Chemometrics (Wiley)*, 45 páginas, febrero 2016.

3. **R. Magán-Carrión**, R.A. Rodríquez-Gómez, J. Camacho y P. García-Teodoro. "Optimal Relay Placement in Multi-hop Wireless," Aceptado en *Ad Hoc Networks (Elsevier)*, 34 páginas, marzo 2016.

4. L. Sánchez-Casado, G. Maciá-Fernández, P. García-Teodoro y **R. Magán-Carrión**. "A model of data forwarding in MANETs for lightweight detection of malicious packet dropping," *Computer Networks (Elsevier)*, vol. 87, pp. 44–58, julio 2015.

5. **R. Magán-Carrión**, J. Camacho y P. García-Teodoro. "Multivariate Statistical Approach for Anomaly Detection and Lost Data Recovery in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks (Hindawi)*, vol. 2015, pp. 1–20, mayo 2015.

6. **R. Magán-Carrión**, F. Pulido Pulido, J. Camacho Páez y P. García-Teodoro. "Tampered Data Recovery in WSNs through Dynamic PCA and Variable Routing Strategies," *Journal of Communications*, vol. 8, pp. 738–750, noviembre 2013.

Conferencias y congresos internacionales

7. **R. Magán-Carrión**, J. Camacho and P. García-Teodoro, E. F. Flushing y Gianni A. Di Caro. "DRNS: Dynamical Relay Node placement Solution," Aceptado en *Demonstration in Advances on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, junio 2016.

8. **R. Magán-Carrión**, J. Camacho, P. García-Teodoro, E. F. Flushing y Gianni A. Di Caro. "Dynamical Relay Node placement Solution in MANETs," *Demonstration in 3rd International Black Sea Conference on Communications and Networking (BlackSeaComm)*, mayo 2015. [Demo online; Accessed 15-December-2015]

9. **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Multiagent Self-healing System against Security Incidents in MANETs," *Highlights of Practical Applications of Heterogeneous Multi-Agent Systems (PAAMS)*, vol. 430, pp. 321–332, junio 2014.

10. 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, septiembre 2013.

11. **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents," *Advances on Practical Applications of Agents and Multi-Agent Systems (PAAMS)* , vol. 7879, pp. 182–191, mayo 2013.

12. **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents: A Practical Vision," *Demonstration in Advances on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, vol. 7879, pp. 308–311, mayo 2013.

Capítulos de libro

13. 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 and J. Lloret (Eds.), CRC Press, pp. 377-400, abril 2014.

Congresos y conferencias nacionales

14. 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," *Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, pp. 321-326, septiembre 2014.
15. **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents: Limitations and Improvements," *XI Jornadas de Ingeniería Telemática (JITEL)*, pp. 445–452, octubre 2013.
16. 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, octubre 2013.
17. **R. Magán-Carrión**, J. Camacho Páez y P. García-Teodoro. "Supervivencia en redes de sensores mediante técnicas multivariantes," *12th Reunión Española sobre Criptología y Seguridad de la Información (RECSI*) pp. 315–320, septiembre 2012.

Trabajos dirigidos

Así mismo, dentro del contexto de esta tesis, se han llevado a cabo los siguientes Proyectos Fin de Carrera y Trabajos Fin de Grado:

- José Moreno Molina, "Integración de soluciones de seguridad en redes MANET con NETA". *Proyecto Fin de Carrera*, P. García Teodoro y **R. Magán Carrión** (Directores), Ingeniería de Telecomunicación, Universidad de Granada, Julio 2015.
- Miguel Tabernero Ridao, "Posicionamiento óptimo de nodos retransmisores en redes WiFi con Android". *Proyecto Fin de Carrera*, J. Camacho Páez y **R. Magán Carrión** (Directores), Ingeniería de Telecomunicación, Universidad de Granada, Julio 2014.

- Francisco Jesús Cadenas Callejón, "Medidas de mejora y gestión de la conectividad en redes MANET mediante el empleo de agentes móviles". *Trabajo Fin de Grado*, J. Camacho Páez y **R. Magán Carrión** (Directores), Grado en Ingeniería de Tecnologías de la Comunicación, Universidad de Granada, Septiembre 2014.

1.3. Estructura del documento

De acuerdo a los objetivos expuestos anteriormente, el presente documento se estructura en tres partes fundamentales, todas ellas dirigidas a combatir las principales vulnerabilidades de seguridad que aparecen en las redes ad hoc. Aunque sin duda relacionados, todos y cada uno de los capítulos presentados a lo largo del trabajo han sido redactados para que, en la medida de lo posible, sean autocontenidos, de modo que se pueda realizar su lectura y compresión sin necesidad de leer el resto.

1.3.1. Parte I: Seguridad y supervivencia en redes ad hoc

Esta parte se compone íntegramente del Capítulo 2. A lo largo de este capítulo se estudiarán las principales amenazas a la seguridad de las redes ad hoc, que son fundamentalmente originadas por vulnerabilidades derivadas de sus especiales características. Estudiaremos los requisitos principales de seguridad que deberían considerarse de cara a fortalecer y robustecer este tipo de entornos frente a dichas amenazas y potenciales ataques. Finalmente, efectuaremos una búsqueda intensiva acerca de las soluciones de seguridad encontradas en la literatura, haciendo especial hincapié en aquellas propuestas basadas en esquemas reactivos y/o tolerantes ante ataques cuyo fin es la adición de capacidad de supervivencia a la red.

1.3.2. Parte II: Reacción y tolerancia ante amenazas a la seguridad en redes ad hoc

Dentro de esta parte, y principalmente motivados por las conclusiones obtenidas en el anterior estudio, se proponen y desarrollan dos soluciones de reacción/tolerancia para dos tipos de redes ad hoc en concreto. Dentro del contexto de las WSN, en el Capítulo 3 se describe una solución de imputación de datos faltantes basada en el uso de técnicas de análisis multivariante. Esta solución, como ejemplo de aplicación práctica, se enfoca en combatir ataques que atentan contra la integridad de la información que se maneja en este tipo de redes. Si bien este es un posible escenario de uso, dicha solución es también válida ante ataques como el de *blackhole*, cuya

eliminación de paquetes puede provocar pérdidas en la información recogida por un sensor.

Por su parte, a lo largo del Capítulo 4 se desarrolla una solución de respuesta/tolerancia basada en el posicionamiento optimizado de nodos *relay*. Dicha propuesta está orientada a mantener servicios esenciales en este tipo de redes como son: capacidad de comunicación, encaminamiento y conectividad. Aplicaremos este esquema para solventar los efectos producidos ante la presencia de ataques que atentan contra la disponibilidad y rendimiento en este tipo de redes, por ejemplo, los ataques de *packet dropping*. Ambas soluciones han sido implementadas y testadas en entornos simulados para probar así su viabilidad. Además, y con objeto de corroborar su aplicación práctica, dichos esquemas han sido convenientemente desplegados y probados en entornos reales.

1.3.3. Parte III: Integración de soluciones de seguridad

Con el propósito de fortalecer y robustecer este tipo de redes frente amenazas a la seguridad, agrupados en esta parte se encuentran los Capítulos 5 y 6. En el Capítulo 5 se diseña un *framework* para la implementación y pruebas de ataques en redes heterogéneas. Apoyándonos en el anterior sistema, a lo largo del Capítulo 6 se idea una arquitectura que permite la integración e interacción de diferentes esquemas de seguridad que se ubican dentro de diferentes líneas de defensa. Dicho sistema tiene como objetivo obtener soluciones globales de seguridad como parte imprescindible para la adición de capacidad de supervivencia en este tipo de redes. En conjunto, dispondremos de una herramienta integral de defensa que es capaz de simular ataques en redes ad hoc, integrar diferentes soluciones de seguridad y proveer el soporte necesario para su interacción.

Conclusiones y trabajo futuro

Por último, en el Capítulo 7 se exponen las principales conclusiones que se derivan del trabajo completo, resaltando algunas propuestas de trabajo futuro que han de abordarse para continuar avanzando en la línea de investigación iniciada con la presente tesis doctoral.

Apéndices

Como complemento a los capítulos correspondientes a las partes mencionadas anteriormente, se incluyen una serie de apéndices al final del documento. Los Apéndices A y B describen algunos algoritmos básicos que se utilizan durante el trabajo.

Así, el Apéndice A expone el método *ekf* de validación cruzada utilizado como paso previo de selección de componentes principales en el contexto de la imputación de datos faltantes. Por otro lado, el Apéndice B introduce los fundamentos del algoritmo PSO que sirve de base en la solución de posicionamiento de nodos *relay* propuesta.

Adicionalmente y con objeto de cumplir con la normativa vigente relativa a las tesis con mención internacional, se incluyen dos apéndices más en inglés. El primero de ellos, Apéndice C, presenta un resumen amplio del documento completo que expone de forma reducida los trabajos realizados, motivados por el objetivo principal del presente documento, la mejora y robustecimiento de la seguridad, como parte de la supervivencia, en redes ad hoc. En el segundo, Apéndice D, se indican las conclusiones y líneas de trabajo futuro ya presentadas en el Capítulo 7.

Parte I

Seguridad y supervivencia en redes ad hoc

Capítulo 2

Seguridad para la supervivencia en redes ad hoc

Contenido

Gracias a sus características, capacidades y posibilidades, las redes ad hoc son aplicables a multitud de contextos y problemas [2]. Sin embargo, son precisamente las cualidades que las hacen únicas las responsables de sus principales vulnerabilidades, que al final se convierten en amenazas reales dando lugar a un gran número de potenciales ataques. Dichos ataques tienen un efecto directo sobre servicios básicos de seguridad de la red como, por ejemplo, la disponibilidad o la integridad [7]. A su vez, esto puede tener consecuencias importantes de cara a la consecución de los objetivos o provisión de servicios para los que la red fue concebida.

En general y especialmente en este tipo de redes, es necesaria la inclusión de técnicas y metodologías que aseguren ciertos requerimientos de seguridad asociados directamente a la continuidad, rendimiento y servicios que ofrece el sistema. En este sentido, se contemplan tres líneas diferentes de defensa. Estas son: *prevención*, *detección* y *respuesta/tolerancia*.

La acción de las anteriores líneas de defensa contribuye al fortalecimiento y robustecimiento del sistema ante amenazas o ataques. Desde esta perspectiva, toda red o sistema debería ser provisto de la capacidad de *supervivencia* que, como se ya se introdujo con anterioridad, va más allá de la mera utilización de una sola línea de defensa, sobre una sola capa del modelo de red y para objetivos y ataques muy concretos. Sin embargo, es escasa la propuesta de soluciones que abogan por introducir cualidades o requisitos que doten de esta capacidad a este tipo de redes [10].

A lo largo del presente capítulo se realizará un estudio de las principales amenazas de seguridad existentes en el contexto de la redes ad hoc, así como el esfuerzo realizado por la comunidad investigadora para combatirlas. Haremos especial hincapié en la búsqueda de soluciones basadas en esquemas reactivos y/o tolerantes.

El resto del capítulo se organiza de la siguiente forma. En la Sección 2.1 discutiremos las principales vulnerabilidades de seguridad encontradas en este tipo de entornos. Seguidamente, en la Sección 2.2 se describen aquellos requisitos de seguridad que garanticen su correcto funcionamiento. En la Sección 2.3 nos centraremos en la especificación de las diversas amenazas a las que se enfrentan las redes ad hoc, haciendo especial hincapié en el alto número de ataques referenciados en la literatura. Continuaremos en la Sección 2.4 con el estudio de las principales soluciones de seguridad encontradas en la bibliografía especializada, para centrarnos después, de acuerdo a la línea principal del presente trabajo de tesis, en esquemas de respuesta/tolerancia en la Sección 2.5. Una vez concluido este estudio, en la Sección 2.6 presentaremos las tendencias y retos abiertos en el contexto de la seguridad en redes ad hoc. Por último, las conclusiones del presente capítulo se exponen en la Sección 2.7.

2.1. Vulnerabilidades de seguridad en redes ad hoc

Algunas características de las redes ad hoc que derivan en vulnerabilidades de seguridad y por ende en potenciales ataques, son las siguientes [3, 6, 13]:

- *La ausencia de una infraestructura fija y la falta de una gestión centralizada.* Ambos aspectos están en concordancia con la propia naturaleza distribuida de este tipo de redes. Esto hace que sean los propios nodos los que se organicen en cierto sentido para establecer las oportunas comunicaciones salto a salto (*multi-hop*) y ayudar así en la provisión de determinados servicios de red. De este modo es necesaria la cooperación de cada nodo en el proceso de comunicación, sobre todo en el reenvío de información cuando es necesario. Sin embargo, la cooperación entre nodos no siempre se cumple, principalmente debido a la

potencial aparición de comportamientos maliciosos (*malicious nodes*), nodos egoístas (*selfish nodes*) o funcionamientos erróneos (*faulty nodes*). La diferencia fundamental entre los dos primeros radica en la motivación del ataque. En el primer caso, se trata de interrumpir malintencionadamente el proceso normal de funcionamiento de la red, mientras que en el segundo caso los nodos intentan aumentar su propio beneficio a costa de otros nodos; por ejemplo, ahorrando energía como consecuencia de disminuir su grado de participación en el proceso de retransmisión.

- *La protección física.* Los nodos de la red puede que no estén protegidos físicamente, lo que los hace propensos a posibles accesos al hardware y software instalado. Una vez dentro, el agente malicioso podría robar información sensible, cambiar su modo de funcionar, dañarlo en algún sentido, etc.
- *Topología cambiante.* Las redes ad hoc son propensas a cambiar su topología, sobre todo si les suponemos la capacidad de establecer conexiones inalámbricas. Cambios continuos en la topología de la red, provocados principalmente por la entrada y salida de nodos hacia y desde la red, hace difícil distinguir entre nodos legítimos y aquellos que han sido comprometidos. Desde el punto de vista de los algoritmos de encaminamiento desplegados, estos tendrían que proveer mecanismos robustos de distinción entre comportamientos cuyo origen es malicioso y aquellos derivados del propio dinamismo de la red.
- *La provisión de canales de comunicación inalámbricos.* Aunque el término "red ad hoc" no implica directamente la capacidad de establecimiento de canales de comunicación inalámbricos, sí es una característica compartida por la mayoría de este tipo de redes. Así, la propia naturaleza abierta del medio de transmisión escogido las hace propensas a escuchas, modificación, inyección, etc., de información sin necesidad de tener acceso físico a los dispositivos.
- *Los recursos limitados de la red.* Aún no siendo un problema relevante en algunos tipos de redes ad hoc, este es en general un aspecto a tener en cuenta. Estamos hablando de los recursos limitados que poseen los nodos de la red, haciendo hincapié en aquel que les dota de autonomía propia: la batería. Dada la limitación en el tiempo de funcionamiento que conlleva el uso de baterías, este tipo de redes son propensas a ataques enfocados en agotar lo antes posible los recursos energéticos de los nodos. Por ejemplo, haciéndolos trabajar de manera continuada aún sin ser imprescindible para el correcto funcionamiento de la red.

2.2. Requisitos de seguridad en redes ad hoc

Una vez expuesta la intrínseca relación existente entre cualidades y vulnerabilidades en este tipo de redes, se hace necesaria la provisión de una serie de requerimientos

de seguridad que, en cierto modo, garanticen la fiabilidad y continuidad de los servicios que ofrece la red [6, 7, 13, 14]. Podemos distinguir así entre los siguientes requerimientos o servicios de seguridad principales:

- *Disponibilidad.* Los servicios provistos por la red han de estar disponibles a pesar de la existencia de problemas, ataques o fallos. La disponibilidad de la red se puede ver mermada ante, por ejemplo, el agotamiento de los recursos energéticos de los nodos (*depletion attacks*) o la eliminación de paquetes que en condiciones normales se hubieran reenviado para llegar a su destino (*dropping attacks*).
- *Confidencialidad.* La confidencialidad en las comunicaciones garantiza que la información transmitida entre entidades legítimas de la red no pueda ser revelada a entes ajenos a esta. Es usual el uso de mecanismos criptográficos para garantizar este servicio. A su vez, en entornos dinámicos y cambiantes con el tiempo, es recomendable denegar el acceso a la información que circula por la red a aquellos nodos que dejen la red (*forward secrecy*). En su caso, se tendría que evitar también que nuevos nodos en la red accedan a información anterior al instante de tiempo en el que se incorporaron (*backward secrecy*).
- *Privacidad.* A través de este servicio se persigue ocultar aquellos datos que son privados y que no resultan necesarios para el normal funcionamiento de la red; por ejemplo, preservar la identidad de los nodos de una comunicación ante el resto de la red. Por el contrario, y para clarificar la diferencia con respecto a la confidencialidad, a través de esta última se oculta la información transmitida por la red a entidades externas y no autorizadas.
- *Autenticidad o autenticación.* Capacidad de poder determinar la identidad de un determinado nodo de la red. Sin mecanismos que garanticen este servicio, un nodo malicioso podría suplir a cualquier otro legítimo actuando como tal para después llevar a cabo otro tipo de acción maliciosa.
- *Autorización.* Como su propio nombre indica, se basa en garantizar que solo aquellos nodos autorizados puedan formar parte de la red y, por lo tanto, se les permita hacer uso de los servicios que esta oferta.
- *Integridad.* La integridad se refiere a la garantía de que la información intercambiada entre los nodos o entidades de la red no ha sido alterada. La alteración o modificación de la información que circula por la red puede ser intencionada o accidental; esta última debida, por ejemplo, a interferencias producidas durante la transmisión por un medio inalámbrico. En todo caso, sea cual sea el motivo, es necesaria la pronta detección y corrección de la información alterada.
- *No repudio.* El no repudio se entiende como la capacidad que tiene la red para evitar que un nodo que participó en una comunicación no pueda negarla (repudiarla) de manera alguna. Este servicio es útil para la detección y aislamiento

de nodos comprometidos y se soporta mediante el uso de algoritmos de encaminamiento seguros basados en, por ejemplo, esquemas de reputación (*reputation*) o confianza (*trust*) en donde se penalizan comportamientos no adecuados para el correcto funcionamiento de la red [15].

- *Freshness.* A través de este servicio se persigue garantizar que los mensajes que circulan por la red provengan de comunicaciones recientes. Esto evita el uso de mensajes antiguos que usuarios mal-intencionados pudieran aprovechar de cara a la perturbación de uno o varios servicios de la red.

2.3. Amenazas a la seguridad en redes ad hoc

Cualquier vulnerabilidad detectada en un sistema y que comprometa algún aspecto o servicio de seguridad, conlleva un amenaza (riesgo) contra el mismo. Así, un atacante podría sacar provecho de dicha debilidad para perpetrar o llevar a cabo un determinado ataque.

Son muchos los trabajos en los que se estudian los diferentes tipos de ataques en el contexto de las redes ad hoc [6, 9, 16, 17]. En la Tabla 2.1 se exponen los principales ataques que afectan a este tipo de redes. Su elevado número se debe, principalmente, a la generalidad del término "ad hoc" y los diferentes y específicos subtipos de red que engloba.

Tabla 2.1: Principales ataques en redes ad hoc [5][18].

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.
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.

Continúa en la página siguiente

Tabla 2.1 – *Continúa de la página anterior*

Ataque	Descripción
Eavesdropping	Escucha de las comunicaciones privadas, es decir, 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, por ejemplo, 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, por ejemplo, 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 (*Transfer 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.
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.

Continúa en la página siguiente

Tabla 2.1 – *Continúa de la página anterior*

Ataque	Descripción
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 (por ejemplo, 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, por ejemplo, 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).

Establecer una clasificación de ataques diferenciada en algún sentido o aspecto es una tarea difícil dada la gran cantidad y variedad de estos. A su vez, tal diferenciación no siempre es acertada. En ocasiones se discierne entre ataques cuya disimilitud es muy sutil y que tienen un impacto y comportamiento similares [5].

A continuación se presentan algunas propuestas de clasificación que, podríamos decir, son las más extendidas y aceptadas dentro de la comunidad investigadora. Así, se catalogan los ataques según sea su estado, su comportamiento, su propósito [6], la capa de protocolo en la que actúa [9] y algún otro criterio distinto de los anteriores [19–21]:

- *Clasificación de ataques según su estado.* Atendiendo a esta organización, distinguiremos entre ataques internos o externos dependiendo de si el atacante es una entidad que pertenece a la red o, por el contrario, ajena a esta, respectivamente. Algunos ejemplos de ataques considerados como externos son los ataques de *jamming* y *tampering*, mientras que ataques como el de *rushing* o de modificación de rutas de encaminamiento pueden verse como ataques internos.
- *Clasificación de ataques según su comportamiento.* De igual manera a la organización anterior, diferenciaremos aquí entre ataques pasivos o activos. Los primeros se caracterizan porque no interfieren en modo alguno en el normal funcionamiento de la red y solo se limitan a monitorizar, escuchar y/o analizar la información que circula por la red. Con respecto a los segundos, requieren de un acceso físico a la red o a parte de ella para influir en su comportamiento normal. El ataque de *eavesdropping* es el único que puede considerarse como pasivo dentro de esta clasificación.
- *Clasificación de acuerdo al propósito del ataque.* De acuerdo a esta clasificación, tendremos ataques que atentan contra la disponibilidad de los servicios de la red, en donde encajarían ataques de DoS que modifican o alteran el normal funcionamiento del algoritmo de *routing* a nivel de red o interfieren sobre diferentes funcionalidades a nivel físico; ataques dirigidos a romper la privacidad y la confidencialidad de la información transmitida como, por ejemplo, el ataque de *eavesdropping*; y por último, contra la integridad de la información, que intentan alterar los datos que se transmiten. Algunos ataques que se enmarcan dentro de esta categoría son el de *data tampering* o *tampering* a nivel físico y el de modificación (*modification*) o duplicación *replay* a nivel de red.
- *Clasificación según la capa del protocolo de red donde actúan.* Como su propio nombre indica, los ataques pueden organizarse según la capa del protocolo de red sobre la que actúan [9]. Por ejemplo, *jamming* o *tampering* a nivel físico o *blackhole* a nivel de red.
- *Otras clasificaciones.* Aún compartiendo similares características, subtipos de redes ad hoc añaden a su vez sus propias cualidades que se traducen en vulnerabilidades específicas. Por ejemplo, en el trabajo [19] se define un ataque específico en redes VANET al que los autores denominan ataque de *movement tracking*. Este ataque se refiere a la capacidad que tiene el nodo malicioso para conocer la posición y velocidad de un determinado nodo objetivo, pudiendo inferir o detectar el comportamiento futuro del nodo e interferir, por ejemplo,

en sus comunicaciones. Adicionalmente, se propone clasificar los ataques producidos en redes VANET en cuanto a la naturaleza, el impacto, el ámbito o el objetivo perseguido. Ahmed *et al.* [20] proponen una clasificación de ataques en función de su capacidad de deteriorar el rendimiento de sistemas de defensa basados en medidas de confianza. Los autores denominan a estos ataques *TM (Trust Management) related attacks*. Desde el punto de vista de la teoría de juegos, el trabajo [21] propone una clasificación de ataques entre los que los autores denominan *palpable attacks* y *subtle attacks*. Dentro del primer grupo se engloban aquellos que tienen un impacto relevante sobre el funcionamiento de la red y, por tanto, de cara al usuario final. Un ejemplo de ataque que los autores consideran claramente englobado dentro del primer grupo, es el ataque *jamming*. En relación al segundo grupo, el trabajo engloba aquellos ataques considerados no tan lesivos para la red como los anteriores. Ataques como *eavesdropping* o *grayhole* son algunos ejemplos.

Es obvio que no existe una guía definitiva para clasificar ataques en redes ad hoc. Con objeto de solventar esta cuestión, García-Teodoro *et. al* [5] proponen una nueva taxonomía para la clasificación de los principales ataques contemplados en este tipo de redes, cuyo fin es conseguir sistemas holísticos de detección más eficaces. Para llevar a cabo la clasificación, consideran una jerarquía compuesta por tres niveles que, podríamos decir, encauza o guía la catalogación ubicando a cada ataque dentro unas de las siete posibles clases propuestas. Con el fin de etiquetar un ataque en una de las anteriores clases, se considera el siguiente flujo: *acción* → *efecto* → *procedimiento*. El primer aspecto discierne entre qué tipo de *acción* ha de realizar el ataque para que este se lleve a cabo. El segundo se centra en cuál es el *efecto* producido sobre el sistema y los servicios de seguridad afectados. Por último, se considera el *procedimiento* llevado a cabo para realizar el ataque como diferenciador del mismo.

2.4. Soluciones de seguridad en redes ad hoc

Tras la exposición y discusión anterior, parece necesario el diseño y propuesta de esquemas de seguridad globales que abarquen la mayor cantidad posible de características asociables a sistemas con capacidad de supervivencia y, por ende, que sean capaces de actuar sobre un mayor número de ataques o amenazas. Sin embargo, este hecho no se refleja en las soluciones que podemos encontrar en la literatura especializada, las cuales se centran en líneas de defensa concretas, ataques bien definidos y/o capas específicas dentro de la pila de protocolos TCP/IP [5, 6, 22].

Estrictamente hablando, solo aquellos mecanismos o métodos orientados a la evitación de ataques se podrían considerar como soluciones preventivas (resistentes, desde el punto de vista de la supervivencia). De acuerdo a [23], una red ad hoc

debería "(*i*) *proveer mecanismos de seguridad efectivos en contra de comportamientos malintencionados en la red*, *y* (*ii*) *abogar por la cooperación entre nodos de la red.*" Según estas premisas, se considerarán como preventivas aquellas soluciones que, aunque fomentan la cooperación entre nodos de la red, no evitan que se produzca el ataque. Estas soluciones se centran en su mayoría en técnicas que proveen mecanismos de autenticación de los nodos durante el proceso de encaminamiento [24, 25]. Esta metodología es especialmente útil para controlar la adición de nuevos nodos no autorizados, siendo efectivas frente ataques externos. Por otro lado, son costosas de implementar y poco eficientes desde el punto de vista energético, siendo este último factor un hándicap importante para su empleo en redes WSN, normalmente limitadas en este sentido.

Adicionalmente, encontramos técnicas preventivas que se basan en la propuesta de soluciones seguras mediante la modificación de los protocolos habituales de encaminamiento en este tipo de redes. A diferencia de las soluciones basadas en la certificación de la identidad de los nodos, este tipo de propuestas persiguen garantizar que las rutas creadas son correctas, bien mediante la comparación de rutas entre nodos vecinos, a través de la solicitud de confirmación de rutas, o mediante la inserción o envío adicional de información sobre las rutas que se establecen. Algunos autores añaden nuevos mensajes al protocolo de encaminamiento existente para asegurar el proceso de establecimiento de rutas, como por ejemplo en el caso del protocolo SAODV (*Secure AODV*) [26], o las soluciones planteadas en [27] y [28]. Otros esquemas se basan en reforzar la cooperación entre nodos mediante cifrado y descifrado colaborativo de paquetes para el establecimiento de rutas [29]. Finalmente, otras soluciones se basan en almacenar información relativa al comportamiento de nodos considerando varios saltos en la ruta. A su vez, consideran nuevas métricas para establecer cierto grado de confiabilidad sobre aquellos nodos que participan en el proceso de establecimiento de rutas [30, 31]. Este tipo de soluciones introduce una carga computacional mayor, así como un *overhead* mayor en la red, aspectos ambos que repercuten directamente en un mayor consumo energético global.

Por otro lado, se podría pensar en soluciones preventivas que motiven la participación de los nodos en el proceso de encaminamiento, evitando aquellos que no desean contribuir con el correcto funcionamiento de la red. Dentro de este conjunto diferenciaremos entre métodos basados en la gestión de la reputación de los nodos, o aquellos que premian la buena actuación de estos mediante el pago o acumulación de créditos. En relación a los primeros encontramos soluciones como Pathrater [32], otras basadas en el uso de autoridades de confianza que centralizan la gestión de la reputación de los nodos y otras que descentralizan este proceso, claramente más escalables, como CORE [33], *Friends & Foes* [34] o SORI (*Secure and Objective Reputation-based Incentive*) [35]. A su vez, en lo referido a soluciones basadas en créditos, se puede diferenciar entre dos tipos de modelos de pago: los basados en *monederos* de mensajes o los que emplean el *comercio* de mensajes [36]. En el primer

caso, es el nodo origen el que paga a los nodos intermedios por su servicio. Por lo tanto, un nodo que requiera comenzar una nueva transmisión deberá asegurarse de tener la cantidad suficiente de créditos para poder iniciarla. Un ejemplo de este tipo de sistemas es la solución TFT (*Tit-For-Tat*) [37]. Por el contrario, en el modelo basado en el comercio de mensajes, estos últimos son considerados como mercancía y, consecuentemente, es el receptor quien paga tanto a los nodos intermedios como al nodo origen. En estos esquemas es usual la presencia de entidades globales que centralizan el comercio de los créditos. Algunos ejemplos de este tipo de soluciones son Mobicent [38] o Sprite [39], mejorada en la propuesta Express [40]. Este tipo de técnicas motivan a los nodos a participar en el proceso de encaminamiento, evitando así comportamientos *selfish*. Sin embargo, adolecen de posibles fraudes en el pago e injusticias cuando no todos los nodos de la red transmiten cantidades similares de información.

En otro contexto y a pesar de los grandes esfuerzos realizados por la comunidad científica en la propuesta de soluciones preventivas, todavía es necesaria la subsecuente etapa de detección (reconocimiento, en el contexto de la supervivencia de sistemas). Con este propósito, son muchas las soluciones propuestas por la comunidad investigadora en el marco de las redes ad hoc [41]. Dentro de este tipo de soluciones, abundan aquellas basadas en el empleo de la solicitud explícita de la confirmación de los paquetes enviados por parte de los nodos que iniciaron la comunicación. Es la información recogida de las confirmaciones (ACK) la que se tiene en cuenta a la hora de etiquetar un nodo como malicioso. Por ejemplo, son varias las soluciones que proponen el empleo de confirmaciones a dos saltos [42–44]. Algunas de ellas más complejas, como la solución TWOACK de Balakrishnan *et al.* [45] o [46], emplean estructuras en árboles binarios *Merkle* formados a partir de las confirmaciones recibidas con objeto de chequear si una ruta es segura.

Aunque consideradas como estrategias preventivas, las técnicas basadas en reputación pueden ser vistas también como soluciones de detección ya que, aunque establecidas como primera línea de defensa, contribuyen a la detección de nodos maliciosos disminuyendo el nivel de reputación o confianza de dichos nodos. Un esquema de este tipo de soluciones es el protocolo CONFIDANT (*Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks*) [47], que controla el comportamiento de sus vecinos asignándoles una cierta valoración. De manera similar encontramos el esquema *Friends & Foes* [34], anteriormente citado, o el trabajo [48], que se basa en el concepto de consistencia dentro de un círculo de confianza o *inner-circle consistence* para identificar respuestas falsas o falsificadas. La principal desventaja de estos esquemas es el exceso de tráfico adicional que introducen en la red debido a la compartición de información de reputación entre nodos.

Otras soluciones se basan simplemente en monitorizar el entorno donde se aplican. Estas técnicas consideran una serie de características y las comparan con ciertos umbrales (adaptativos o no) para tildar a un nodo como malicioso. Marti *et al.* [32]

presentan un sistema pionero llamado Watchdog, que compara los paquetes enviados y que fueron reenviados por el siguiente nodo en la ruta para detectar actuaciones maliciosas. Kurosawa *et al.* [49] proponen un método más complejo teniendo en cuenta el flujo de paquetes RREQ (*Route REQuest*) y RREP (*Route REPly*) en el proceso de establecimiento de rutas del algoritmo de encaminamiento. La solución bautizada por sus autores como DPRAODV (*Detection, Prevention and Reactive AODV*) [50] se basa en chequear si el número de secuencia recibido por un nodo a partir de un paquete RREP de otro nodo intermedio en la ruta, supera un determinado umbral.

Son numerosos los ámbitos de uso de las técnicas de *machine learning* para la búsqueda de patrones inmersos en cantidad de conjuntos de datos variopintos. Un ejemplo de uso de estas técnicas es en el contexto de la predicción del tiempo atmosférico a través de la recolecta y medición previa de valores relativos a la humedad, velocidad del viento, temperatura o precipitaciones [51]. En el contexto de la seguridad y concretamente en el de detección de intrusiones, estas técnicas también se aplican de manera satisfactoria. Sin ir más lejos, Zhang *et al.* [52] introducen un esquema local y colaborativo en el que cada nodo móvil incorpora un agente IDS basado en SVM (*Support Vector Machine*), que se encarga de monitorizar su entorno local. En la referencia [53] los autores presentan una solución *cross-feature* en la que aglutinan un total de 141 características de tráfico y topología de red para llevar a cabo la posterior detección mediante el empleo de un clasificador. El trabajo [54] introduce una aproximación multi-capa para las capas MAC, encaminamiento y aplicación. La solución contempla tres sistemas diferentes que actúan sobre cada una de las capas mencionadas: un clasificador Bayesiano en la capa MAC, cadenas de Markov en la capa de encaminamiento y un algoritmo de asociación de reglas en la capa de aplicación. Los resultados obtenidos para cada uno de los sub-sistemas se integran en un módulo local y el resultado final se envía a un módulo global de detección.

Finalmente, son algunos los trabajos que computan un *modelo analítico* para determinar la dinámica de un determinado protocolo con el fin de concluir potenciales inconsistencias durante su funcionamiento. En este sentido, los autores del trabajo [55] modelan el protocolo AODV (*Ad hoc On-demand Distance Vector*) mediante una máquina de estados finitos EFSA (*Extended Finite State Automaton*) para obtener su comportamiento normal y proponer así la detección de actuaciones maliciosas basada en especificaciones de tipo estadístico. Los autores del trabajo [56] proponen un modelo teórico para las diferentes causas en la pérdida de paquetes en redes ad hoc cuyo protocolo de encaminamiento es DSR (*Dynamic Source Routing*). Con este modelo son capaces de detectar ataques de *dropping* y distinguirlos de eventos legítimos como son las colisiones o errores en el canal. Nos obstante, el trabajo estudia una topología limitada y no considera aspectos de movilidad en los nodos. En la propuesta [57] y con objeto de completar el modelo propuesto en [56], se expone una

heurística que tiene en cuenta la influencia de la movilidad de los nodos en redes MANET. Para llevar a cabo la solución se consideran diferentes parámetros tanto de la capa MAC como de la capa de red, proponiendo así una solución multi-capa que ofrece mejores resultados que las anteriores propuestas.

2.5. Esquemas de respuesta/tolerancia ante amenazas de seguridad

Aunque el uso y despliegue de soluciones preventivas y de detección es necesario de cara a conseguir sistemas más robustos desde el punto de vista de la seguridad, no son suficientes para contrarrestar las consecuencias de un ataque en curso. Por este motivo es necesaria la adición de nuevas medidas defensivas que reaccionen o toleren en cierta medida los efectos producidos por el ataque. Así, y en concordancia con la línea principal de investigación del presente trabajo, a lo largo de esta sección prestaremos especial atención a las principales metodologías, técnicas o esquemas de respuesta/tolerancia propuestos por la comunidad científica.

Dentro del conjunto de las soluciones reactivas, la gran mayoría de ellas se basa en intentar aislar el nodo o nodos maliciosos con el objetivo de preservar o recuperar el funcionamiento normal de la red. Aunque no es trivial establecer una clasificación de soluciones de seguridad de reacción o tolerancia ante ataques, se propone a continuación una distinción tentativa que pretende ordenar las soluciones existentes en la siguientes clases: (*i*) exclusión de nodos, (*ii*) exclusión de nodos y notificación, (*iii*) aislamiento de nodos, y (*iv*) otros esquemas. En la Figura 2.1 se muestra un esquema que presenta las soluciones que se describen a continuación, agrupadas acorde a la clasificación anterior.

2.5.1. Soluciones basadas en la exclusión de nodos

El principal propósito de este tipo de técnicas es eludir el mal comportamiento de un nodo de tal manera que se excluye como nodo válido intermedio dentro de un esquema de encaminamiento *multi-hop*. Estos esquemas tienen en común la siguiente manera de proceder: primero, solo los nodos que pertenecen al vecindario del malicioso son conscientes de este hecho para, en segundo lugar, tratar de evitar aquellas rutas de las que forma parte el nodo malicioso en cuestión.

Por ejemplo, en la referencia [58] se extiende el protocolo DSR añadiendo para cada nodo un modelo HMM (*Hidden Markov Model*) encargado de obtener su confia-

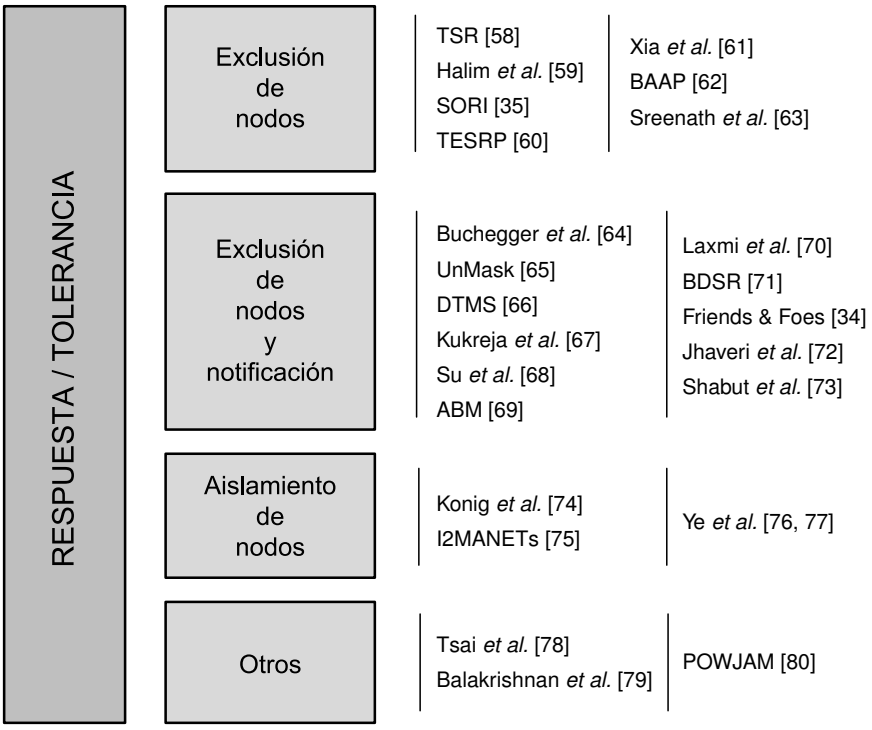


Figura 2.1: Clasificación de las soluciones de respuesta/tolerancia en redes ad hoc.

bilidad o *trustworthiness*[1]. TSR (*Trust-based Secure Routing*), como llaman los autores a su propuesta, actúa frente a los comportamientos egoístas seleccionando aquellas rutas cuyos nodos poseen un valor más alto de confiabilidad. De esta forma, el o los nodos maliciosos son eludidos. Otra modificación sobre el protocolo DSR se expone en [59], en donde se incluyen dos agentes en 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. Cuando se detecta que un nodo no se comporta como era de esperar, dicho valor se decrementa. Después, el agente ROA se encarga de seleccionar una ruta fiable descartando aquellos nodos con un nivel bajo de confianza.

Los autores en [60] proponen el protocolo TESRP (*Trust and Energy aware Secure Routing Protocol*), especialmente indicado para la exclusión de nodos maliciosos considerando las restricciones energéticas existentes en las redes WSN. Dicho protocolo se compone principalmente de un módulo evaluador (*Trust evaluator*) que, en función de los comportamientos actuales y esperado de un nodo, le asigna un determinado nivel de confianza. A su vez, un módulo adicional denominado *Route setup* se encarga de establecer rutas seguras compuestas por aquellos nodos cuyo nivel de confianza

[1]La confiabilidad o *trustworthiness* se define como un valor cuantitativo que indica la probabilidad de que un nodo se comporte como se espera [81].

supera un determinado umbral. Además, para determinar dichas rutas, también considera la energía restante en los nodos y el número de saltos hacia el destino.

Otros mecanismos existentes hacen uso de la información que provee algún sistema de reputación para reaccionar ante el ataque. Por ejemplo, SORI [35] descarta los paquetes generados por un nodo egoísta con cierta probabilidad. Este valor aumenta conforme el valor de reputación del nodo disminuye, reduciendo así la capacidad de transmisión de este. Más recientemente, el trabajo [61] presenta un algoritmo de reputación que se utiliza como mecanismo de respuesta. En este caso, se evalúa qué valor futuro de confiabilidad tendrá un nodo a través de un algoritmo dinámico de predicción que tiene en cuenta comportamientos pasados. De manera similar a [58], cuando un nodo no se comporta de manera adecuada, el sistema decrementa su valor de confiabilidad de tal manera que no se envían ni se reciben paquetes a través de dicho nodo una vez superado un determinado umbral. Esta propuesta posee una particularidad especial: el nodo puede volver a ser considerado legítimo. Por lo tanto, este sistema puede ser visto como un método de tolerancia que conserva el nodo dentro de la red sopesando las consecuencias negativas que pudiera tener su exclusión sobre el rendimiento de esta. En [62], cada nodo crea su propia tabla de *legitimidad* durante la fase de establecimiento de rutas. Las filas de dicha tabla (una por cada nodo de la red) se calculan atendiendo a dos factores: el número de veces que un determinado nodo ha sido seleccionado como nodo intermedio, y el número de veces que se llegó al nodo destino utilizando dicho nodo como nodo intermedio. De igual forma a como actuaban las soluciones anteriores, si un nodo se comporta maliciosamente se decrementa su valor de legitimidad. De esta manera, el sistema elegirá aquellos nodos que posean un valor elevado como candidatos para incluirlos dentro de una posible ruta.

Otros sistemas, más simples, hacen uso de umbrales preestablecidos sobre determinados parámetros para luego llevar a cabo una respuesta. Por ejemplo, en la solución [63] el nodo origen bloquea aquellos nodos cuyos valores de secuencia en los paquetes RREP tienen un valor que supera un umbral predefinido, valor que es habitual que un nodo *blackhole* modifique para atraer tráfico hacia él. Adicionalmente, también cuenta el número de RREQ que se reciben de cada nodo para luego descartar aquellos que superan, en media, un determinado umbral establecido o *cut-offRate*.

2.5.2. Soluciones basadas en la exclusión de nodos y notificación

Estos mecanismos de respuesta mejoran los anteriores mediante la adición de métodos de notificación de nodos maliciosos al resto de la red, usualmente empleando mensajes específicos. Una vez que se han emitido las notificaciones, cualquier nodo de la red puede evitar ahora dentro de sus rutas a aquellos nodos que se comportan

de manera inadecuada. De esta forma, se puede decir que se produce una respuesta más global que en las propuestas que se vieron en la sección anterior.

Un trabajo posterior a [47] basado en un esquema de reputación se presenta en la referencia [64]. Aquí, la respuesta es llevada a cabo de manera cooperativa entre varios módulos. Estos son: el módulo de gestión de reputación, el módulo de gestión de confianza y el módulo de gestión de rutas. Una vez que se detecta un comportamiento sospechoso, es el módulo de gestión de reputación el que se encarga de evaluar el comportamiento pasado. Si se supera un determinado umbral para el comportamiento normal esperado, se envía una notificación al módulo de gestión de rutas que, acto seguido, eliminará dicho nodo de la ruta. Adicionalmente, el módulo de gestión de confianza enviará un mensaje de alarma a la vecindad. Cada mensaje de alarma que se recibe en un nodo se pasa al módulo de gestión de confianza con objeto de determinar si ese mismo nodo ya se evaluó de manera similar por otros nodos de confianza. Si existen suficientes evidencias sobre la malignidad del nodo se notifica al módulo de reputación, que descarta al susodicho como posible alternativa de encaminamiento.

En [65] se presenta un mecanismo de respuesta aplicable de dos maneras: directa o indirectamente. En el primer caso, cada nodo se encarga de eliminar de su tabla de encaminamiento un nodo malicioso previamente detectado. En el segundo caso, un nodo monitor enviará un mensaje de alarma a los vecinos. Dependiendo de la cantidad de mensajes de alarma recibidos en un nodo dado, este último podría eliminar al nodo malicioso de su tabla de encaminamiento. De manera similar, el sistema DTMS (*Distributed Trust Management System*) [66] calcula el nivel de confianza de un nodo en función de (*i*) la observación directa de parámetros que recoge de sus vecinos, (*ii*) de la observación indirecta que cada nodo distribuye acerca de su vecindario, y (*iii*) de los valores previos de confianza. Principalmente, es la observación directa la que tiene mayor peso para la asignación final del valor de confianza, ya que considera una gran cantidad de parámetros. Por ejemplo, tiene en cuenta la participación del nodo en el proceso de encaminamiento, la integridad de los paquetes de datos y control, congestión de sus enlaces, energía, etc. Una vez obtenido el nivel de confianza, dicho nodo no se tendrá en cuenta como parte de una ruta válida cuando este valor no supere un cierto umbral. A su vez, estos valores se difunden por toda la red para su consideración de forma global.

A través del trabajo [67] se presenta una solución de respuesta ante ataques *selfish*, *blackhole* y aquellos que introducen cambios en la topología de una red MANET de manera intencionada. La propuesta se basa en la adecuada modificación del protocolo de encaminamiento DSR para conformar el sistema de evaluación de los nodos, por un lado, a través de su comportamiento en el proceso de reenvío de paquetes (para combatir los ataques *selfish* y *blackhole*) y, por otro, computando el número medio de veces que el nodo abandona y entra en la red en un determinado tiempo para contemplar el segundo tipo de ataques que afectan a la topología de la red. Dicho

sistema se compone de una serie de nodos convenientemente seleccionados y ubicados que permiten la monitorización de todos los demás (nodos monitores). Al existir un número limitado de monitores (a diferencia de las soluciones encontradas en la literatura, en las que cada nodo es a su vez un monitor), es necesaria la propagación de la información de confianza obtenida por cada monitor de los nodos "normales" u otros monitores. Tanto para esta propagación de valores de confianza como para cuando un nodo necesita comunicarse con otro, se establece una selección de rutas que evita aquellos nodos cuyo nivel de confianza está por debajo de cierto umbral.

En relación a la auto-protección de los sistemas de respuesta/tolerantes basados en el establecimiento de relaciones de confianza entre nodos, es de mencionar la aparición de varias amenazas que no están directamente dirigidas a la red ad hoc, sino que se centran en el propio esquema de seguridad. Por ejemplo, los autores en [73] proponen un esquema de confianza que, a priori, se idea para establecer rutas confiables ante la presencia de ataques de *dropping* mediante la observación directa e indirecta del comportamiento de un determinado nodo. Sin embargo, este sistema de reputación o confianza es susceptible de ser corrompido ante ataques que intentan desprestigiar de alguna manera la reputación de un nodo. Estos tipos de ataques son los denominados *bad-mouthing* y *ballot stuffing attack*, entre otros. Para ello idean un sistema de recomendación que trata de corroborar la honestidad de un nodo de cara a tener en cuenta su valoración de reputación con respecto al que está siendo evaluado. Así, el sistema considera aspectos como la cercanía del nodo del que se ha recibido la evaluación con respecto al evaluado y su interacción con el nodo evaluado, entre otros, las cuales permiten probar la bondad del nodo evaluador.

Los autores de la propuesta [68] presentan un sistema de respuesta basado en el bloqueo de nodos. En dicha solución existe un conjunto de agentes que monitorizan la red y con capacidad de comunicación entre ellos. Cuando se detecta un nodo malicioso, primero, se envía un mensaje de bloqueo a los nodos asociados al agente que descubrió la amenaza y, segundo, este mismo mensaje se disemina entre los demás agentes de manera que el nodo es cuestión es eludido. Del mismo autor primero de la propuesta anterior, en el trabajo [69] se plantea el envío de un mensaje *broadcast* de bloqueo a todos los nodos de la red cuando el valor de sospecha supera un determinado umbral, excluyéndose así 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. De esta forma, al recibir el mensaje, todos los nodos incluyen al nodo malicioso en una lista negra. De manera similar, en el trabajo [70] los autores proponen responder ante ataques *jellyfish* mediante la difusión de listas negras de nodos cuyo comportamiento coincide con este tipo de ataques. Una vez detectado el nodo malicioso, es el nodo detector el que se encarga de difundir este hecho a los demás nodos. El sistema es tolerante en el sentido de que ofrece hasta tres oportunidades para fijar realmente un nodo como malicioso. Por debajo de este

umbral, un nodo etiquetado como malicioso se vuelve a considerar adecuado para establecer rutas a través de él.

De nuevo es el protocolo DSR el elegido para ser oportunamente modificado, en este caso para proporcionar el mecanismo de respuesta desarrollado en [71]. El objetivo de esta solución es la evitación de nodos *blackhole* mediante la creación y difusión de listas negras de nodos. De esta manera, todos los nodos de la red conocerán cuál o cuáles son nodos *blackchole*, evitando procesar cualquier paquete que provenga de ellos. De forma similar, en el algoritmo *Friends & Foes* [34] cada nodo difunde información de dos listas diferentes: aquella formada por los nodos que participan en el proceso de encaminamiento para el reenvío de paquetes (amigos), y aquella compuesta por los nodos que no desean reenviar paquetes (oponentes). Así, un nodo legítimo podrá rechazar paquetes de control provenientes de nodos maliciosos, forzándolos a elegir rutas alternativas.

En [72] cada nodo intermedio reacciona descartando paquetes RREP si su número de secuencia supera un determinado umbral. Dicho valor se calcula a partir del número de secuencia almacenado en la tabla de encaminamiento del nodo intermedio, del número de secuencia del paquete RREP entrante y el número total de RREP recibidos. Además, la identificación del nodo malicioso se disemina por toda la red ya que se introduce en los mensajes RREP.

2.5.3. Soluciones basadas en el aislamiento de nodos

Dentro de este tipo de soluciones se incluyen aquellas propuestas que aíslan activamente al nodo malicioso. Dentro de esta categoría de mecanismos de respuesta, dicho nodo es cercado o rodeado por otros de manera que efectúan un bloqueo tanto de las comunicaciones entrantes como salientes. Por tanto, estas soluciones no realizan solo una mera exclusión pasiva del nodo de las rutas de encaminamiento como se proponía en las soluciones de las anteriores secciones.

Con este fin en mente, los autores del trabajo [74] proponen una solución multicapa. Aunque el ataque se produce en la capa de red, la reacción ante dicho ataque se realiza en la capa física creando un especie de zona de cuarentena alrededor del atacante. Aquellos nodos que se encuentren dentro de la zona de cuarentena no serán capaces ni de enviar ni de recibir paquetes. Esta propuesta se apoya en el uso de un sistema de posicionamiento, el cual provee las posiciones de los nodos en todo momento.

Persiguiendo el mismo objetivo, en la literatura existen propuestas que se basan en el empleo de agentes autónomos. Por ejemplo, en [75] se introduce un esquema que imita el sistema inmunitario humano. Así, existen una serie de agentes inmunes o IA (*Immune Agent*) que se distribuyen por toda la red encargados de detectar,

clasificar, aislar y recuperar el sistema de los efectos del ataque (esta última acción solo se realiza si es necesaria). Un nodo se aislará del resto de la red cuando haya llevado a cabo un cierto número de ataques. Aquel nodo que previamente se aisló puede recuperarse como nodo legítimo de la red cuando no suponga una amenaza para el entorno. Un esquema similar es el propuesto en [76], en donde existen dos tipos de agentes en el sistema: de detección y de contraataque. Cuando se detecta una amenaza, se disemina un mensaje de activación a todos los agentes de contraataque pero solo se activarán aquellos que pertenecen a la vecindad del nodo malicioso. Acto seguido, dichos agentes se encargarán de que el nodo con mal comportamiento no pueda recibir ni enviar paquetes. En la referencia [77], la red ad hoc se divide en *clusters* donde existirá un nodo supervisor del mismo o CH (*Cluster Head*). Cuando el nodo CH detecta que un nodo es malicioso se crea un agente de acción o AA (*Action Agent*). Este agente AA se clona y se distribuye de manera que se sitúa uno en cada vecino. Es ahora cuando cada AA chequea si el nodo malicioso se encuentra en su vecindad a un salto. En caso afirmativo, el AA permanece en el nodo; en caso contrario, se auto-clona y se mueve a los nodos de su vecindad. Esta operación se repite hasta que el nodo malicioso es completamente sitiado. El siguiente paso, una vez rodeado el nodo en cuestión, puede ser diverso: aislarlo de la red, eliminarlo de las tablas de encaminamiento, bloquear su tráfico tanto entrante como saliente, reducir su nivel de confianza para evitar que sea tenido en cuenta para construir rutas, etc.

2.5.4. Otros esquemas

La mayoría de las soluciones de respuesta o tolerancia en redes ad hoc se centran en propuestas que actúan sobre la capa de red o encaminamiento, y principalmente ideadas para contrarrestar o tolerar los efectos de determinados ataques. Presentamos a continuación algunas soluciones orientadas a la lucha contra ataques que no se centran en perjudicar el normal funcionamiento del protocolo de encaminamiento. Por ejemplo, en [78] se propone un sistema tolerante a ataques a nivel físico que tratan de agotar los recursos energéticos de los nodos de una red WSN. El atacante envía continuamente información dirigida al nodo en cuestión que hace que este permanezca siempre activo (*sleep deprivation attacks*). Los autores proponen un método que controla cuándo ha de activarse el sensor para aceptar la comunicación entrante. Así, ante la recepción de una señal externa, el sensor se activará si el número de eventos correspondientes a la recepción de una señal externa ha superado un cierto número. Este número se calcula en función del gasto energético ahorrado, estando el sensor inactivo durante N ciclos de reloj, y el gasto energético que conlleva la activación o desactivación de este.

Los autores en [79] proponen un método de tolerancia ante ataques NFJ (*Null Frequency Jamming*), que actúan sobre el proceso de recuperación de rutas en proto-

colos reactivos como DSR o AODV. Este ataque se auto-sincroniza con el intervalo de tiempo *RequestPeriod* que indica el periodo temporal establecido entre dos peticiones RREQ sucesivas hacia el mismo destino, y que representa el periodo de recuperación del protocolo. Acto seguido inyecta tramas de *jamming* a nivel de enlace en esos precisos instantes para interrumpir el funcionamiento normal del proceso de recuperación del protocolo. Los autores proponen ser tolerantes a dichas tramas imponiendo periodos *RequestPeriod* aleatorios. Para la lucha ante ataques de *jamming* inteligentes que solo actúan cuando detectan la presencia de transmisiones y en frecuencias específicas, los autores del trabajo [80] proponen un mecanismo adaptativo y tolerante ante dichos ataques. Dicha solución consiste en la adaptación de potencia con la que transmite un nodo legítimo para, primero, no ser detectado por el nodo atacante y, segundo, que se establezcan rutas alternativas multi-salto que eviten la acción del *jammer*.

2.6. Tendencias y retos abiertos

A lo largo del presente capítulo ha quedado constancia de la gran cantidad de propuestas existentes en el campo de la seguridad en redes ad hoc. No obstante, a pesar de los grandes esfuerzos realizados, todavía es necesario reforzar y estimular el desarrollo de soluciones que mejoren el rendimiento ofrecido a los usuarios finales. Para contribuir a los objetivos anteriores se introducen aquí algunos de los retos que aún quedan por solventar así como las nuevas y potenciales tendencias de investigación que al respecto pudieran surgir.

Algunos autores defienden la necesidad del diseño de nuevos protocolos y procedimientos para reforzar aspectos de seguridad tradicionales como puede ser el proceso de autenticación. En este sentido, se están desarrollando y proponiendo protocolos de encaminamiento más robustos y procedimientos colaborativos con el fin de reforzar la confiabilidad [82–84]. Aunque estos mecanismos se pueden usar de manera dinámica en un variado número de tareas (control de acceso, sistemas de basados en confianza o reputación, etc.), generalmente son vistas desde la perspectiva de la seguridad preventiva. En otras palabras, la continua aparición de nuevas amenazas concluye la necesidad de la mejora de las condiciones iniciales de seguridad establecidas para una determinada red en sintonía con las características de las nuevas amenazas. Adicionalmente, este tipo de soluciones son computacionalmente costosas y conllevan un coste energético elevado. Este hecho limita su uso en función del tipo de red ad hoc objetivo del despliegue.

A medida que surgen nuevos tipos de ataques y/o variantes, es también necesaria la aparición de nuevos esquemas más potentes y fiables de detección. Generalmente, la respuesta a ello por parte de la comunidad científica es la propuesta y desarrollo de esquemas de detección especializados. Esta diversificación o especialización

posee dos consecuencias importantes. Por un lado, proveen un mejor rendimiento en términos de detección, ya que son específicas para determinados ataques o amenazas. No obstante, esta filosofía lleva implícitos unos costes elevados de detección a medida que aumenta el número de ataques o amenazas que se quieren detectar. Para evitar este inconveniente al tiempo que se mantiene la eficacia en la detección, sería conveniente el desarrollo de esquemas de detección holísticos. De esta manera, la construcción de modelos semánticos ayudaría a la implementación de nuevos paradigmas de detección que eviten las particularidades de cada ataque añadiendo capacidades de detección más globales.

Adicionalmente y también motivado por la continua evolución de las amenazas de seguridad presentes hoy en día, sería recomendable desarrollar nuevos mecanismos de reacción que garanticen la continuidad de los servicios o rendimiento del sistema. En contraposición con los actuales esquemas de respuesta que normalmente actúan de manera local, es deseable la propuesta de soluciones de respuesta globales principalmente basadas en la colaboración de toda la red. En otro caso la respuesta podría se ineficaz. Por ejemplo, si se aísla un nodo *dropper* prohibiéndole participar en cualquier comunicación mediante la actuación de sus nodos vecinos, el nodo malicioso podría evitar dicha restricción simplemente moviéndose a otra zona de la red. De manera similar a las propuestas existentes en el contexto de la prevención y detección, este tipo de esquemas se centran en amenazas y objetivos específicos. Así, por un lado, son especialmente eficaces ante los ataques a los que se enfrentan y, por otro, resultan poco útiles para contrarrestar otros tipos.

Otro aspecto importante y que hoy día todavía supone un reto, es la propuesta de diseño e implementación de sistemas integrales de defensa como parte fundamental en la construcción de sistemas con capacidad de supervivencia. Esto es, agrupar de manera conjunta prevención, detección y mecanismos de respuesta de manera que el sistema actúe como un todo permitiendo su adaptación y evolución dinámica (supervisada o no). Esta adaptación global debería converger a soluciones óptimas y estables que a su vez serían cuidadosamente controladas por el mismo sistema. En otras palabras, cada elemento funcional debe estar convenientemente interrelacionado con el resto para conseguir soluciones globales. Por ejemplo, cuando se detecta un nuevo ataque se evalúa dependiendo de su riesgo antes de efectuar la respuesta adecuada y, si es necesario, se podrían instalar nuevos sistemas de prevención o modificar los ya existentes con objeto de evitar amenazas similares en un futuro. Adicionalmente, el modelo usado en el proceso de detección podría ser reconstruido dinámicamente para adaptarse a las condiciones de la red a lo largo del tiempo.

Una de las principales consecuencias de las nuevas tendencias en investigación es la necesidad de colaboración intra- e inter-nodo. Ello implica un nuevo nivel de complejidad y, como consecuencia, un mayor consumo de recursos físicos y lógicos. Ya que tales recursos (por ejemplo, batería o espacio de almacenamiento en disco) son escasos en algunos dispositivos, entornos, y aplicaciones, se hace necesario

llevar a cabo un buen balance entre seguridad y coste implicado. En esta línea, es recomendable el desarrollo de propuestas holísticas de seguridad que también aboguen por el ahorro de recursos. Este compromiso entre seguridad y coste es relevante desde el punto de vista del impacto que tiene sobre la calidad de servicio QoS (*Quality of Service*) de las comunicaciones. Como consecuencia, algunas de las propuestas actuales no son válidas desde un punto de vista práctico ya que obvian aspectos como el consumo de recursos o su impacto real sobre el rendimiento de la red.

Finalmente, se podría pensar que asegurando el sistema mediante la propuesta de soluciones que agrupen más o menos cantidad de los aspectos mencionados anteriormente, garantizaríamos que dicho sistema es "seguro". Pero, ¿qué pasaría si el sistema de seguridad presentase, a su vez, vulnerabilidades de seguridad? Es entonces cuando surge la pregunta siguiente: ¿quién o qué asegura al propio sistema de seguridad? Esta cuestión resulta altamente relevante, ya que vulnerabilidades a explotar en el esquema de seguridad propuesto se podrían traducir en la potenciación del ataque combatido. Quizá un aspecto fácil de resolver en esquemas sencillos, no se antoja tan simple en esquemas y propuestas que involucran un número considerable de características, requisitos, aspectos, etc., como las que se plantean aquí como propuestas de futuro.

2.7. Conclusiones del capítulo

Durante el presente capítulo se han introducido los principales aspectos relacionados con la seguridad en redes ad hoc.

Resulta un tanto irónico el hecho de contar con un tipo de redes cuya flexibilidad, versatilidad y demás características las hacen únicas y prácticamente aplicables a cualquier contexto de uso, pero que al mismo tiempo las condicionan a la exposición de un elevado número de amenazas. De hecho, tal es el número de ataques potenciales que resulta difícil clarificarlos de manera coherente y estructurada, como se ha constatado a lo largo del capítulo. Directamente derivado de lo anterior, se hace necesaria la aportación de soluciones de seguridad que eviten, detecten y respondan ante los ataques que se puedan producir en la red y que afectan al correcto funcionamiento de esta, así como a los servicios que ofrece.

Existe una gran cantidad de soluciones de seguridad propuestas por la comunidad científica en el contexto de redes ad hoc, lo cual, con cierta lógica, está en concordancia con lo vulnerables que son este tipo de entornos. Dentro de tal abanico de propuestas, nos encontramos con soluciones que tratan de prevenir ciertos ataques; otras que, una vez traspasada esta primera línea de defensa, se centran en la detección de actuaciones maliciosas; y por último, otros esquemas que tratan de, una

vez detectado el ataque, reaccionar para recuperar o mantener el rendimiento de la red y los servicios que esta ofrece.

Es concluyente el hecho de que dentro del conjunto de soluciones aportadas, predominan aquellos esquemas que se centran en una sola línea de defensa y ataques concretos. También es destacable el actual auge de los sistemas que modifican en cierta manera el protocolo de encaminamiento, sobre todo para el establecimiento de esquemas de reputación o confianza [20]. Este tipo de soluciones tan específicas, aunque muy eficientes para sus objetivos originales, no lo son ante escenarios distintos, como la presencia de nuevos ataques.

Para el caso concreto de las soluciones reactivas o tolerantes, es conveniente idear esquemas que persigan objetivos más globales. Para ello es necesaria la propuesta de soluciones que aboguen por el mantenimiento, recuperación o mejora de parámetros de rendimiento de la red. Nos referimos, por ejemplo, a la conectividad, el *throughput*, el ratio de entrega de paquetes o PDR (*Packet Delivery Ratio*), etc., los cuales están directa o indirectamente influenciados por la mayoría de ataques y estrechamente ligados a servicios de seguridad.

Para concluir, es sin duda indiscutible la necesidad de la integración e interacción entre las diferentes líneas de defensa como parte de un sistema de seguridad completo. Aún así, son muchas las consideraciones a tener en cuenta para conseguir sistemas robustos y fiables; no solo ante amenazas de seguridad, sino ante fallos o accidentes. Hablamos entonces de sistemas con capacidad de supervivencia. Conseguir este tipo de sistemas es todavía hoy un reto para la comunidad científica cuya dificultad, entendemos, debería abordarse desde el punto de vista de la propuesta de soluciones holísticas que consideren los múltiples aspectos relacionados.

Publicaciones relacionadas

Para finalizar este tema, se presenta a continuación la publicación derivada y relacionada con el ámbito de estudio objeto de discusión. Esta es:

- 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 and J. Lloret (Eds.), CRC Press, pp. 377-400, Abril 2014.

Parte II

Tolerancia y reacción ante amenazas a la seguridad en redes ad hoc

Capítulo 3

Recuperación de datos faltantes en redes de sensores inalámbricas

Contenido

Una red de sensores inalámbrica o WSN está formada por cientos de dispositivos cuyo fin, normalmente, es el de monitorizar una determinada área o región a través de la medición de ciertos parámetros o variables físicas. Escenarios comunes para el despliegue de este tipo de redes son: el militar, el médico y/o el industrial [4, 85, 86]. Típicamente, existe una unidad central o CU (*Central Unit*) encargada de recoger y analizar los datos generados por cada uno de los sensores. Estos datos pueden transmitirse bien directamente hacia la CU o mediante caminos multisalto a través de la colaboración conjunta de varios nodos/sensores. También es útil la ordenación de los sensores en grupos, o *clusters*, dentro de los cuales existirá un sensor central, o

CH (*Cluster Head*), encargado de recoger y agregar los datos provenientes de cada vecino y posteriormente enviarlos a la CU. La agregación de datos reduce el consumo de energía, aspecto muy deseable en este tipo de redes.

Es también común que los sensores mantengan su ubicación a lo largo del tiempo. Dotarlos de capacidad de movimiento introduce ventajas adicionales en términos de conectividad, coste, fiabilidad y eficiencia energética [87].

Para garantizar y robustecer el servicio ofrecido por una WSN, y en general por cualquier red, es deseable el despliegue y uso de mecanismos de seguridad. Estos mecanismos son realmente necesarios en entornos hostiles como acciones militares, gestión de situaciones de crisis, detección y recuperación frente a desastres naturales o de otra índole, etc. En estos contextos de uso, la pérdida o modificación de la información puede tener consecuencias nefastas no solo a nivel de coste material, sino también humano. Por esta razón, dichos sistemas han de ser lo suficientemente robustos como para contrarrestar la pérdida o modificación de información, ya sea malintencionada o no, y abogar así por la supervivencia del sistema [88]. Es importante tener en cuenta que las WSN (y en general las redes ad hoc inalámbricas, como ya se discutió a en el Capítulo 2) son vulnerables ante amenazas de seguridad inherentes a su propia naturaleza. Algunos ejemplos son [89][90]: *packet dropping*, *route poisoning*, *identity spoofing*, *data tampering*, etc. En concreto, el ataque de *data tampering*, *environmental tampering* o simplemente *tampering*, que afecta a la integridad de la información, es especialmente perjudicial en este tipo de entornos. Un ejemplo claro en donde este ataque podría tener consecuencias importantes, en el sentido de pérdidas no solo materiales sino incluso personales, es en la monitorización y control de fuegos en entornos naturales. En este contexto, un ataque de *data tampering* puede provocar que el fuego no sea detectado a tiempo como para poder sofocarlo. Así, un potencial pirómano podría alterar las mediciones obtenidas por varios sensores con el único fin de distraer la atención de las brigadas de bomberos mientras que el verdadero fuego se localiza en otra parte.

En el presente capítulo se evaluará la aplicación de las técnicas de análisis multivariante en este tipo de redes para la monitorización, detección y, sobre todo, la recuperación de datos faltantes como contramedidas ante amenazas de seguridad. Dichas técnicas encajan bastante bien cuando existe alta correlación tanto espacial como temporal entre las variables medidas por la red de sensores, característica común en las WSN. El esquema que se utilizará para la monitorización tiene como objetivo la búsqueda de anomalías en los datos recibidos por la CU. Un vez detectada la anomalía, se procederá al análisis de esta para determinar si fue debida a la propia lectura proveniente de uno o varios sensores o motivada por la pérdida o modificación de la información. Si se detecta la pérdida o modificación de datos (malintencionada o no), entrará en juego el sistema de respuesta que proporcionará una estimación de los datos afectados. Para monitorizar y detectar anomalías sobre el normal comportamiento del sistema se usarán técnicas MSPC (*Multivariate Statistical*

Process Control) basadas en PCA (*Principal Component Analysis*)[91][92] y PLS (*Partial Least Squares*)[93][94]. Por otro lado, para la recuperación de los datos faltantes se usará TSR (*Trimmed Scores Regression*)[95][96], basado en modelos PCA (TSR-PCA) y PLS (TSR-PLS).

Una cuestión importante cuando se utilizan técnicas de análisis multivariante es cómo organizar los datos con los que se está trabajando. Este problema ha sido ampliamente estudiado en campos como la monitorización estadística [97], el control de procesos [98] o el procesado de imagen [99], y puede llegar a tener un impacto importante dependiendo de cuál sea la aplicación final. Se evaluará, así, el impacto del modelado de los datos en la eficiencia de recuperación de datos faltantes, observando cómo el rendimiento del método de imputación varía dependiendo de la utilización de uno u otro modelado.

Adicionalmente y de manera innovadora, se estudiará el impacto sobre el rendimiento del método de recuperación de datos según el algoritmo de *routing* empleado. Así, se analizarán y propondrán diferentes estrategias de *multi-hop routing* y evaluaremos su efecto sobre el rendimiento de la imputación, además de cómo influye la localización del sensor comprometido por el ataque y el modelado de datos utilizado.

El resto del capítulo se estructura de la siguiente manera. En la Sección 3.1 se realiza un estudio sobre las propuestas existentes en la literatura especializada que versan sobre detección de anomalías y, especialmente, sobre imputación de datos faltantes en WSN. A continuación, en la Sección 3.2, se describen los fundamentos de las técnicas y métodos de análisis multivariante. La demostración de la viabilidad y aplicabilidad de dichas técnicas en el contexto de las WSN se presenta en la Sección 3.3. En la Sección 3.4 se expone la visión general y enfoque de la solución propuesta para remediar los efectos de ataques contra la integridad de la información en este tipo de redes. El entorno de simulación utilizado se describe a lo largo de la Sección 3.5. En la Sección 3.6 se introducen y usan los denominados *modelos globales* para la monitorización y detección de anomalías, así como para la recuperación de datos faltantes. En la Sección 3.7 se introduce una primera mejora en el rendimiento de la imputación de datos gracias al empleo de modelos globales y encaminamiento dinámico de manera conjunta. Una segunda mejora en el rendimiento del método de recuperación de datos se presenta en la Sección 3.8. Esta se basa en la aplicación de los denominados *modelos locales* en lugar de modelos globales. Para corroborar la utilidad práctica de nuestra propuesta, en la Sección 3.9 se valida esta en despliegues WSN reales. Por último, las conclusiones de capítulo se exponen en la Sección 3.10.

3.1. Recuperación de datos faltantes y detección de anomalías

Aunque la prevención es importante en la construcción de cualquier esquema de seguridad, son los sistemas de detección y reacción/tolerancia en los que recae la responsabilidad de combatir el ataque una vez superada la primera línea de defensa. En el contexto de la detección de anomalías y recuperación de datos faltantes existen diferentes sistemas y soluciones propuestas por la comunidad investigadora, siendo más abundantes las soluciones, técnicas o sistemas enfocados en la detección de anomalías [100] en WSN, y menor el número de trabajos que proponen esquemas para la recuperación de datos faltantes. Por ejemplo, en [101] se propone un esquema de imputación de datos faltantes y detección de anomalías basado en el empleo de redes neuronales. En primer lugar, la WSN se divide en *clusters*. Acto seguido, el algoritmo diseñado para la imputación de datos escoge el vecino más cercano o el valor más repetido de entre todos sus vecinos, con el fin de estimar el valor faltante del sensor implicado. En caso de no haber vecinos, será el último valor medido el utilizado como imputación. El objetivo principal de esta técnica de recuperación de datos faltantes es mejorar el proceso de clasificación de la red neuronal. Para aumentar la fiabilidad de sistemas de monitorización de pacientes en la medición de constantes vitales o parámetros corporales, el trabajo realizado en [102] propone un esquema distribuido que detecta y aísla aquellos sensores cuyas medidas se perdieron o no son del todo fiables. Gracias a la inherente redundancia en la información presente en estos sistemas, los autores emplean técnicas de redundancia analítica prediciendo, para cada uno de los sensores, un valor virtual. Estos valores virtuales se obtienen de otros sensores con los que existe clara correlación, así como del conocimiento previo adquirido por el sistema. De esta manera, un valor inconsistente será detectado a través de los residuos o errores obtenidos mediante la comparación del valor virtual y el directamente observado.

Los autores del trabajo [103] introducen una metodología basada en el empleo de técnicas de *data mining* para la recuperación de datos faltantes en WSN móviles. Para ello, se divide el área monitorizada en subáreas. En cada una de estas existirá un sensor virtual estático cuya función es monitorizar las lecturas reales de los sensores asociados y computar la media obtenida sobre dichas lecturas. Mediante la comunicación existente entre los sensores virtuales y a través de la explotación de la correlación espacio-temporal de la información obtenida, esta propuesta es capaz de predecir datos faltantes de los sensores reales. Otro estudio [104] propone un método robusto para la imputación de datos faltantes usando tres predictores: dos temporales y uno espacial. El algoritmo selecciona el mejor predictor de entre todos en presencia de datos faltantes, mostrando cómo la frecuencia de muestreo en la recogida de datos y la pérdida de paquetes influyen en la exactitud de la recuperación.

La referencia [105] propone una novedosa técnica de detección de anomalías e imputación de datos faltantes a través del uso de redes bayesianas. Al igual que muchas otras técnicas, este trabajo también se aprovecha de la correlación tanto temporal como espacial entre las sucesivas mediciones. Si existe alguna discrepancia entre el modelo normal (modelo de calibración) y el valor actual del sensor, se lanza una alarma. En cuanto a la recuperación de valores faltantes, esta técnica infiere el valor más probable del sensor afectado a partir del actual y de los inmediatamente anteriores. Como en este último trabajo, los autores en [106] tienen en cuenta la inherente correlación espacial y temporal exhibida en las WSN. Se propone aquí un esquema de recuperación del vecino más cercano o *nearest neighbor missing data imputation* basado en el uso de árboles k-*d*. Estos son construidos considerando varianzas y distancias euclídeas ponderadas que se obtienen a partir de porcentajes de datos faltantes. El algoritmo usa los vecinos más cercanos encontrados en el árbol k-*d* para imputar el valor del sensor perdido. En la referencia [107] se aborda la recuperación de datos faltantes mediante el uso de filtros distribuidos H_∞. Cada sensor se provee con un filtro específico diseñado para mantener un error constante de predicción, así como un determinado rendimiento en base a sus propios valores y los de su vecindario.

Aunque las metodologías multivariantes han sido extensivamente utilizadas en la literatura, su aplicación a redes WSN aún es limitada. Hasta ahora, son pocos los trabajos que hacen uso del análisis multivariante en WSN, estando la mayoría de ellos limitados a la detección de intrusiones o anomalías, mostrándose en cambio escaso interés en su empleo para recuperación de datos faltantes. En [108] se introduce un sistema IDS para la detección de ataques de *routing* mediante PCA. En este esquema la red se organiza en grupos dentro de los cuales existe un nodo monitor. Este nodo construye dos modelos PCA: uno creado a partir de su propio tráfico de red y otro a partir del tráfico global observado. Este último se obtiene a través del intercambio de los modelos PCA locales con los demás monitores. Los autores concluyen que el modelo PCA global distribuido consigue mejores resultados en comparación con su versión centralizada, al menos para el caso concreto de detección de ataques de *sinkhole*. De manera similar, en [109] se propone un sistema de detección de anomalías basado en PCA. Este esquema se compone de dos fases: el modelado de datos y la detección de anomalías. Para la primera fase se discuten un par de métodos que mejoran el modelado PCA en la mejora de la inconsistencia en los datos o *outliers*. Después se realiza la detección de anomalías comparando el modelo previamente calibrado con los nuevos datos recibidos usando la distancia de Mahalanobis.

3.2. Análisis estadístico multivariante

La mayoría de los procesos naturales (o incluso los creados por el hombre) se pueden considerar sistemas multivariantes, ya que para su adecuada caracterización requieren el uso conjunto de varias variables. Por ejemplo, una buena predicción meteorológica necesita del empleo de varios parámetros o variables; entre otras, la velocidad del viento, la presión atmosférica o la temperatura.

La descripción y el modelado de datos, la discriminación y clasificación o la regresión y predicción son algunos de los campos de aplicación para este conjunto de técnicas [110]. En los siguientes apartados se exponen los fundamentos del análisis estadístico multivariante.

3.2.1. Análisis por componentes principales

El objetivo principal del análisis por componentes principales o PCA (*Principal Component Analysis*) es la transformación del conjunto original de variables en un conjunto reducido de nuevas variables descorreladas. PCA identifica una serie de combinaciones lineales de manera que el conjunto resultante recoja la mayor cantidad de información relevante de las variables originales (variabilidad). A estas combinaciones lineales o nuevo conjunto de variables se las denomina *componentes principales* o PC (*Principal Components*). Se podría decir que PCA realiza un cambio de variables desde el espacio original al subespacio de PC. Si $\mathbf{X}$ es una matriz de datos con un número I de observaciones sobre J variables asociadas a cierto experimento, PCA reduce su dimensión a $A \leq J$ abogando porque el subespacio A-dimensional latente capture la mayor variabilidad posible de las variables originales.

Un modelo PCA queda representado por la siguiente ecuación:

$$\mathbf{X} = \mathbf{T}_A \cdot \mathbf{P}_A^T + \mathbf{E}_A \tag{3.1}$$

donde $\mathbf{P}_A$ es la matriz de cargas o *loading matrix*, de dimensión $J \times A$; $\mathbf{T}_A$ es la matriz de puntuaciones o *score matrix*, con tamaño $I \times A$; y $\mathbf{E}_A$ es la matriz de residuos o *residual matrix* que, al igual que la matriz de datos original $\mathbf{X}$, posee un tamaño $I \times J$. A partir de los autovectores de $\mathbf{X}^T \cdot \mathbf{X}$ se obtienen las direcciones de máxima varianza, siendo estos ordenados como columnas de $\mathbf{P}_A$ según la varianza que son capaces de extraer de las variables originales. A la varianza que contempla el modelo se le suele denominar varianza capturada o *explained variance*, siendo $\mathbf{E}_A$ la matriz que contiene el error o la varianza residual, *residual variance*. $\mathbf{E}_A$ aloja aquella variabilidad que no ha sido capturada por el modelo. Para obtener la proyección (*score*) de una nueva observación sobre el subespacio PCA, se aplica la siguiente ecuación:

$$\mathbf{t}_{new} = \mathbf{x}_{new} \cdot \mathbf{P}_A \tag{3.2}$$

donde $\mathbf{x}_{new}$ es un vector de dimensión $1 \times J$ que representa una nueva observación de cada una de las variables. De manera similar, el vector $\mathbf{t}_{new}$, de dimensión $1 \times A$, representa la proyección de la anterior observación dentro del subespacio latente.

Un modelo PCA recoge más o menos variabilidad del conjunto de datos original dependiendo del número de PC utilizado. De esta manera, es importante decidir qué número de PC se ha de utilizar. Para tal fin existen varios métodos posibles. Uno de ellos es la validación cruzada o *cross-validation* [111][112]. La Sección 3.2.6 describirá los entresijos y funcionamiento de este método y su uso para PCA.

3.2.2. Análisis por componentes principales dinámico

Como se mencionó anteriormente, la matriz de *loadings* de un modelo PCA captura la relación existente entre las variables originales. Si cada observación (cada fila de la matriz $\mathbf{X}$) se corresponde con la medición de las variables en un determinado instante temporal, dicho modelado solo es capaz de obtener relaciones estáticas, es decir, no se tiene en cuenta la información o variación temporal que pudieran contener estas variables. Para solventar esta limitación surge DPCA (*Dynamic PCA*) [113], cuyo principal objetivo es incorporar la interdependencia temporal de la información dentro del modelo. DPCA no es más que la aplicación de PCA sobre una reordenación de los datos. El modelado de la dinámica de los datos recogidos de un determinado fenómeno o proceso en cuestión es de interés relevante en diferentes aplicaciones y disciplinas de la ingeniería, tales como el control automático en modelado de sistemas [114].

DPCA extiende la matriz de datos original $\mathbf{X}$ añadiendo observaciones de dichas variables en instantes de muestreo anteriores (*lagged in time*). De esta manera, se obtiene una nueva matriz $\mathbf{X_d}$ incrementada en d decalajes (*lags*) respecto a la anterior. Esto significa que el número de variables en $\mathbf{X_d}$ crece con d, siguiendo esta la expresión:

$$\mathbf{X}_d = \begin{bmatrix} x(1) & x(2) & \cdots & x(d+1) \\ x(2) & x(3) & \cdots & x(d+2) \\ \vdots & \vdots & \ddots & \vdots \\ x(I-d) & x(I-d+1) & \cdots & x(I) \end{bmatrix} \tag{3.3}$$

donde $\mathbf{x}(k) = [x_1(k)\ x_2(k)\ \ldots\ x_J(k)]$ es el vector J-dimensional de observaciones de cada variable en el instante de tiempo k, d denota el número de decalajes temporales

empleado e I el número de observaciones totales por variable. En general, la nueva matriz obtenida tendrá una dimensión de $(I-d)$ filas por $J\cdot(d+1)$ columnas.

La selección del parámetro d está estrechamente ligada a la dinámica de los datos extraídos del fenómeno que se está modelando o estudiando. Así, existen varios métodos en la literatura que estudian la dinámica del fenómeno y que se basan en el uso de gráficos de autocorrelación y de autocorrelación parcial [115].

3.2.3. Mínimos cuadrados parciales

Otro problema de notable interés dentro del análisis multivariante es la regresión de datos, donde dos son los conjuntos de datos implicados: $\mathbf{X}$ e $\mathbf{Y}$. El primero de ellos es una matriz de tamaño $I\times J$ que se corresponde con los datos observados y que se usa para predecir al segundo, $\mathbf{Y}$ $(I\times M)$.

Para estimar $\mathbf{Y}$, se computa el modelo $\mathbf{B}$ continente de la relación de regresión entre los conjuntos de datos $\mathbf{X}$ e $\mathbf{Y}$. De esta manera pueden predecirse nuevos valores de $\mathbf{Y}$ a partir de nuevos valores de $\mathbf{X}$, aplicando el modelado anterior. En general, el problema de regresión que se aborda aquí puede definirse como:

$$\mathbf{Y} = \mathbf{X}\cdot\mathbf{B} + \mathbf{F} \tag{3.4}$$

La solución mínimo cuadrática para (3.4) se define como:

$$\hat{\mathbf{B}} = (\mathbf{X}^T\cdot\mathbf{X})^{-1}\cdot\mathbf{X}^T\cdot\mathbf{Y} \tag{3.5}$$

Se observa en (3.5) las implicaciones sobre el resultado que conlleva la singularidad o el mal condicionamiento de la matriz $\mathbf{X}^T\cdot\mathbf{X}$. La singularidad de $\mathbf{X}^T\cdot\mathbf{X}$ implica que no se puede calcular su inversa ni, por tanto, $\hat{\mathbf{B}}$. De igual manera, el cálculo de $\hat{\mathbf{B}}$ es impreciso e inestable cuando $\mathbf{X}^T\cdot\mathbf{X}$ está mal condicionada, lo que ocurre si las variables del conjunto de datos $\mathbf{X}$ están correladas. Para solucionar esta limitación, el método de mínimos cuadrados parciales o PLS (*Partial Least Squares*) aplica la idea que subyace de PCA: utilizar un subespacio de variables latentes en el problema de regresión. En este caso, las variables en $\mathbf{X}$ se transforman en un conjunto reducido de variables latentes tal que maximicen la covarianza entre $\mathbf{X}$ e $\mathbf{Y}$.

El problema de regresión lineal parcial para matrices normalizadas, $\mathbf{X}$ e $\mathbf{Y}$ en este caso, se puede establecer como:

$$\mathbf{X} = \mathbf{T}_A\cdot\mathbf{P}_A^T + \mathbf{E}_A \tag{3.6}$$

$$\mathbf{Y} = \mathbf{T}_A \cdot \mathbf{Q}_A^T + \mathbf{F}_A \tag{3.7}$$

donde $\mathbf{T}_A$ ($I \times A$) se corresponde con la matriz de *scores*, $\mathbf{P}_A$ ($J \times A$) y $\mathbf{Q}_A$ ($M \times A$) son las matrices de *loadings*, respectivamente. Por último, $\mathbf{E}_A$ ($I \times J$) y $\mathbf{F}_A$ ($I \times M$) son las matrices de error o residuos correspondientes a $\mathbf{X}$ e $\mathbf{Y}$, respectivamente.

Los coeficientes de regresión del modelo PLS se establecen como:

$$\hat{\mathbf{B}}_{PLS} = \mathbf{W}_A \cdot (\mathbf{P}_A^T \cdot \mathbf{W}_A)^{-1} \cdot \mathbf{Q}_A^T \tag{3.8}$$

donde $\mathbf{W}_A$ ($J \times A$) es la matriz de pesos, tal que $\mathbf{T}_A = \mathbf{X} \cdot \mathbf{W}_A \cdot (\mathbf{P}_A^T \cdot \mathbf{W}_A)^{-1}$. Por lo tanto, un modelo PLS quedaría representado mediante la terna de matrices $\mathbf{P}_A$, $\mathbf{W}_A$ y $\mathbf{Q}_A$.

Finalmente, una nueva salida correspondiente a una nueva observación se puede estimar con PLS simplemente aplicando dicho modelo de regresión de la siguiente manera:

$$\hat{\mathbf{y}}_{new} = \mathbf{x}_{new} \cdot \hat{\mathbf{B}}_{PLS} \tag{3.9}$$

donde $\mathbf{x}_{new}$ ($1 \times J$) es el vector que representa una nueva observación para cada una de las variables de entrada tras la normalización pertinente. Por otro lado, $\hat{\mathbf{y}}_{new}$ ($1 \times M$) se corresponde con la estimación producida normalizada.

El número de dimensiones del subespacio de variables latentes (A), puede estimarse también a través del método de validación cruzada, de manera similar al caso de PCA. Este método de selección de LV (*Latent Variable*) se describirá en detalle en la Sección 3.2.6.

3.2.4. Monitorización multivariante

Una de las aplicaciones mas extendida de PCA y PLS es la monitorización de procesos y detección de anomalías, frecuentemente llamado en su conjunto como MSPC (*Multivariate Statistical Process Control*). En un sistema MSPC son comúnmente utilizados los estadísticos Q y T^2 [116]. Q comprime los residuos de cada observación, siendo T^2 obtenido a partir de los *scores* del modelo. Los estadísticos calculados a partir de los datos de calibración y en condiciones normales se utilizan para establecer límites de control para un cierto nivel de confianza. De esta manera, nuevos datos provenientes del mismo proceso son susceptibles de ser anómalos si estas fronteras se rebasan. A través de los denominados *gráficos de contribución* es posible observar qué aporta cada variable a una posible anomalía detectada en el sistema [117], pudiendo determinar así el origen de esta desviación.

Para una determinada observación, los estadísticos Q y T^2 se obtienen de la siguiente manera:

$$T_i^2 = \sum_{a=1}^{A} \left(\frac{\tau_{ai} - \mu_a}{\sigma_a} \right)^2 \tag{3.10}$$

$$Q_i = \sum_{j=1}^{J} \left(e_{ij} \right)^2 \tag{3.11}$$

donde τ_{ai} representa el *score* de la observación i-ésima de la a-ésima variable latente. A su vez, μ_a y σ_a simbolizan la media y la desviación estándar de los *scores* de dicha variable, respectivamente, obtenidas de los datos de calibración. Por último, e_{ij} se entiende como el valor residual correspondiente a la i-ésima observación de la j-ésima variable.

3.2.5. Imputación multivariante de datos faltantes

Dentro de la variedad de técnicas y soluciones que se proponen hoy en día en el contexto de la imputación de datos faltantes con modelos multivariantes como PCA y PLS, podemos distinguir los métodos basados en regresión de los de imputación directa. En el trabajo [95], los autores concluyen que las técnicas de recuperación de datos faltantes basadas en regresión ofrecen un mejor rendimiento. De entre todas las técnicas de regresión, el método TSR (*Trimmed Scores Regression*) ofrece un balance equilibrado entre simplicidad y rendimiento [96]. TSR estima el valor de los *scores* desde los *trimmed scores*, es decir, los *scores* obtenidos al insertar ceros en sustitución de los datos faltantes. El hecho de rellenar con ceros aquellos datos que se han perdido o no están disponibles, equivale a establecer como estimación inicial la media incondicional de los datos siempre que previamente estén centrados. Hemos de señalar que un conjunto de datos se considera centrado cuando la media de cada variable es cero.

Sin pérdida de generalidad, se considera una observación incompleta x_{inc} que contiene un número de k medidas disponibles, tratando el resto como faltantes. A través del uso de modelado PCA se pueden calcular los *trimmed scores* de x_{inc} de la siguiente manera:

$$\tau_A^* = \left(\mathbf{P}_{A,k}^* \right)^T \cdot x_{inc}^* \tag{3.12}$$

donde

$$\mathbf{P}^*_{A,k} = \begin{bmatrix} p_{1,1} & \cdots & p_{A,1} \\ \vdots & \ddots & \vdots \\ p_{1,k} & \cdots & p_{A,k} \end{bmatrix} \tag{3.13}$$

$$x^*_{inc} = [x_1, \ldots, x_k]^T \tag{3.14}$$

siendo $p_{a,j}$ el *loading* correspondiente a la j-ésima variable de la a-ésima PC. Solo las variables disponibles en x_{inc} y sus correspondientes *loadings* se usarán para calcular los *trimmed scores*.

Con objeto de mejorar la estimación de los *scores* en presencia de observaciones incompletas, se utiliza el conjunto de calibración $\mathbf{X}$. Llamemos $\mathbf{X}^*$ a aquella submatriz de $\mathbf{X}$ que contempla únicamente las variables disponibles de x_{inc}. Con esto, la matriz de *trimmed scores* correspondiente a los datos de calibración se puede calcular de la siguiente manera:

$$\mathbf{T}^*_A = \mathbf{X}^* \cdot \mathbf{P}^*_A \tag{3.15}$$

La matriz completa de *scores* $\mathbf{T}_A$ se obtiene a partir de $\mathbf{T}^*_A$ aplicando el siguiente modelo de regresión:

$$\mathbf{T}_A = \mathbf{T}^*_A \cdot \mathbf{B} + \mathbf{F} \tag{3.16}$$

donde la matriz de los coeficientes de regresión $\mathbf{B}$ se construye, por ejemplo, aplicando mínimos cuadrados dado que la matriz $(\mathbf{T}^*_A)^T \cdot \mathbf{T}^*_A$ está típicamente bien condicionada. En caso contrario, sería adecuada la utilización de PLS u otros métodos sesgados. Una vez calculado $\mathbf{B}$, su uso mejorará la estimación del *score* de la observación incompleta como sigue:

$$\tau^{TSR}_A = (\mathbf{P}^*_A \cdot \mathbf{B})^T \cdot x^*_{inc} \tag{3.17}$$

Finalmente, la estimación de la predicción incompleta se obtiene de aplicar τ^{TSR}_A como sigue:

$$\hat{x} = \mathbf{P}_A \cdot \tau^{TSR}_A \tag{3.18}$$

Un propiedad característica de TSR y en general de la imputación con PCA, es su incremento de rendimiento cuanto mayor sea la correlación entre variables del conjunto de datos. Con variables muy relacionadas entre sí, aquellos valores perdidos

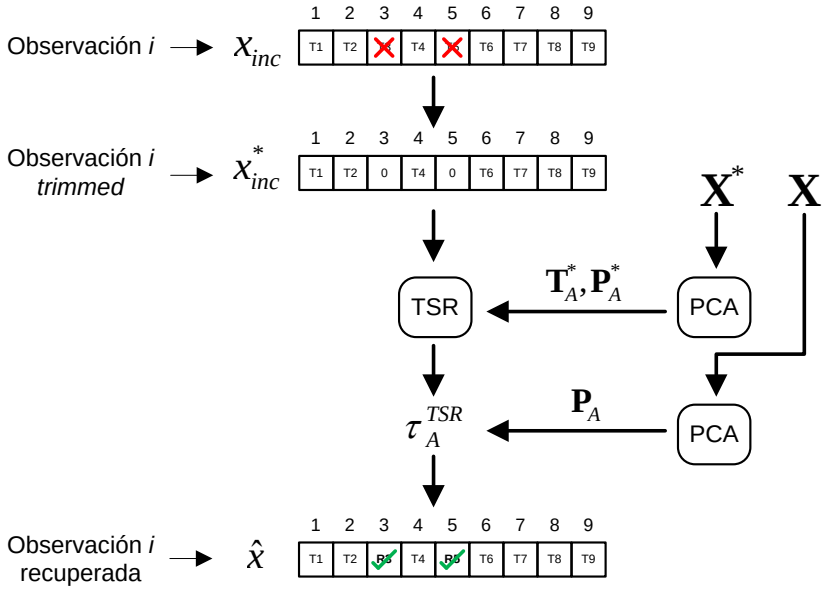


Figura 3.1: Esquema ilustrativo que presenta los principales pasos implicados en el proceso de recuperación de datos faltantes TSR-PCA.

en una determinada observación se pueden recuperar a partir de los valores de otras variables del conjunto.

Con el fin de clarificar la filosofía de trabajo de TSR, en la Figura 3.1 se ilustra de manera gráfica dicho procedimiento. Aunque para el ejemplo se utilizan modelos PCA (TSR-PCA), el uso de modelos PLS (TSR-PLS) no cambiaría la metodología general propuesta. Primeramente, el procedimiento de recuperación se activa cuando se detecta alguna observación alterada o incompleta. Una vez detectada la anomalía, es un sistema de monitorización supervisado o no el que se encargará de determinar qué datos de la observación han sido alterados (este procedimiento se detalla en la Sección 3.6.2). Aunque el proceso es en sí mismo autoexplicativo (T3 y T5 son los valores perdidos, siendo recuperados y sustituidos por los valores R3 y R5, respectivamente), dos son los principales aspectos que han de ser remarcados aquí: (*i*) el método de imputación solo considera la información disponible para estimar los *scores*, y (*ii*) el sistema es capaz de estimar la observación original considerando y aplicando el modelo de calibración completo PCA.

Hasta este punto, TSR se construye con modelos PCA (TSR-PCA). De igual manera se pueden emplear modelos PLS (TSR-PLS). Cuando se emplea PLS, los *trimmed scores* de x_{inc} en (3.12) se calculan siguiendo [118]:

$$\tau^*_A = (\mathbf{R}^*_A)^T \cdot x^*_{inc} \tag{3.19}$$

donde

$$\mathbf{R}_A^* = (\mathbf{W}_{A,k}^*) \cdot (\mathbf{P}_A^T \cdot \mathbf{W}_A)^{-1} \tag{3.20}$$

Hemos de recordar que $\mathbf{W}_A$ es la matriz de pesos y $\mathbf{P}_A$ la matriz de *loadings* del modelo PLS, considerando un número A de variables latentes. En su lugar, $\mathbf{W}_{A,k}^*$ se puede expresar como:

$$\mathbf{W}_{A,k}^* = \begin{bmatrix} w_{1,1} & \cdots & w_{A,1} \\ \vdots & \ddots & \vdots \\ w_{1,k} & \cdots & w_{A,k} \end{bmatrix} \tag{3.21}$$

donde $w_{a,j}$ es el peso correspondiente a la j-ésima variable de la a-ésima variable latente. $\mathbf{P}_A$ y $\mathbf{W}_A$ se usan para la inversión en (3.20). Nótese que ambas matrices se construyen solo con información disponible; es decir, están completas. Consecuencia directa del uso de toda la información del modelo al completo es la mejora de la capacidad de predicción. También relevante a efectos prácticos, es de señalar que adicionalmente se evitan problemas en la inversión presente en (3.20).

3.2.6. Selección del número de variables latentes

Encontrar el número óptimo (en algún sentido) o al menos un valor aproximado para el número de variables latentes en PCA, no es una tarea sencilla. Esto es especialmente difícil cuando se trabaja con conjuntos de datos de calibración pequeños provenientes de experimentos costosos de realizar y para los que generar un conjunto de validación adecuado no es viable desde un punto de vista práctico [119].

PCA es una herramienta muy versátil y aplicable en multitud de escenarios y problemas. A su vez, dependiendo de la aplicación en la que se utilice, la determinación del número de PC también ha de resolverse de diferente manera. Desde un punto de vista más teórico, se podría decir que este número ha de ser seleccionado para que maximice una determinada función objetivo que, lógicamente, dependerá del problema en cuestión. Así, si el problema acometido cambia, este número también debería cambiar potencialmente. Consecuentemente, establecer una única vía general para calcular el número de PC es un objetivo mal definido [120].

En concreto, en el presente capítulo distinguiremos dos ámbitos diferentes de aplicación para el modelado PCA: monitorización de anomalías e imputación de datos faltantes. Aunque son varios los métodos propuestos en la literatura que abordan la obtención del número de PC para estas aplicaciones, son dos los que, por su amplio uso, utilizaremos aquí: (*i*) el análisis de la varianza capturada por el

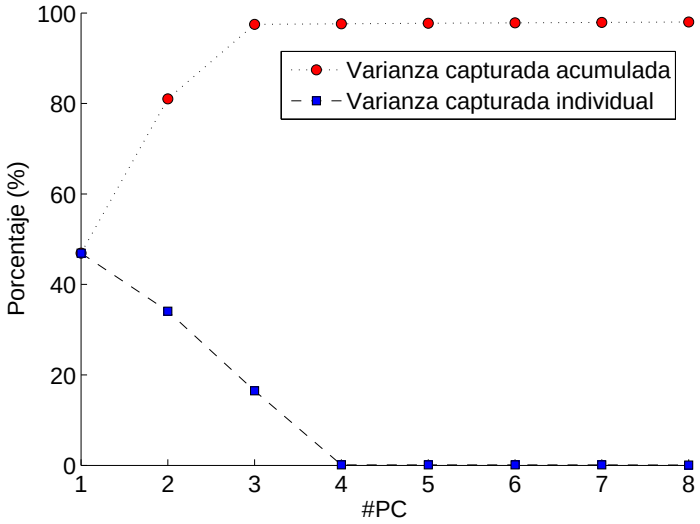


Figura 3.2: Varianza capturada por el modelo PCA obtenida a medida que el número de PC se incrementa. Adicionalmente, se muestra la varianza capturada acumulada.

modelo en el contexto de la monitorización de anomalías, y (*ii*) la validación cruzada aplicada a la imputación de datos faltantes.

Análisis de la varianza capturada por el modelo

Este método consiste en añadir PC hasta la obtención de un cierto nivel de varianza capturada por el modelo. Los valores de varianza capturada obtenidos por cada PC se trazan sobre un gráfico denominado *scree plot*, que muestra la evolución de dicho parámetro conforme aumenta el número de componentes principales considerado. En estos gráficos y de manera subjetiva, se buscará aquel número de PC que contempla un nivel de varianza significativo con respecto a lo que añaden subsiguientes PC. A modo de ejemplo y utilizando el conjunto de datos de calibración elaborado para el sistema de monitorización (ver Sección 3.6), en la Figura 3.2 podemos ver la evolución de la varianza capturada a medida que aumenta el número de PC. A su vez, de utilidad es, la representación de la varianza capturada acumulada. De acuerdo a dicha varianza capturada, podemos ver que a partir de 3 PC la adición de varianza es poco significativa, lo que llevaría a determinar dicho número.

Por lo tanto, para este ejemplo en concreto, el número adecuado de PC a seleccionar sería 3.

Validación cruzada

La habilidad de predicción de un modelo está estrechamente relacionada con su capacidad para estimar nuevos datos u observaciones no tenidas en cuenta durante la fase de calibración o entrenamiento. La manera habitual de evaluar la eficacia o rendimiento de predicción de un modelo es a través del uso de conjuntos de datos de validación [110]. Uno de los métodos que se utiliza para evaluar el rendimiento de un modelo de predicción es la validación cruzada, siendo especialmente recomendado cuando el número de observaciones disponibles es pequeño para dividir los datos en sendos conjuntos de calibración y validación.

La validación cruzada divide las observaciones disponibles en un número determinado de grupos, G, para después calcular el error de predicción que se comete en cada uno de ellos. Para toda iteración del procedimiento, se obtiene un modelo de calibración a partir de los $G-1$ grupos, mientras que el grupo restante es utilizado como conjunto de validación. A continuación se estima la salida y se computa y almacena el error de predicción obtenido. Este proceso se repite para todos los grupos, siendo el error total cometido la combinación de todos los errores obtenidos.

Wold [111] propuso el uso de la validación cruzada para determinar el número de PC en PCA. El error de predicción (típicamente PRESS (*Prediction Error Sum of Squares*)) que se obtiene en la validación cruzada se computa para diferente número de PC: una, dos y así sucesivamente. De esta manera, el número de PC a utilizar se selecciona inspeccionando la evolución de la curva PRESS.

La validación cruzada se puede aplicar directamente a modelos PLS. No obstante, esto se antoja difícil para modelos PCA debido a que la noción de error de predicción no existe como tal: los modelos PCA no son modelos de predicción per se. En [121] los autores concluyen que el algoritmo *ekf (element-wise k-fold)* es una opción válida para efectuar validación cruzada con modelos PCA, siempre que el modelo se use para imputación de datos faltantes, objetivo principal del presente capítulo. En el Apéndice A se introduce y describe someramente dicho algoritmo.

3.3. Viabilidad y aplicabilidad de las técnicas multivariantes en WSN

Como ya se introdujo en este capítulo, una WSN está compuesta por un conjunto de sensores estratégicamente distribuidos y cuyo objetivo principal es medir y recoger información de utilidad sobre un determinado entorno cada cierto tiempo predefinido. Estos datos son luego procesados dependiendo del fin del sistema en su conjunto; la monitorización de un área forestal sería un claro ejemplo de aplicación.

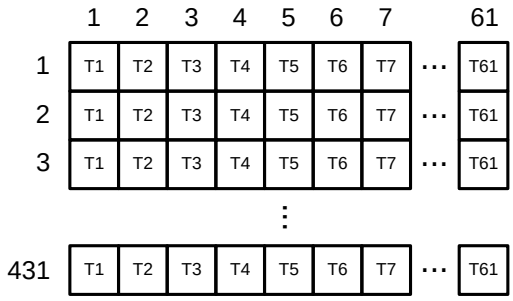


Figura 3.3: Ejemplo de estructuración del conjunto de datos LUCE, **X**, formado por $I = 431$ observaciones de $J = 61$ variables cada una.

La información obtenida por la WSN puede ser contemplada de manera directa como un conjunto de datos **X** en el que cada sensor se correspondería con una variable j, obteniéndose una observación i cada cierto tiempo de muestreo. Atendiendo a lo ya introducido en las secciones anteriores sobre el análisis multivariante, parece claro que dichas técnicas son perfectamente aplicables a entornos y escenarios de este tipo. Además, estos métodos son especialmente recomendables cuando la información a procesar o modelar está altamente correlada, característica común en WSN. Ahondando un poco más en esta justificación, y sin ánimo de entrar en mucho detalle, se justifica a continuación la aplicabilidad y viabilidad de este tipo de análisis en WSN. Para ello se ha escogido un conjunto de datos proveniente del despliegue WSN real correspondiente al proyecto LUCE (*Laussanne Urban Canopy Experiment*) [122]. Aunque más adelante (ver Sección 3.9) se describe en detalle este proyecto y la estructura y organización del conjunto de datos utilizado, en la Figura 3.3 se muestra con fines ilustrativos cómo se han estructurado los datos para conformar el conjunto de datos de calibración **X**. Se observa en la figura cómo la información se estructura en 61 variables o sensores y 431 observaciones. Cabe notar que la información contenida en **X** se corresponde con la temperatura medida por un determinado sensor T_j para cada una de las observaciones. El mínimo valor de correlación obtenido de entre todas las variables fue $0,89$, corroborándose así la existencia de una alta relación entre las variables del conjunto. Otro sencillo experimento que apoya aún más la idoneidad del uso de estas técnicas en redes WSN se muestra en la Figura 3.4. Se observa en la figura la varianza residual obtenida de PCA en función del número de PC utilizado. Una consecuencia directa de la alta correlación entre variables es que el modelo PCA es capaz de recoger prácticamente toda la información de variabilidad con una sola variable latente. En concreto, se recoge el 97 % de la varianza contemplada, siendo la varianza residual solo del 3 %.

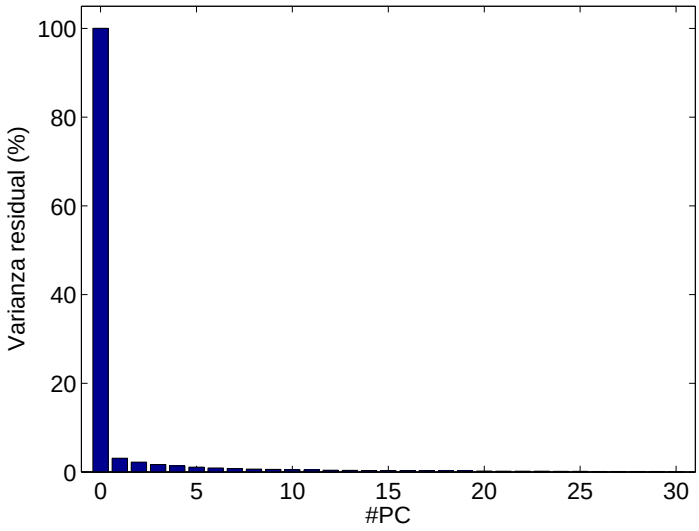


Figura 3.4: Varianza residual del modelo PCA obtenida a medida que el número de PC se incrementa.

3.4. Visión general y enfoque de la solución propuesta

Aunque son varios los escenarios de uso en donde sistemas de monitorización, detección y respuesta tienen cabida, estos esquemas son especialmente aconsejados para entornos críticos como acciones militares, gestión de crisis o recuperación ante desastres [86]. En particular, en el contexto de la extinción de incendios forestales, entorno objetivo sobre el que se aplicarán las técnicas aquí propuestas. Es de destacar la relevancia del estudio en este tipo de escenarios, debido al impacto tanto social como económico que un incendio causa en la región afectada. Mediante la simulación de dicho entorno se podrán recrear condiciones normales de temperatura así como situaciones con fuego. También se simularán ataques a la integridad de los datos como el ya mencionado ataque de *data tampering*. Dentro de este escenario se aplicará un sistema de detección de anomalías basado en el uso de técnicas multivariantes que alertará a un supervisor humano ante la ocurrencia de un evento anómalo. Dicho supervisor se encargará de clarificar la proveniencia de dicha anomalía, discerniendo entre si se está ante una situación de fuego real o, por el contrario, ante una actuación maliciosa. Para ello, esta persona se apoyará en el uso de gráficos y visualizaciones integrados en el sistema de monitorización. En caso de que se dictamine la presencia de un ataque, se iniciará el subsecuente proceso de recuperación para restaurar el valor del sensor o sensores afectados por el ataque. En la Figura 3.5 se observan los módulos del sistema completo y su relación, así como en qué módulos existe intervención humana. Hemos de destacar que el objetivo principal de la solución contemplada aquí se centra en la respuesta ante ataques de *data tampering* mediante el empleo de técnicas de análisis multivariantes empleadas para la recuperación de

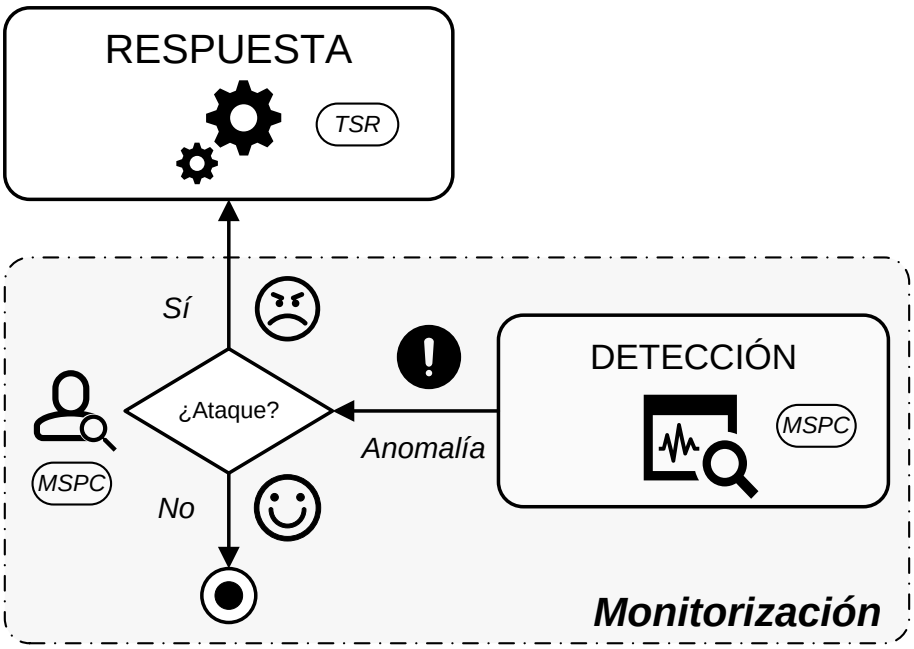


Figura 3.5: Esquema ilustrativo que presenta los principales módulos que han de intervenir para contribuir a la detección de anomalías y respuesta ante ataques en WSN.

datos faltantes. Aunque se expondrá una posible solución de detección automatizada para este tipo de ataques, que en conjunción con la monitorización empleada determinarán la existencia o no de dicho ataque, este no es el foco principal del capítulo. Por otro lado, es de señalar la necesaria presencia de todos y cada uno de los módulos expuestos en la Figura 3.5 para configurar un sistema completo y resistente que luche contra ataques a la seguridad.

A lo largo de las secciones venideras se analizarán los efectos de un hipotético ataque sobre el rendimiento del esquema de recuperación propuesto y su estrecha relación con el algoritmo de encaminamiento subyacente. Para validar la solución se efectuarán una serie de experimentos tanto es escenarios simulados como en entornos reales.

3.5. Descripción del entorno de simulación

Son evidentes las ventajas de utilizar entornos y herramientas de simulación, no solo para el objetivo que se presenta aquí, sino para cualquier esquema o escenario en general. Entre otros aspectos, se puede destacar el hecho de que mediante la simulación se ahorra notablemente en costes de despliegue, se agiliza el desarrollo y pruebas de la solución y se determina la posible viabilidad de aplicación de la solución en entornos reales.

Hoy en día existen bastantes herramientas útiles para simular WSN [123]. No obstante, la mayoría de ellas están enfocadas a la recreación de características propias de la red (por ejemplo, aspectos físicos de las transmisiones, protocolos, modelos de energía, entre otros), ignorando aspectos relacionados con el entorno como pueden ser la simulación de magnitudes físicas objeto de medición de los sensores desplegados. Por este motivo, se decide desarrollar un simulador específico basado en Matlab 2009b, recreando la evolución de la temperatura dentro de un área forestal. Esta herramienta se inspira en el trabajo realizado en [124], donde los autores idean un modelo en el que la temperatura obtenida por un determinado sensor puede ser calculada a través de la contribución conjunta de focos de fuego cercanos. Para modelar un foco de fuego se utilizan distribuciones gaussianas 2D; aunque en el presente caso se emplearán también para simular situaciones normales de temperatura. La Figura 3.6 muestra el escenario de simulación utilizado, en donde se observa la distribución de los sensores en el área monitorizada sobre la Figura 3.6(a). Además, son dos los tipos de mapas de temperatura mostrados en la figura. El primero de ellos, representado en la Figura 3.6(b), muestra tres fuentes diferentes de temperatura (en $^{\circ}C$). Estas simulan zonas más templadas del área, emulando posibles valles y lugares más fríos, que perfectamente pueden corresponder a montañas, zonas elevadas o de umbría. En el segundo mapa de temperatura, representado en a Figura 3.6(c), se puede observar cómo un foco central de fuego ha ido creciendo hasta quemar gran parte del área.

Se asume un área forestal cuadrada de 1000 m × 1000 m a monitorizar, para lo cual se distribuyen 81 (9×9) sensores de manera regular, localizándose cada uno de ellos a una distancia aproximada de 100 m con respecto a su vecinos (ver Figura 3.6(a)). Cada cierto tiempo de muestreo, todo sensor perteneciente a la red mide la temperatura ambiente, que luego es enviada hacia la CU. Esta estructuración se basa en el despliegue real que proporciona la empresa Libelium[1]. Atendiendo a la Figura 3.6(a) la CU se situará a la derecha de la malla de sensores.

Con la herramienta de simulación desarrollada, se generará un conjunto de datos cuyo fin es la calibración del modelo PCA, en adelante referenciado como CAL (*CALibration dataset*). Para este conjunto de datos, la matriz $\mathbf{X}$ contendrá 100 observaciones de 81 variables (la temperatura obtenida por cada sensor) obtenidas en condiciones normales, es decir, sin la actuación del fuego. Después se simula la acción y evolución del fuego obteniéndose un nuevo conjunto de datos, en adelante denominado FIR (*FIRe dataset*). Ambos conjuntos de datos, CAL y FIR, serán utilizados para evaluar la capacidad de detección de nuestro sistema ante eventos o anomalías no contempladas.

[1] http://www.libelium.com/wireless_sensor_networks_to_detec_forest_fires/

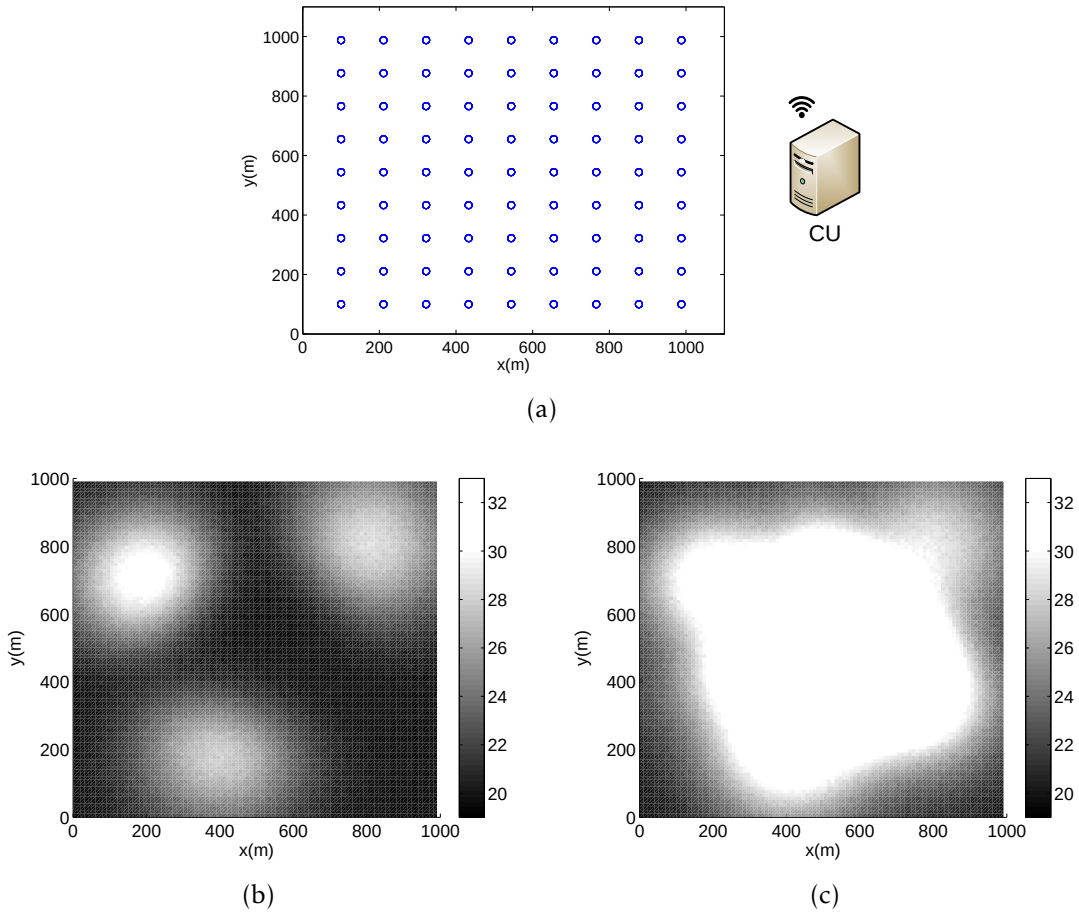


Figura 3.6: Escenario de simulación: (a) distribución de los sensores, (b) mapa de temperatura en condiciones normales y (c) mapa de temperatura con la presencia de un foco de fuego en avanzado estado.

3.5.1. Estrategias de encaminamiento

El diseño o selección de la estrategia de encaminamiento en cualquier red, y en concreto para el contexto específico en el que se enmarca el presente capítulo, las WSN, está estrechamente ligado a aspectos como la arquitectura de la red, restricciones inherentes a escenarios WSN (por ejemplo, el consumo energético, capacidad de procesamiento, ancho de banda, entre otros) y, sobre todo, la aplicación objetivo del despliegue. Adicionalmente, parámetros como la escalabilidad, conectividad, cobertura, calidad de servicio o tolerancia a fallos, que pueden ser vistos como un conjunto de requisitos técnicos y funcionales del despliegue, influyen directamente sobre el tipo o método de *routing* empleado [85].

En el contexto de la seguridad, la elección de uno u otro esquema podría comprometer la prestación del servicio objetivo del despliegue WSN. Concretamente, el

ataque *data tampering*, como ya se justificó en la introducción del capítulo, puede tener consecuencias importantes no solo materiales sino personales en este tipo de entornos y dependiendo del contexto en el que se usen. Considerando esquemas de *multi-hop routing* muy utilizados en WSN, comprometer uno u otro sensor puede afectar en mayor o menor medida al rendimiento de la red. De esta manera, serán situaciones distintas aquellas en las que se ataca un sensor encargado de reenviar la información proveniente de un grupo de sus semejantes hacia la CU o, por el contrario, aquellas en las que el dispositivo comprometido es el que envía directamente su información hacia la CU. Claramente, se puede esperar que en el primer caso los efectos del ataque tengan mayores consecuencias que en el segundo caso, ya que la información comprometida es también mayor.

Para corroborar la hipótesis anterior se utilizan diferentes estrategias de encaminamiento con el fin de evaluar la incidencia de dichas estrategias no solo en el servicio ofrecido por la red sino, principalmente, de cara a la solución de respuesta ante intrusiones propuesta basada en la recuperación de datos faltantes. De esta manera, clasificaremos estos diferentes esquemas como estrategias estáticas, en las que la información se encamina siempre por las mismas rutas (obviamente, una vez superada la fase de descubrimiento de rutas y para una red de topología estática) a lo largo del tiempo.

Dentro de estos algoritmos, nos basaremos en dos en concreto:

- MCFA (*Minimum Cost Forwarding Algorithm*) [85]. Explota el conocimiento previo de la dirección de encaminamiento multisalto, normalmente hacia la CU. De esta manera, en un esquema en donde la CU se localiza en dirección Este, los sensores enviarán y/o reenviarán la información en esa dirección.
- LEACH (*Low Energy Adaptive Clustering Hierarchy*) [85]. Se diseñó para reducir el consumo de energía de los sensores y, por ende, de la red al completo. Este esquema organiza los sensores en *clusters* o grupos, tal que existirá un sensor llamado CH que recogerá y agregará todas las medidas provenientes del grupo para remitirlas luego a la CU.

El encaminamiento y su efecto ante ataques de *data tampering*

Para la evaluación de las capacidades del modelo PCA en lo que se refiere a la detección y recuperación ante ataques de *data tampering*, se simularán tres escenarios diferentes. Su principal diferencia radica en el algoritmo de *routing* utilizado para el envío de la información de cada sensor hacia la CU. A lo largo de esta sección se consideran las filosofías de encaminamiento estático que se introdujeron previamente.

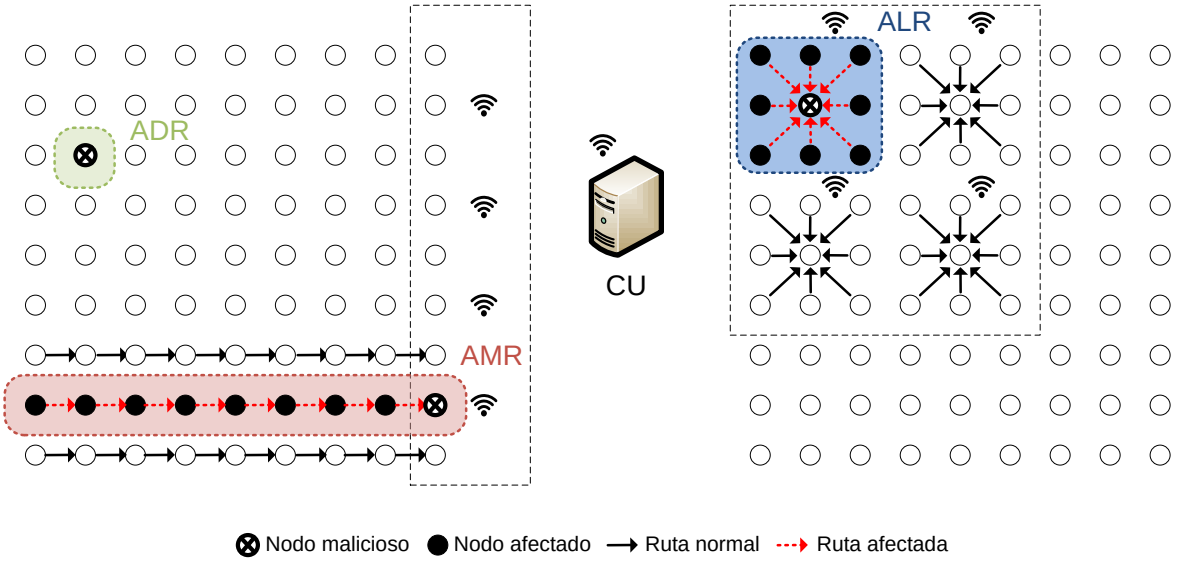


Figura 3.7: Diferentes escenarios de ataque considerados.

En primer lugar se establece una comunicación directa entre cada sensor y la CU. En este caso no tiene cabida ningún proceso de encaminamiento, siendo la información transmitida mediante el uso de, por ejemplo, tecnología GPRS (*General Packet Radio Service*). Aquí, el ataque solo tiene efectos sobre las medidas del sensor atacado. Esta situación será denominada en adelante ADR (*Attack on Direct Routing*). En la parte superior izquierda de la Figura 3.7 se ilustra esta situación. En segundo lugar se utiliza un esquema de *multi-hop routing*. Aquí no solo se verán afectadas las medidas del sensor comprometido, sino también aquellas medidas que se encaminen a través de dicho sensor. En la parte inferior izquierda de la Figura 3.7 se muestra cómo utilizando un esquema de *routing* lineal, inspirado en el protocolo MCFA (la información se encamina de izquierda a derecha), el peor caso a considerar sería atacar al sensor más a la derecha, encargado de enviar toda la información que pasa por él hacia la CU. Este hecho implica que toda la información reenviada por dicho sensor también está comprometida. En adelante, a esta situación se la denominará AMR (*Attack on MCFA Routing*). Considerando esquemas de *routing* más sofisticados se evaluará también la estrategia LEACH. A la derecha de la Figura 3.7 se muestra el escenario asociado, en donde se observa cómo comprometiendo el sensor CH, también lo estará toda la información del grupo. De manera similar a las anteriores situaciones, de aquí en adelante denominaremos a esta situación como ALR (*Attack on LEACH Routing*).

En este punto es necesario aclarar que, aunque las denominaciones de ataque anteriores se usan con el fin de facilitar su uso y entendimiento, en todos los casos es uno y solo uno el sensor directamente atacado. Así, estas situaciones no deben ser entendidas como diferentes variantes del ataque de *data tampering*, sino como las

situaciones acontecidas y consecuencias del mismo ataque debido a los diferentes esquemas de *routing* utilizados.

Para cada uno de los escenarios o situaciones de *data tampering* se genera un conjunto de datos diferente, en adelante denominados conjunto de datos ATT (*ATTack test dataset*). Además de la acción de los ataques, también interviene el fuego con un crecimiento progresivo e incremental con el tiempo de simulación.

A partir de los conjuntos de datos generados (CAL, FIR y ATT) se evaluará la capacidad del sistema para: (*i*) determinar la ocurrencia de ataques de *data tampering*, y (*ii*) recuperar los datos modificados por el ataque con el fin de restaurar el funcionamiento normal del entorno, permitiendo que las brigadas de extinción de incendios puedan atacar el verdadero fuego posicionándose adecuadamente.

3.6. Estructuración y organización de los datos: modelo global

En general, las técnicas de análisis multivariante son efectivas en diferentes y heterogéneos contextos, siendo aspectos relevantes tanto la organización de los datos para la construcción del modelo como la elección o selección del número de variables latentes. El problema de la estructuración de la información en el modelado multivariante ha sido tratado en un número considerable de trabajos, abarcando aplicaciones diversas tales como la monitorización estadística [97], el control de procesos [98] o el procesamiento de imagen [99]. Por ejemplo, DPCA, que ha alcanzado un gran interés por parte de la comunidad científica, se basa simplemente en un proceso de reorganización de los datos seguido del tradicional modelado PCA, como ya se introdujo en la Sección 3.2.2. Las referencias anteriores concluyen que la problemática de la estructuración y organización de los datos cobra un papel importante a la hora de incorporar aspectos como el dinamismo, localidad y/o segmentación en los datos en el modelado multivariante. También es aceptado el hecho de que una organización óptima es altamente dependiente de la aplicación objetivo [125]. En concreto, y como será evidenciado a través de la actual sección y a lo largo de las Secciones 3.7 y 3.8 más adelante en este capítulo, nos centraremos en la aplicación de técnicas de análisis multivariante en dos contextos específicos: la detección e identificación de anomalías y la recuperación de datos perdidos o alterados. Atendiendo a la motivación anterior, y puesto que se trabajará en escenarios diferentes, con requisitos y condiciones particulares diferentes, se hace necesaria la propuesta de esquemas de organización de la información también diferentes. A lo largo del capítulo se discutirán los modelos creados para ambos fines. Distinguiremos entonces: *modelos globales*, para propósitos de detección de anomalías; *modelos dinámicos globales*, para la recuperación de datos faltantes; y

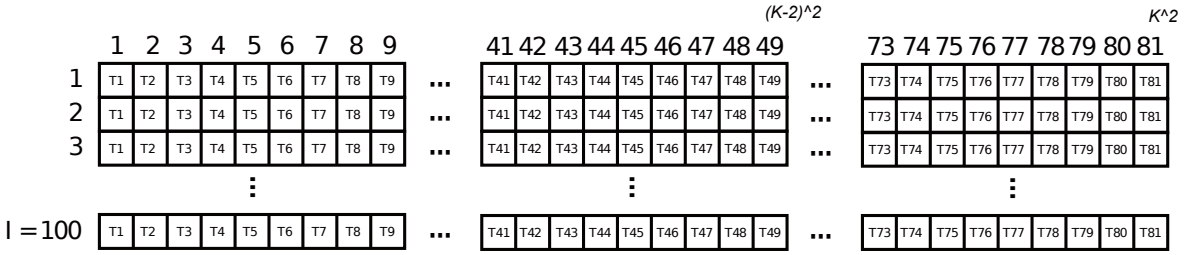


Figura 3.8: Modelado global $\mathbf{X}$, conformado por $I = 100$ observaciones de $J = 81$ variables cada una. K se corresponde con el número de sensores situados en cada lado de la red. T_j, con $j = 1, 2, \cdots, J$, denota cada una de las mediciones del sensor j.

modelos locales, como alternativa mejorada para la recuperación de datos faltantes, en donde se justificará el uso del modelado adecuado dependiendo del objetivo de este.

Concretamente, en esta sección se describirá y hará uso del modelado global de los datos. Así mismo, se evaluará su aplicación para la detección de anomalías y recuperación de datos faltantes mediante la consecución de una serie de experimentos considerando los esquemas de encaminamiento estático que se introdujeron en la Sección 3.5.1.

3.6.1. Modelo global

Cada cierto tiempo, cada uno de los sensores recoge una determinada información de algún proceso o fenómeno físico que se envía a la CU. Esa información se estructura de tal manera que cada columna representa al sensor o variables j del modelo y cada fila se corresponde con una observación i recogida por la CU en un determinado instante de tiempo. A esta forma de organizar los datos se le denomina *modelo o modelado global*. Por consiguiente, y en notación PCA, este modelo se representa como una matriz $\mathbf{X}$ de J columnas (variables o sensores en este caso) e I filas (total de observaciones del conjunto de datos).

La Figura 3.8 muestra gráficamente la estructura de este modelo que, en este caso y a modo de ejemplo, se corresponde con una red compuesta por 81 sensores en total (distribuidos de forma regular en un área cuadrada y con $K = 9$ sensores en cada lado) y 100 observaciones para cada uno de ellos. Claramente, el modelo global $\mathbf{X}$ será de dimensión 100×81.

Como aclaración, hemos de decir que la posición actual de un sensor dentro del modelo no influye en lo que se refiere a los procesos de calibración, monitorización o recuperación.

3.6.2. Escenario de uso I: monitorización y detección de anomalías

Una vez generados los conjuntos de datos sobre los que trabajar, en esta sección veremos cómo se emplean las técnicas de análisis multivariante en el contexto de la monitorización y detección de anomalías. Para ello, el presente estudio se apoya en el uso de la herramienta PLS-toolbox para Matlab [126]. En primer lugar, se ha de elegir el número de PC con el que se generará el modelo PCA. Para ello analizamos la varianza capturada por el modelo cuando se incrementa el número de PC, tal y como se describió en la Sección 3.2.6. Se concluye que 3 componentes principales son suficientes para capturar prácticamente toda la variabilidad existente en el conjunto de datos original (ver Figura 3.2). Además, este número es coherente con el número de focos de temperatura existentes. Comparando el modelo PCA obtenido del conjunto CAL, junto con las nuevas observaciones a monitorizar (aquellos datos que no se tuvieron el cuenta en la calibración, FIR y variantes de ATT), seremos capaces de detectar anomalías en el entorno. Esta detección es realizada a través de gráficos de monitorización como los representados en la Figura 3.9. En la figura, el eje de las abscisas se corresponde con el estadístico T^2 (3.10); representándose Q (3.11) en el eje de ordenadas. Cada instante de muestreo se recoge una nueva observación de la WSN, computándose los *scores* (ver (3.2)) y los estadísticos T^2 y Q. Cada uno de los puntos representados en el gráfico se corresponde con una observación. A través de la Figura 3.9 se aprecia la existencia de observaciones que se escapan de los límites de control establecidos (triángulos rojos invertidos), hecho indicativo del efecto de la evolución del fuego en el conjunto de datos ATT. Todas estas observaciones se consideran anómalas, ya que difieren de aquellas obtenidas para mediciones de temperatura en condiciones normales (círculos oscuros) del conjunto de datos CAL. Cuando se detecta una anomalía en el sistema de monitorización, automáticamente se lanza una alarma. Esto ocurre cuando se rebasan los límites de control durante tres observaciones consecutivas[2]. En el sistema propuesto, los límites de control se eligen de tal manera que el 95 % de las muestras recogidas durante el proceso de calibración queda dentro de dichos límites. Esto significa que, en teoría, la probabilidad de falsos positivos de este simple sistema de detección sería de $0,05^3 = 0,000125$. De nuevo observando la Figura 3.9, el sistema es capaz de detectar una situación de fuego prácticamente desde el inicio de este.

Es también importante remarcar en este punto que el sistema de monitorización no es capaz de distinguir automáticamente entre tipos de anomalías. Es decir, cuál de ellas se corresponde con el efecto del verdadero fuego o cuál de ellas es debida a una actuación maliciosa o un mal funcionamiento de un sensor. Es común en este tipo de sistemas de monitorización con modelos PCA o PLS, la existencia de un supervisor humano que distinga entre anomalías reales o no (ver Sección 3.4). Un ejemplo de aplicación de estos sistemas es la monitorización de procesos industriales [117]. Con

[2]Este número de observaciones anómalas consecutivas es típico en sistemas MSPC.

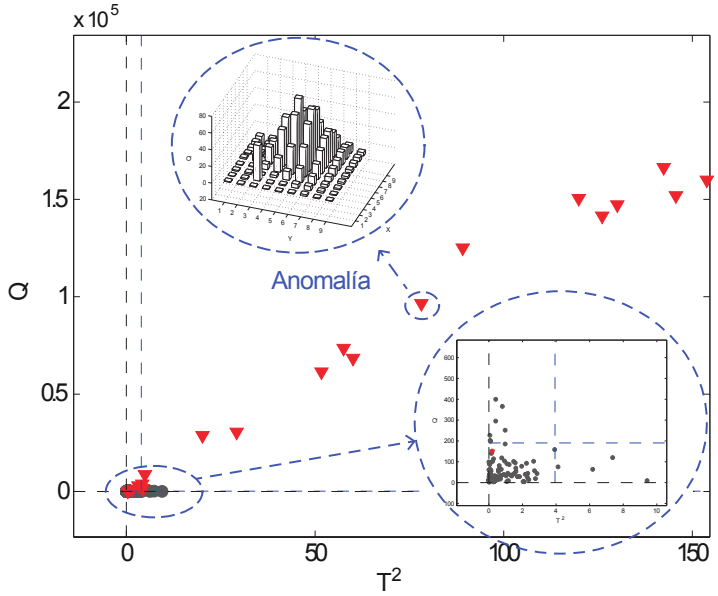


Figura 3.9: Gráfico de monitorización en donde se aprecian los datos iniciales de calibración (círculos oscuros) así como los límites de control establecidos (líneas azules discontinuas) para la detección de anomalías (triángulos rojos invertidos). Arriba a la izquierda se detalla el gráfico de contribución de Q para una observación específica.

el fin de ayudar a dicho supervisor en el proceso de decisión, se hace uso de los gráficos de contribución una vez consolidada la alarma. En la parte superior de la Figura 3.9 se expone un ejemplo de gráfico de contribución para el estadístico Q asociado a una determinada observación anómala.

En la Figura 3.10 se muestra el perfil típico del fuego teniendo en cuenta la contribución a Q de cada uno de los sensores. A su vez, las Figuras 3.11(a), 3.11(b) y 3.11(c) ilustran los perfiles correspondientes a las diferentes situaciones de ataque (ADR, AMR y ALR, respectivamente) para el mismo escenario de fuego. Sobre las figuras anteriores, se observa cómo los ataques producen claros y pronunciados artefactos cuya forma depende del algoritmo de *routing* seleccionado. Resulta visible que estos gráficos de contribución distan bastante de parecerse a la forma suavizada del perfil típico del fuego.

Aunque el hecho de que una persona intervenga en el sistema de monitorización y detección podría ser considerado como una limitación, el caso práctico que nos ocupa, la extinción de incendios, es de suficiente relevancia como para que dicha persona exista. No obstante, y de manera tentativa, se podría proponer una solución automática a la detección de ataques en este contexto, empleando técnicas de filtrado de uso típico en el procesado de imágenes [127]. Se podrían utilizar, así, filtros específicos para resaltar y aislar los artefactos que produce un ataque sobre el gráfico

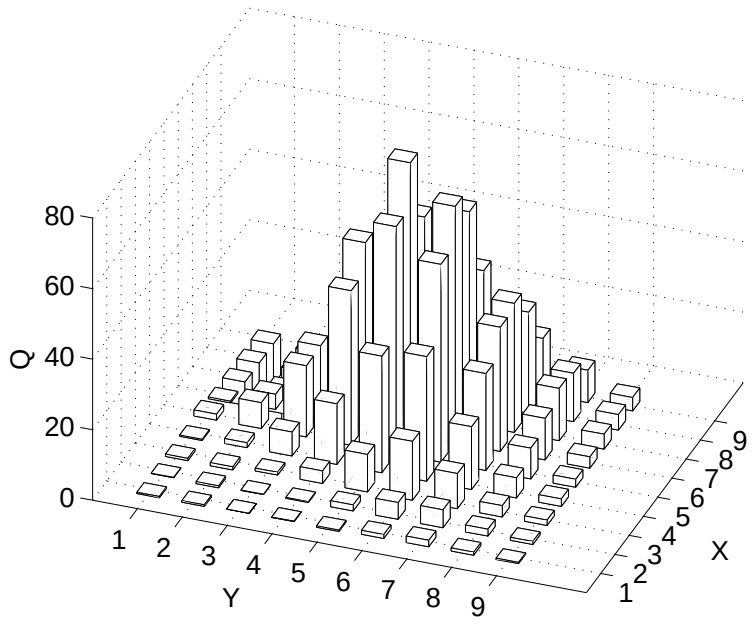


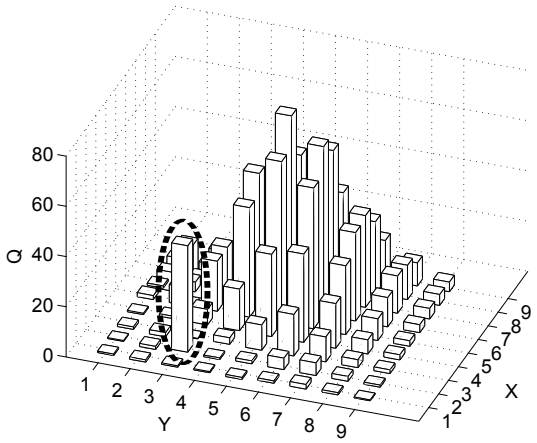
Figura 3.10: Gráfico de contribución Q de una observación determinada bajo la influencia del fuego obtenida a partir del conjunto de datos FIR.

de contribución Q. Como se ha mencionado anteriormente, el perfil Q de cada situación de ataque está estrechamente ligado al algoritmo de *routing* subyacente; por lo tanto, el filtro ideado debería adaptarse en consecuencia. Tomando como ejemplo la situación AMR, bastaría con aplicar una ventana de filtrado que detectase líneas, tal y como se muestra en la Figura 3.12. Una vez aplicado el filtrado, se observa cómo se acentúa considerablemente la línea afectada por el ataque. Después de este filtrado, se establecerían umbrales que determinen, ahora sí, y de manera automática, si la anomalía proviene del propio fuego o es generada por una actuación maliciosa.

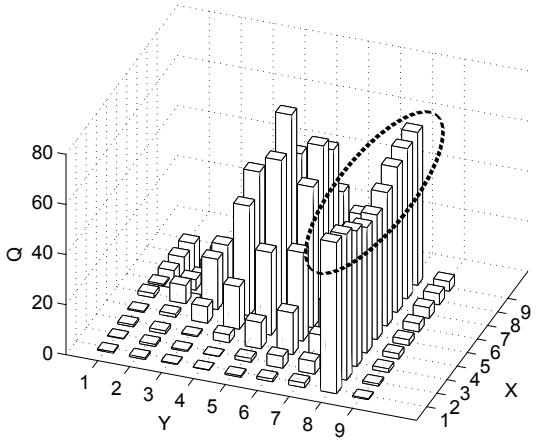
Cualquiera que sea el método de detección empleado, manual o automático, una vez detectado el ataque se ejecutará el procedimiento de recuperación de datos para mitigar los efectos adversos que pudiera tener la pérdida o alteración de datos, es decir, para recalcular los valores que sustituyan a los artefactos encontrados. A continuación se procede a evaluar la eficiencia del procedimiento de recuperación de datos faltantes para los tres escenarios de ataque considerados.

3.6.3. Escenario de uso II: recuperación de datos faltantes

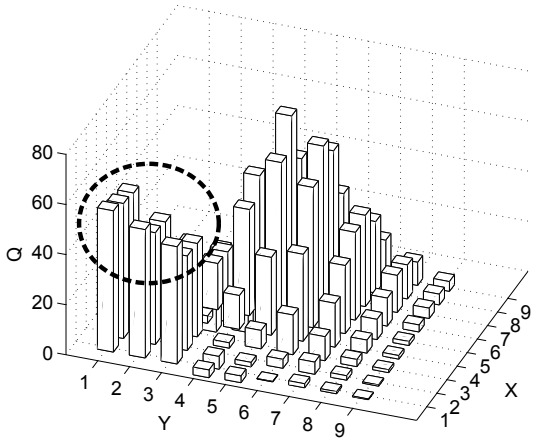
Como parte del sistema conjunto de seguridad, una vez detectado el ataque se trata de mitigar sus efectos empleando metodologías de recuperación de datos faltantes a través del uso de técnicas de análisis multivariante.



(a)



(b)



(c)

Figura 3.11: Perfiles Q correspondientes a un determinada observación para los escenarios de ataque (a) ADR, (b) AMR y (c) ALR y bajo la influencia del fuego.

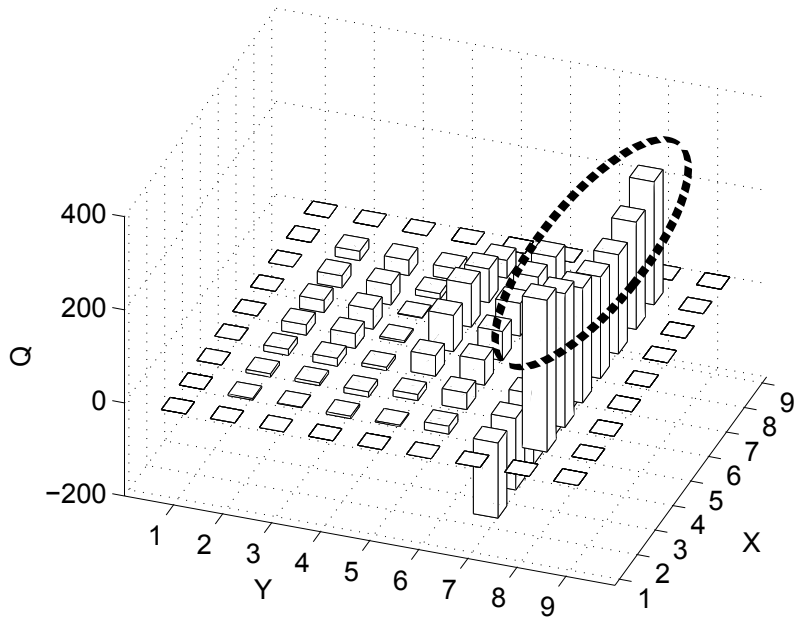


Figura 3.12: Gráfico de contribución Q después de la aplicación del filtrado específico para el ataque AMR mostrado en la Figura 3.11(b). El filtro hace que se acentúe la contribución Q de los sensores afectados, para luego poder aplicar mecanismos de detección basados en la utilización de umbrales. Rodeada con línea discontinua se resalta la línea de sensores afectados por el ataque.

Para que el método de imputación trabaje de manera óptima es necesario seleccionar el número de PC adecuado. Con ese fin se utiliza el método de validación cruzada descrito en la Sección 3.2.6 para el subsecuente análisis de la curva PRESS. En la Figura 3.13 se observa cómo con 3 PC se obtiene el mínimo valor de PRESS, siendo este el número adecuado de PC para ser tenidas en cuenta.

Una vez obtenido el número óptimo de PC, procedemos a evaluar el método de imputación de datos TSR-PCA para las situaciones de ataque antes mencionadas: ADR, AMR y ALR. Las Figuras 3.14(b), 3.14(d) y 3.14(f) ilustran los resultados obtenidos. Los perfiles Q conseguidos suavizan su forma después del proceso de recuperación, con respecto a los casos originales mostrados en las Figuras 3.14(a), 3.14(c) y 3.14(e). Sin embargo, los resultados que produce la recuperación distan mucho de ser óptimos. La contribución de Q ofrecida por sensores alejados de la acción del fuego es menor que la de aquellos más cercanos. Este comportamiento está motivado porque el modelo PCA se estimó en condiciones normales, es decir, sin la presencia de fuego, mientras que en los experimentos simulados para las situaciones de ataque, el fuego es un actor principal. De esta manera, y como se aprecia en las Figuras 3.14(d) y 3.14(f), la recuperación ante los ataques mencionados es más efectiva en aquellas zonas en las que el fuego no está presente o no interviene demasiado, esto es, zonas alejadas del centro del área. Se observa cómo los valores

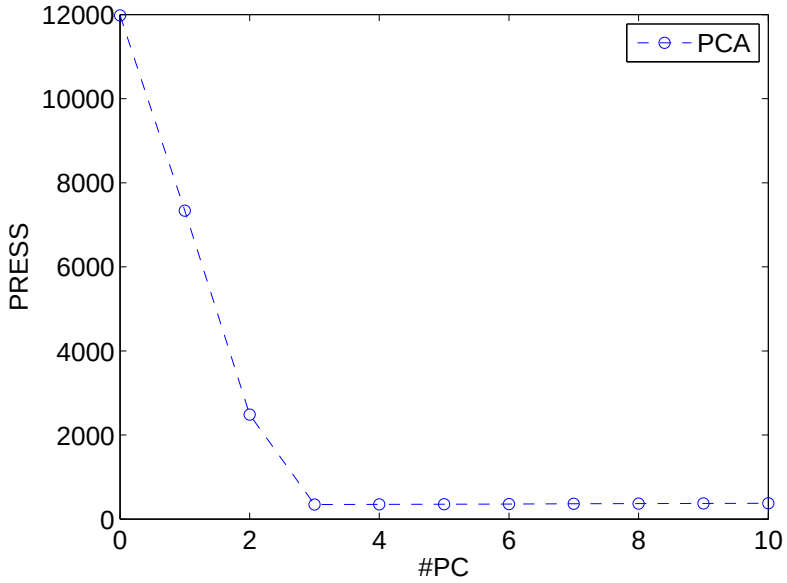


Figura 3.13: Curva PRESS considerando el conjunto de datos CAL del modelo global PCA.

recuperados en zonas cercanas al fuego no se imputan correctamente, obteniéndose valores de contribución Q incluso mayores que los originales.

Más allá de los resultados gráficos obtenidos, la Tabla 3.1 muestra el MSE (*Mean Squared Error*) calculado a partir de la comparación de los datos recuperados y originales (antes del ataque) para las diferentes situaciones de ataque. Con el fin de eliminar la influencia constatada de la localización de los sensores sobre los resultados obtenidos, se computa la media del MSE para todos los sensores de la red (9×9), es decir, cada sensor de la red es atacado y se computa el error recuperación para luego calcular su valor medio total.

Atendiendo a los resultados obtenidos en la Tabla 3.1, ADR obtiene menor MSE ya que el método de recuperación tiene disponible más información válida para imputar. Al contrario, el peor resultado se corresponde con la situación ALR ya que, y siguiendo el razonamiento anterior, ahora TSR-PCA dispone de un menor número de valores válidos para la imputación del sensor afectado; los sensores vecinos del alterado son también modificados, no pudiendo ser utilizados en la recuperación. Por último, en el caso AMR, para cada uno de los sensores influidos por el ataque se dispone de al menos dos, uno arriba y otro abajo, valores válidos para la imputación, permitiendo así una mejor recuperación que en el caso ALR. En resumen, un aspecto clave a considerar en este problema es el algoritmo de *routing* subyacente. Aunque un algoritmo de agregación sea una buena opción desde el punto de vista energético, no lo es tanto desde la perspectiva de la seguridad si se emplean métodos de respuesta basados en imputación de datos.

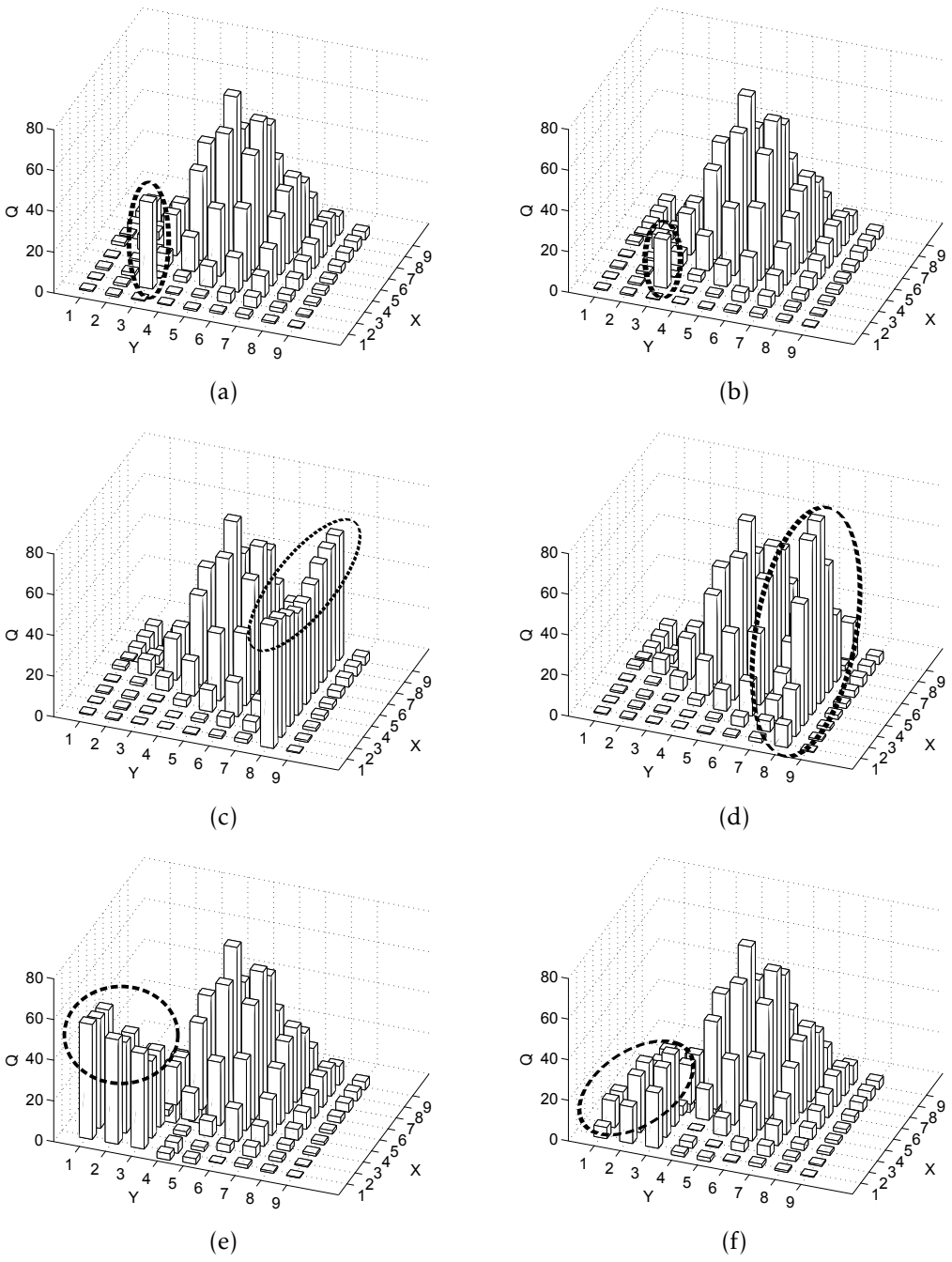


Figura 3.14: Escenarios de ataque *data tampering simulados*. A la izquierda de la figura, y de arriba hacia abajo, se ilustran los perfiles Q correspondientes a una determinada observación para los escenarios ADR, AMR y ALR y bajo la influencia del fuego. A la derecha se expone el resultado de la recuperación para los mismos ataques. Aquellos sensores afectados por cada ataque se remarcan con línea discontinua.

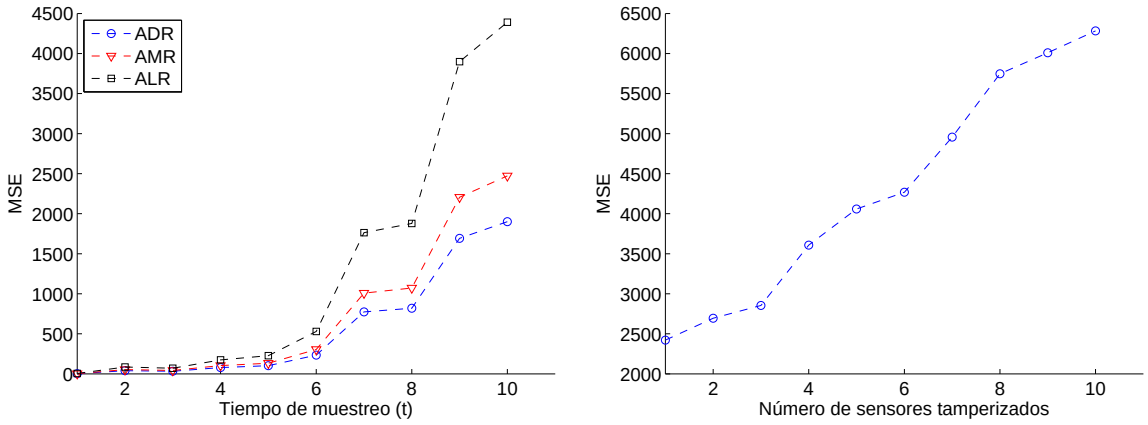
Escenario de ataque	**MSE (TSR-PCA)**
ADR	1900,8
AMR	2472
ALR	4391,6

Tabla 3.1: Comparativa del MSE cometido en la imputación de datos bajo las diferentes situaciones de ataque de *data tampering* usando el método de recuperación TSR-PCA.

Para completar los resultados anteriores, se estudia y evalúa cómo progresa el parámetro MSE a medida que evoluciona el fuego y en función del número de sensores atacados. Estos resultados se presentan en las Figuras 3.15(a) y 3.15(b), respectivamente. En el primer caso se escogen los diez primeros instantes de muestreo desde el origen del fuego. Se observa claramente cómo se produce un incremento en el MSE a medida que evoluciona este. En concordancia con los resultados de la Tabla 3.1, ADR presenta el menor incremento MSE con la evolución del fuego, mientras que ALR incluso llega a doblar este valor en instantes de muestreo en los que el fuego es más intenso. En la Figura 3.15(b) se muestra también el progreso del MSE, pero en este caso en función del número de sensores afectados considerando solamente el caso ADR. Hemos de recordar que en esta situación el número de sensores atacados y afectados es el mismo. Los resultados que se muestran en la figura se obtienen atacando aleatoriamente desde 1 hasta 10 sensores, mostrándose que cuanto mayor es el número de sensores comprometidos, mayor es el error aparecido. Este comportamiento es debido, en parte, a que es posible seleccionar aleatoriamente sensores adyacentes al atacado, de manera que el método de imputación tiene menos valores válidos sobre los que apoyarse para predecir. Como ya se introdujo en secciones anteriores, TSR ofrece mejor rendimiento cuando se trabaja con información correlada. Aun así, se sigue una tendencia lineal con baja constante multiplicativa, tal que, si se observa la figura, el hecho de atacar siete sensores solo dobla el error cometido al comprometer exclusivamente uno.

3.7. Aplicación de modelos globales y encaminamiento dinámicos para la mejora de la recuperación de datos faltantes

Atendiendo a la sección anterior, es apreciable el hecho de que los modelos globales no consiguen ser eficientes en la recuperación de datos alterados o faltantes. Es más, es posible que la imputación del valor de un sensor sea incluso peor que la alteración producida de manera malintencionada. Para mejorar el rendimiento del



(a) (b)

Figura 3.15: Evolución del MSE en presencia de fuego empleando modelos globales para: (a) las 10 primeras observaciones considerando los tres escenarios de ataque ADR, AMR y ALR, y (b) a medida que el número de sensores atacados se incrementa para el caso ADR.

método de recuperación utilizado, se hará uso conjunto de modelos y encaminamiento dinámicos.

3.7.1. Modelo global dinámico

Descrito en la Sección 3.2.5, TSR persigue restaurar valores de sensores afectados cualquiera que sea el origen de la alteración o pérdida, bien por actuación maliciosa (ataque) o por mal funcionamiento del propio sensor. Para ello se apoya en los valores disponibles que no han sido alterados y en la correlación existente en la información. De esta manera, su eficiencia de recuperación aumentará con la existencia de un mayor número de valores válidos, así como con el grado de correlación existente entre ellos. Motivado por la forma de operar de TSR, se propone aquí introducir la relación temporal existente entre las variables o sensores de la WSN, con el objetivo de mejorar el rendimiento de la imputación. Hemos de notar que el modelado global expuesto en la anterior sección no contempla relación temporal alguna entre las variables existentes, de tal manera que solo se aprovecha de la correlación espacial, hecho que puede suponer un hándicap dependiendo de la aplicación final del modelo.

Para incorporar la naturaleza dinámica, por otro lado inherente, de las WSN dentro del modelo, se utilizará DPCA (ver Sección 3.2.2). A modo de recordatorio, decir que DPCA extiende la matriz de datos original PCA ($\mathbf{X}$) añadiendo observaciones de dichas variables en tiempos anteriores o *lags*. En la Figura 3.16 se observa la estructuración final del, en adelante, *modelo global dinámico*. La nueva matriz $\mathbf{X_d}$ tendrá un tamaño que variará según el número de *lags* utilizado. De manera general, la

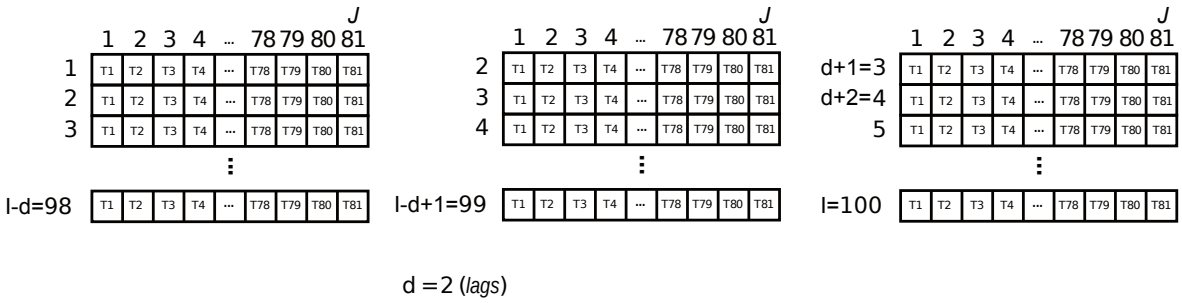


Figura 3.16: Modelado global dinámico $\mathbf{X_d}$, conformado por $J\times(d+1)$ variables e $I-d$ observaciones. J representa el número de variables originales del modelado global PCA, siendo I el número original de observaciones. d se corresponde con el número de *lags* temporales utilizados. Como ya se representó en la Figura 3.8, T_j con $j = 1, 2, \cdots, J$, denota cada una de las mediciones del sensor j.

dimensión de $\mathbf{X_d}$ será $(I-d)\times J\cdot(d+1)$, siendo I el número de observaciones originales, J el número de variables o sensores y d el número de *lags* temporales utilizado para la creación del modelo.

3.7.2. Estrategias de encaminamiento dinámicas

Con el objetivo de introducir el dinamismo de los datos, se propone utilizar el modelado global dinámico DPCA previamente presentado. Con esta reorganización se añade dentro del modelo la información de autocorrelación y correlación cruzada entre variables mediante el empleo de decalajes temporales. Desgraciadamente, tal modelado no serviría de nada si se utilizan las propuestas de encaminamiento estáticas empleadas hasta ahora. Esto es así ya que la información encaminada a través de un sensor intervenido, seguirá estando comprometida si dicha ruta no varía a lo largo del tiempo. Por este motivo, es necesario el diseño y planteamiento de estrategias de *routing* que encaminen los datos alternativamente siguiendo diferentes rutas para diferentes instantes de tiempo. De esta forma, la información comprometida por el sensor atacante variará en cada instante de tiempo de manera que se recoge mayor información temporal útil dentro del modelo DPCA para su posterior utilización en el proceso de imputación de datos.

Los esquemas propuestos pueden ser vistos como modificaciones a la estrategia seguida en MCFA, cuyo fin es evaluar el efecto producido por la adición de variabilidad temporal en las rutas. Todos ellos difieren en cómo se decide cuál será el próximo salto o nodo en la ruta hacia la CU de entre n posibilidades. A continuación se listan las tres filosofías de encaminamiento propuestas:

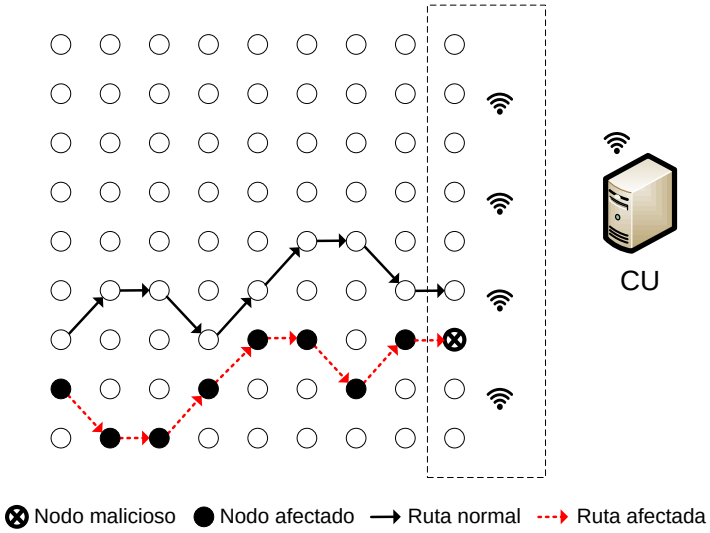


Figura 3.17: Esquema de encaminamiento variable aleatorio (RR, por ejemplo) en el que la información recogida por los demás sensores en la ruta está comprometida por el nodo malicioso, ya que este también es el encargado de reenviar la información hacia la CU.

- RR (*Random Routing*). La selección del siguiente nodo en una ruta *multi-hop* se lleva a cabo de manera aleatoria, tal que cada uno de los n nodos disponibles tendrán la misma probabilidad, $\frac{1}{n}$.
- DRR (*Differential Random Routing*). La selección del siguiente nodo se lleva a cabo aleatoriamente pero solo entre los $n-1$ nodos que no fueron seleccionados justo en el anterior instante de tiempo.
- SR (*Switching-based Routing*). La selección del siguiente nodo se lleva a cabo siguiendo un patrón determinista con el fin de variar las rutas lo máximo posible.

El encaminamiento y su efecto ante ataques de *data tampering*

Basados en las estrategias dinámicas previamente introducidas, se estudian en esta subsección escenarios adicionales de ataque con el fin de evaluar su impacto sobre el sistema ideado. Para la evaluación de dichos escenarios, se considerará que el número de nodos disponibles a elegir como siguiente salto será $n = 3$, que se corresponden con los tres nodos a la derecha de uno dado, considerando la situación de la CU. La Figura 3.17 ilustra un ejemplo ilustrativo de *routing* de este tipo de escenario. Aquí, las rutas ofrecidas por el algoritmo varían (por ejemplo, mediante el uso del esquema RR) a lo largo del tiempo; en concreto, para cada instante de muestro.

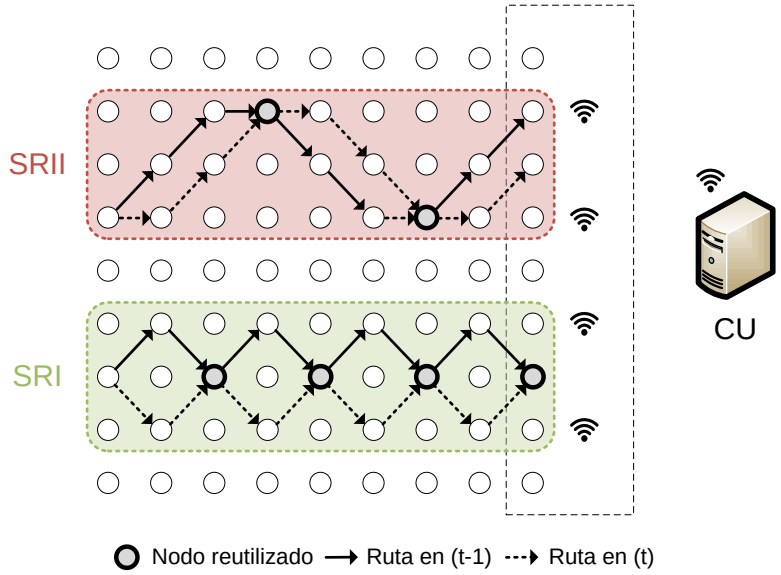


Figura 3.18: Alternativas de encaminamiento SR. En la parte inferior se observa la variante SRI, y en la parte superior se ilustra la variante SRII. Tomando dos instantes de muestreo consecutivos $t-1$ y t, SRII reutiliza solamente dos nodos mientras que SRI usa cuatro cuando la información se reenvía de izquierda a derecha para llegar a la CU.

Además de las estrategias RR y DRR, se proponen aquí tres diferentes variantes del esquema SR. Estas son, SRI, SRII y SRIII. SRI define un patrón de conmutación entre filas adyacentes a la del nodo que selecciona el siguiente salto en el encaminamiento. Dicho nodo podrá elegir entre dos posibles alternativas: el sensor inmediato a la derecha correspondiente de la fila superior o el mismo sensor pero ubicado en la fila inferior. Como se observa en la parte inferior de la Figura 3.18, se trata de un patrón relativamente sencillo y, como tal, no consigue obtener un alto grado de variabilidad en las rutas que propone, ya que la mitad de los nodos se vuelven a usar después de la conmutación. En el ejemplo ilustrativo de la Figura 3.18 comprobamos que, independientemente del tiempo de muestreo considerado, son siempre los mismos cuatro sensores los reutilizados. En el caso de SRII se define un patrón más complejo, implicando a cada conjunto de tres filas. Este esquema reduce considerablemente el número de nodos coincidentes o reutilizados entre conmutaciones. A modo de ejemplo ilustrativo, en la parte superior de la Figura 3.18 se observa este comportamiento. El esquema SR en su variante SRIII introduce una selección aleatoria pero solo para el primer instante de muestreo. A continuación, se retoma la filosofía SRII. Tal y como se hizo para el conjunto de escenarios de ataque estáticos, de aquí en adelante se contemplarán los siguientes escenarios de ataque dinámicos en función del algoritmo de encaminamiento utilizado. Estos son: ARR (*Attack on RR*), ADRR (*Attack on DRR*) y ASR (*Atacck on SR*), este último en sus diferentes versiones.

3.7.3. Evaluación de las mejoras introducidas

Siguiendo el esquema completo de detección y respuesta, una vez que se genera una alarma debida a la detección de la acción maliciosa sobre alguno de los sensores, se lleva a cabo el proceso de respuesta que tratará de mitigar en la mayor medida posible los efectos perjudiciales del ataque en curso.

Para evaluar la eficacia de este nuevo enfoque, se computará el MSE durante 10 observaciones consecutivas (en concreto desde la quinta hasta la decimocuarta) utilizando el conjunto de datos FIR donde se incluye la acción y evolución del fuego. Se considerarán atacados aquellos sensores sobre los cuales el efecto de dicho ataque será el más perjudicial posible: los sensores más a la derecha, dentro de la topología de red regular que se sigue. Los resultados obtenidos para cada uno de los escenarios de ataque considerados junto con el empleo de diferentes *lags* temporales, desde $d = 0$ hasta $d = 4$, se muestran en la Figura 3.19. Con $d = 0$ se utilizará el modelado global expuesto de la Sección 3.6.1, que no incluye información alguna sobre la relación temporal entre variables. Con fines comparativos se incluye también en los resultados la evolución del MSE para el escenario AMR expuesto en la Figura 3.7.

La principal conclusión que se obtiene a partir de la evolución del MSE mostrada en la Figura 3.19 es que en todos los casos el rendimiento de la imputación se incrementa con el número de *lags* considerados, ya que con dicho aumento el modelo global dinámico es capaz de recoger mayor información de correlación entre variables. Es más, con el empleo de un único *lag* la propuesta dinámica mejora claramente a la solución estática. Se corrobora así la eficiencia de la combinación de estrategias de encaminamiento variables con el uso de modelos globales dinámicos en la imputación de datos. Es notable la similitud en los resultados para aquellos escenarios de ataques que emplean estrategias de *routing* con cierto grado de aleatoriedad (ARR, ADRR y ASRIII), así como para las que, por otro lado, están basadas en pura conmutación determinista. A partir de la Figura 3.19 también se observa cómo las estrategias deterministas mejoran las probabilísticas cuando el número de *lags* es bajo. Dentro de la clara mejora introducida sobre la solución estática, se puede concluir que los mejores resultados obtenidos son aquellos que provienen del empleo de estrategias de *routing* con cierto grado de aleatoriedad y para un número mayor de *lags*. Este hecho se observa en detalle en la Figura 3.20, en donde se compara la evolución del MSE para cada estrategia y el máximo número de *lags* considerado, $d = 4$. Dicho comportamiento tiene su motivación en el hecho de que los métodos probabilísticos proporcionan una mejor distribución de rutas. Como consecuencia directa, es mayor la información válida disponible y utilizada por el método de imputación a la hora de recuperar los datos afectados. Dentro de los escenarios de ataque que utilizan algoritmos de *routing* considerados como puramente deterministas, ASRII prevalece sobre su homólogo ASRI, ya que el primero reutiliza un número menor de sensores en la ruta para instantes de muestreo consecutivos. En la Tabla 3.2 se muestran los

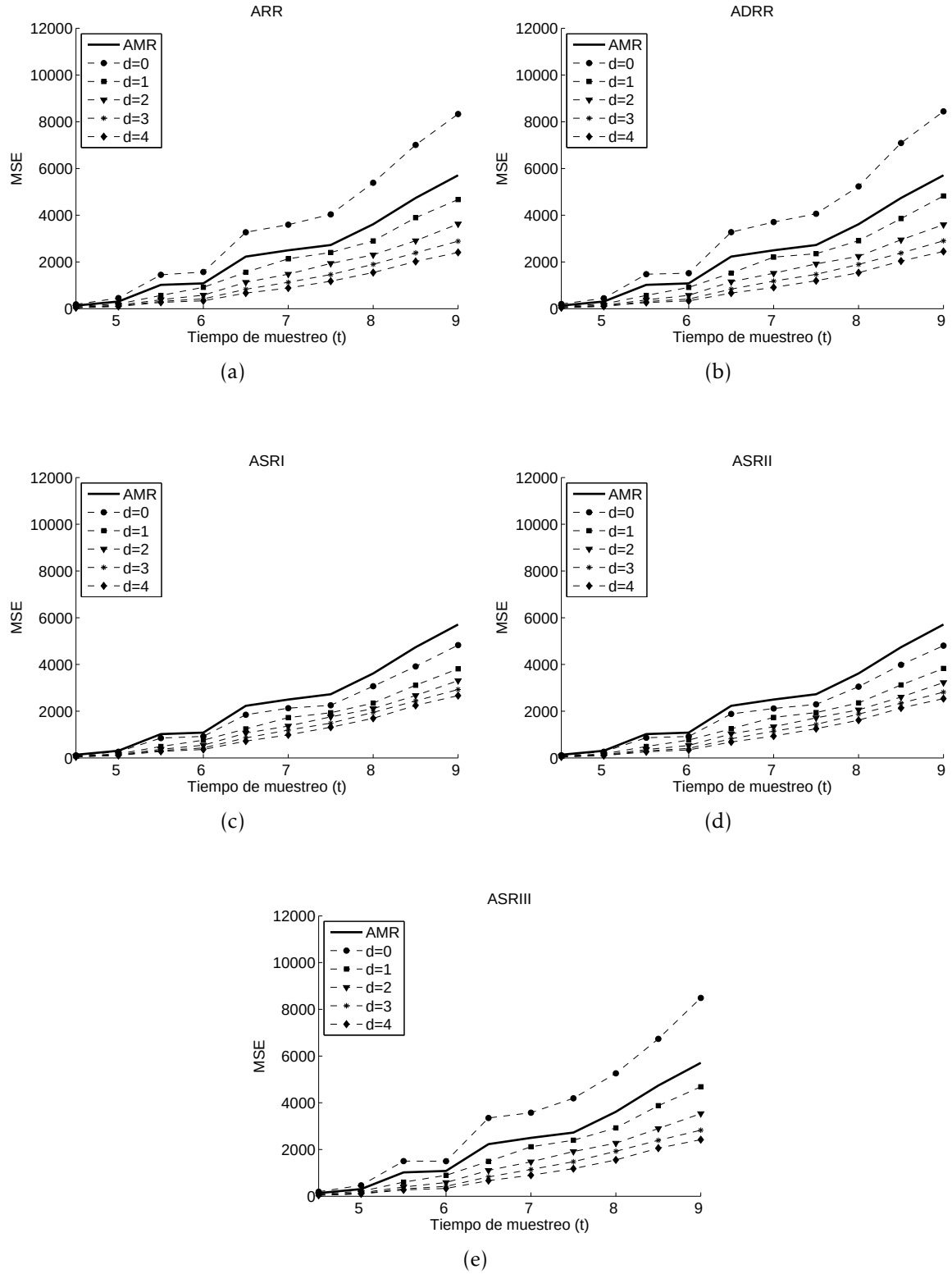


Figura 3.19: Evolución del MSE teniendo en cuenta cada uno de los escenarios de ataque considerados y variando el número de *lags* temporales desde $d = 0$ hasta $d = 4$: (a) ARR, (b) ADRR, (c) ASRI, (d) ASRII y (e) ASRIII.

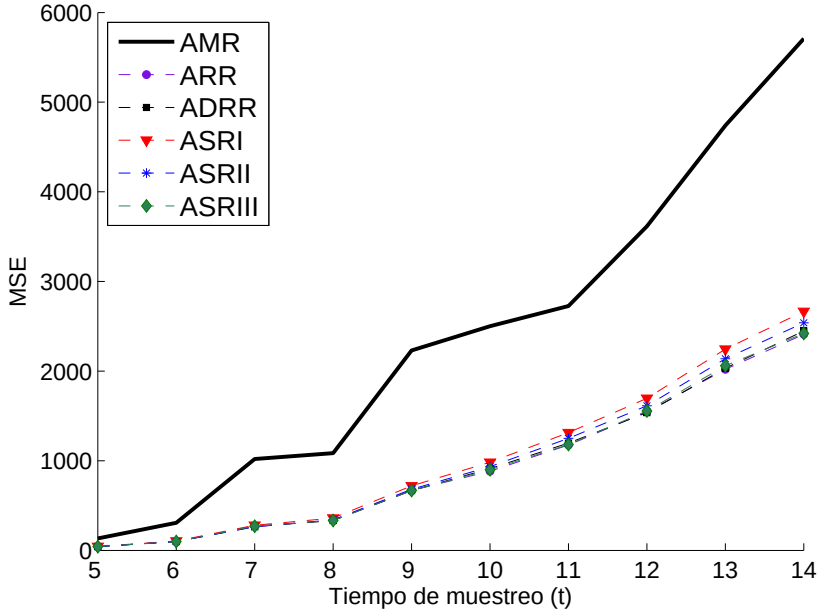


Figura 3.20: Evolución del MSE para cada escenario de ataque considerado empleando $d = 4$ *lags* temporales.

valores de error MSE para el instante de muestreo $t = 10$. Tras el análisis de estos resultados, el método de recuperación propuesto es capaz de mejorar su eficiencia hasta en un 60 % en comparación con la solución estática. Como conclusión final, y a la vista de los resultados, se puede decir que cuanto mayor sea el grado de variabilidad contemplada en las rutas elegidas mayor será la eficiencia del método de recuperación.

	MSE				
Escenarios de ataque	*d=0*	*d=1*	*d=2*	*d=3*	*d=4*
AMR	2500	–	–	–	–
ARR	3594,9	2135,7	1476,2	1127,9	883,2
ADRR	3708,2	2207,7	1514,3	1180,4	912,1
ASRI	2128,7	1727	1364,8	1191,7	983,2
ASRII	2116,3	1735	1330,8	1143,7	933,2
ASRIII	3576,8	2118	1472	1145,3	899,3

Tabla 3.2: Resultados numéricos de MSE para cada escenario de ataque, obtenidos en el instante de muestreo $t = 10$.

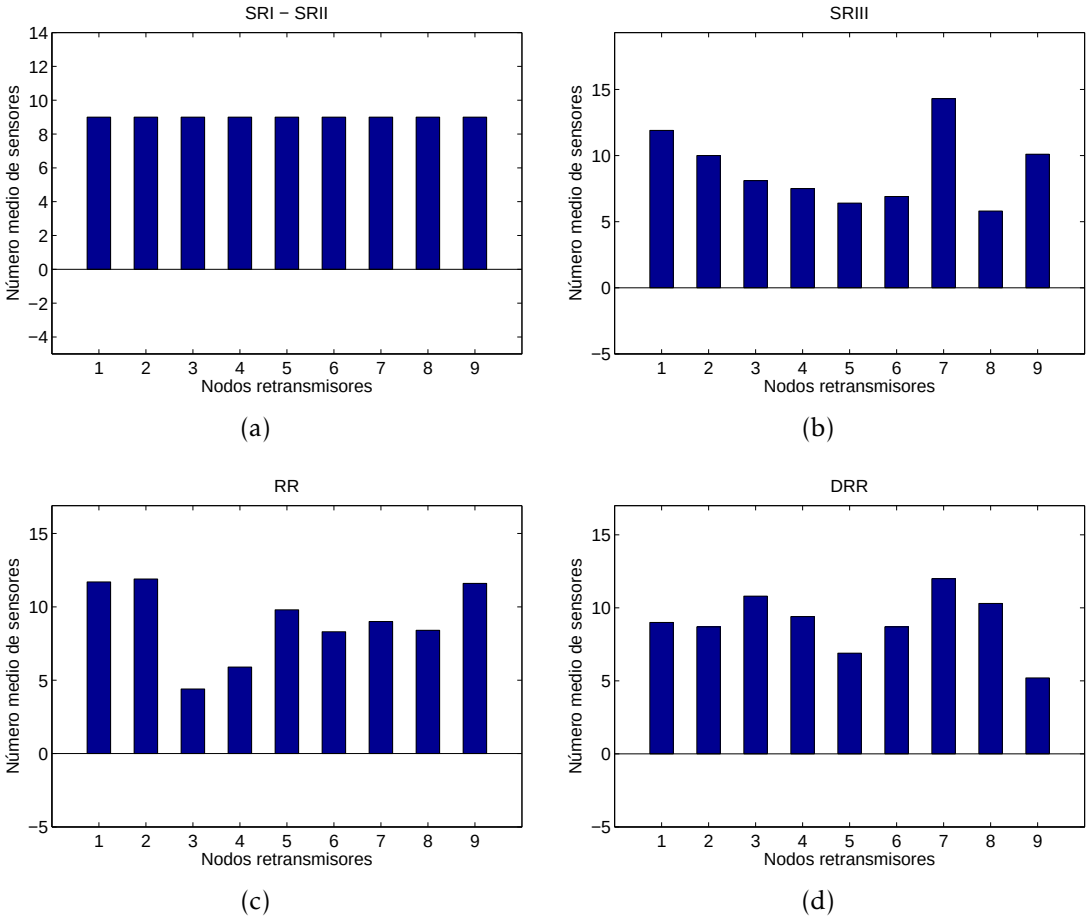


Figura 3.21: Número medio de sensores cuya información se reenvía a través de cada uno de los nodos retransmisores. Se tienen en cuenta los 10 primeros instantes de muestreo, así como cada una de las estrategias de encaminamiento: (a) SRI y SRII, (b) SRIII, (c) RR y (d) DRR.

En contraposición con el rendimiento ofrecido por los algoritmos de encaminamiento probabilísticos en lo que respecta a la imputación de datos, y tras el estudio de la distribución de carga que cada uno de los sensores retransmisores soporta según la estrategia empleada, se puede decir que estos no distribuyen el tráfico de manera equitativa. En la Figura 3.21 se observa este comportamiento, mostrándose el número medio de sensores cuya información se reenvía por cada uno de los nodos retransmisores hasta el instante de tiempo $t = 10$. Si bien este hecho podría considerarse como un inconveniente de cara a la viabilidad práctica de este tipo de técnicas (más carga de tráfico en un sensor retransmisor supone un consumo energético mayor, limitando su tiempo de vida), hay que tener en cuenta que se ha contemplado un número relativamente bajo de instantes de muestreo, esperándose que el nivel de

carga de tráfico soportado por cada nodo retransmisor se equipare a lo largo del tiempo.

En resumen, y como aspectos relevantes a destacar son, primero, el hecho de la viabilidad teórica de estas técnicas y, segundo, la problemática asociada al diseño óptimo de algoritmos de *routing* proporcionando un adecuado balance entre resistencia y eficiencia energética, ambos de vital relevancia para la continuidad de los servicios ofrecidos por la red y, por supuesto, para su supervivencia.

3.8. Aplicación de modelos locales para la mejora de la recuperación de datos faltantes

La adición de la información temporal dentro del modelo, junto con la necesaria utilización de estrategias de encaminamiento dinámico, mejoran el rendimiento de la imputación de datos. Si bien puede ser considerada como una alternativa viable a los modelos globales por sí mismos, dista mucho todavía de ser la mejor solución. Esto es debido principalmente a que no toda la información que utiliza el método TSR está correlada con el sensor cuyo valor se intenta recuperar. Con objeto de solventar este problema, se idean y construyen los denominados *modelos locales*.

Durante esta sección se evaluará el rendimiento de esta novedosa reestructuración en los datos obtenidos junto con su aplicación con las estrategias de encaminamiento estático previamente descritas en la Sección 3.5.1.

3.8.1. Modelo local

Se ha demostrado que los modelos globales no son óptimos para la recuperación de datos puesto que no toda la información que se utiliza está correlada con los sensores afectados. Este hecho se agrava para redes WSN cuyos sensores se dispersan demasiado dentro del área que se está monitorizando. Por esta razón, es necesario idear una organización de datos diferente que estructure dicha información y garantice su correlación con los sensores afectados. Para ello se utiliza la idea de localidad, coherente con la correlación espacial existente en las WSN. Así, una solución es considerar aquellos sensores más cercanos al implicado. A la estructura resultante es a lo que llamaremos *modelo local* PCA o PLS, según la técnica multivariante utilizada. Durante la presente y a lo largo de la Sección 3.9 se estudiará el comportamiento de estos modelos y cómo su empleo afecta al rendimiento de la recuperación de datos.

Es importante tener en cuenta que la obtención del modelo local no es trivial cuando se manejan datos provenientes de despliegues WSN no regulares. A través

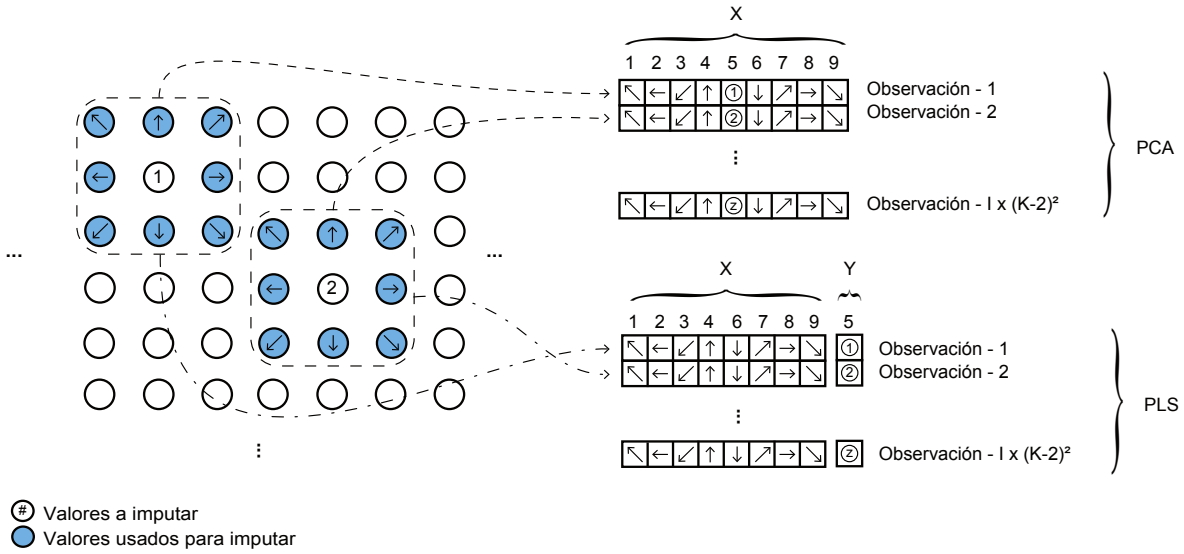


Figura 3.22: Proceso de construcción del modelo local PCA y PLS para una red con topología regular cuadrada. La localidad que se ilustra se establece en 9 sensores, tal que el número total de observaciones es $I \times (K-2)^2$, con I el número de observaciones original y K el número de sensores situado en cada lado de la red. El número de sensores considerados para construir el modelo será $(K-2)^2$, que se corresponde con el número de sensores internos. En la estructura de la figura, z se corresponde con el último valor del sensor de la I-ésima observación original.

de las Figuras 3.22 y 3.23 se observa el proceso de construcción y la estructura final de los modelos locales PCA y PLS para topologías de red regulares e irregulares, respectivamente. En ambos casos se ha definido la vecindad de un determinado sensor como aquellos sensores más cercanos en términos de distancia Euclídea. En concreto, los ocho sensores más cercanos, como se aprecia en las Figuras 3.22 y 3.23. Para el caso de una WSN con topología regular, cada vecino se representa por una flecha indicando su posición relativa al sensor afectado, siendo un identificador numérico el utilizado en el caso de la red no regular. En este último caso los sensores vecinos serán numerados desde el 1 al 8, siendo el 1 el vecino más cercano y el 8 el más alejado. En la Sección 3.9.4 se evaluará el efecto del número de vecinos tenidos en cuenta sobre el rendimiento en la recuperación de valores faltantes en un despliegue no regular real.

Cada tiempo de muestreo se obtiene tanto la medida obtenida por el sensor objetivo como la de sus vecinos, es decir, nueve muestras en total (la localidad es 9-dimensional) que constituyen una observación en este nuevo modelado. Para el caso de la red regular solo se considerarán los $(K-2) \times (K-2)$ sensores más internos de la red, siendo K el número de sensores que se distribuyen en cada arista de la red. Consiguientemente, $\mathbf{X}$ tendrá una dimensión total para este modelo de $I \cdot (K-2)^2 \times 9$. Por otro lado, definir la localidad o vecindad para redes no regulares no es una tarea

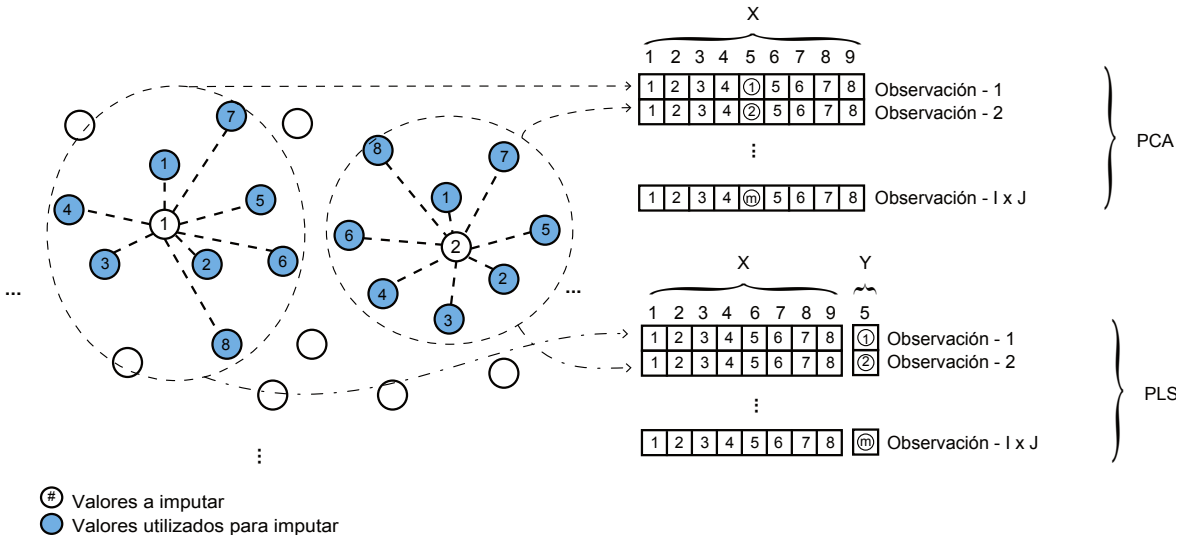


Figura 3.23: Proceso de construcción del modelo local PCA y PLS para una red de sensores con topología no regular. Se establece una localidad de 9 sensores, tal que el número total de observaciones será $I \times J$, donde I es el número de observaciones originales y J es el número de sensores. En esta figura m se corresponde con el valor del último sensor de la I-ésima observación original.

trivial. Por esta razón se define el siguiente procedimiento en donde se considera conocida la posición exacta de cada sensor, o al menos se presupone un cierto grado de exactitud:

- Dado un sensor, se obtienen aquellos 8 sensores más cercanos de acuerdo a la distancia Euclídea que los separa (ver Figura 3.23).
- Para conformar el modelo local, cada observación 9-dimensional se construye combinando el valor del sensor objetivo junto con sus 8 sensores más cercanos previamente localizados. En este punto, es importante destacar que los sensores se organizan en un determinado orden. Por ejemplo, en PCA el valor del sensor objetivo se inserta justo en el medio de sus vecinos, después de los 4 más cercanos y antes de aquellos 4 restantes más alejados, conformando así la matriz $\mathbf{X}$. Aunque el orden no es relevante en sí mismo, sí que lo es el hecho de que todas las observaciones tengan que seguir la misma disposición. Para el caso de PLS existen dos matrices diferenciadas: $\mathbf{X}$ e $\mathbf{Y}$. La primera de ellas contendrá todas las variables (sensores) excepto la objetivo de la recuperación. Esta estará localizada en la matriz $\mathbf{Y}$, estimándose sus valores a partir de $\mathbf{X}$.

Para clarificar el proceso de construcción del modelo local tomemos el siguiente caso concreto a modo de ejemplo. Disponemos de una red WSN regularmente distribuida de $J = 81$ sensores con $I = 100$ observaciones de cada uno de ellos. A su

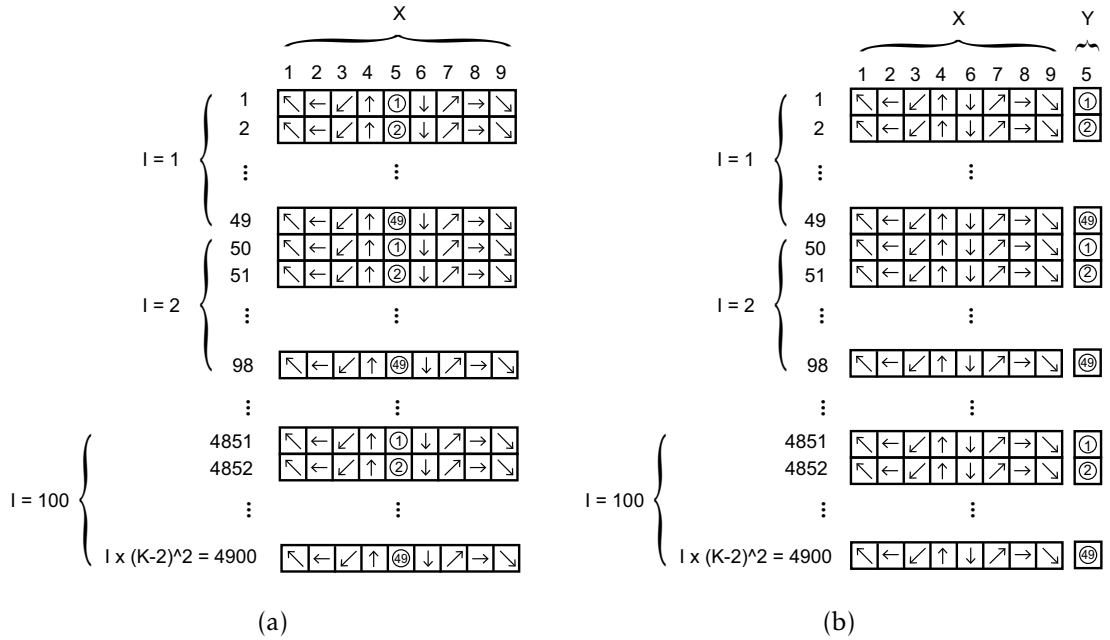


Figura 3.24: Modelos locales PCA (a) y PLS (b) construidos a partir de los datos del modelo global mostrado en la Figura 3.8, compuesto por $I = 100$ observaciones, $J = 81$ sensores y $K = 9$ sensores que componen cada lado de la red.

vez, cada arista de la red regular tendrá $K = 9$ sensores. Las Figuras 3.24(a) y 3.24(b) muestran la nueva organización de los datos en base a la localidad para PCA y PLS, respectivamente. Como ya se mencionó, se hará uso de 9 sensores (el afectado y sus 8 vecinos más cercanos) e $I = 100$ observaciones. Por lo tanto, tendremos $(K-2)^2 = 49$ sensores internos, conformándose un tamaño total de 4900×9 en el caso de PCA y 4900×8 en **X** y 4900×1 en **Y** en el caso de PLS.

3.8.2. Evaluación de las mejoras introducidas

Como ya se hizo en la Sección 3.6.3 con el modelado global, hemos de obtener el número óptimo de PC para el modelo local. De acuerdo a la Figura 3.25, el mínimo valor PRESS se obtiene para 7 PC, tanto para PCA como para PLS.

A modo de comparación entre el uso de modelos locales frente a globales para la recuperación de datos, se calcula el valor del error MSE cometido, así como los gráficos de contribución Q en las situaciones de ataque que se están manejando. La Figura 3.26(b) muestra el perfil Q obtenido después de la imputación de datos para el caso ADR. Si observamos ahora el gráfico, vemos cómo se mejora claramente el resultado de imputación con respecto al obtenido con modelos globales de la Figura 3.14(b). Es más, prácticamente replica el perfil producido por el fuego original con

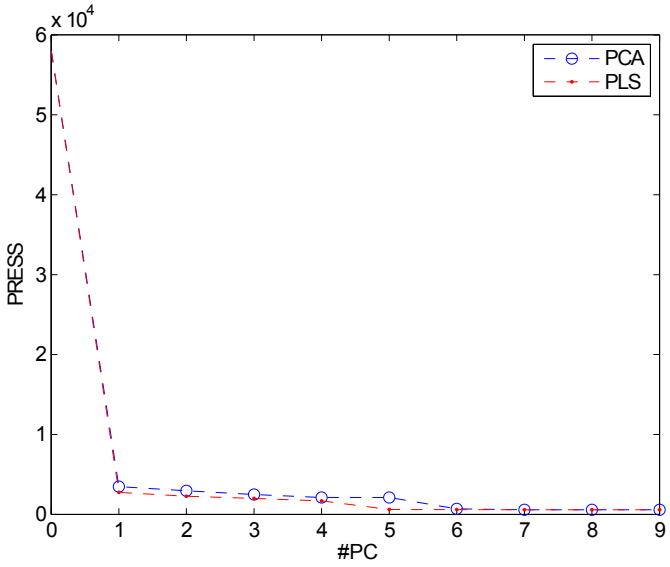


Figura 3.25: Curva PRESS considerando el conjunto de datos CAL del modelo local PCA y PLS. El mínimo valor se obtiene con 7 PC para ambos casos.

la ausencia de ataques (ver Figura 3.26(a)). Numéricamente se puede llegar a la misma conclusión sin más que echar un vistazo a los resultados MSE obtenidos en la Tabla 3.3 que compara estos resultados con los obtenidos con el modelo global. De igual manera, se observa la efectividad y rendimiento de los modelos locales para las situaciones AMR y ALR a través de las Figuras 3.26(c) y 3.26(d), respectivamente. Los correspondientes resultados MSE de los anteriores ataques se pueden también constatar en la Tabla 3.3.

En resumen, los resultados mostrados en la Tabla 3.3 corroboran los beneficios de usar modelos locales frente a globales para la imputación de datos. La nueva estructuración de los datos reduce significativamente el error cometido durante el proceso de recuperación. En el caso de ADR se alcanza una reducción del 99,86 %

	Modelo global	**Modelo local**	
Escenarios de ataque	**MSE (TSR-PCA)**	**MSE (TSR-PCA)**	**MSE (TSR-PLS)**
ADR	1900,8	2,6506	3,4036
AMR	2472	67,7999	71,085
ALR	4391,6	149,5746	149,5746

Tabla 3.3: Comparativa de resultados numéricos de MSE para los escenarios de ataque ADR, AMR y ALR considerando modelos globales y locales.

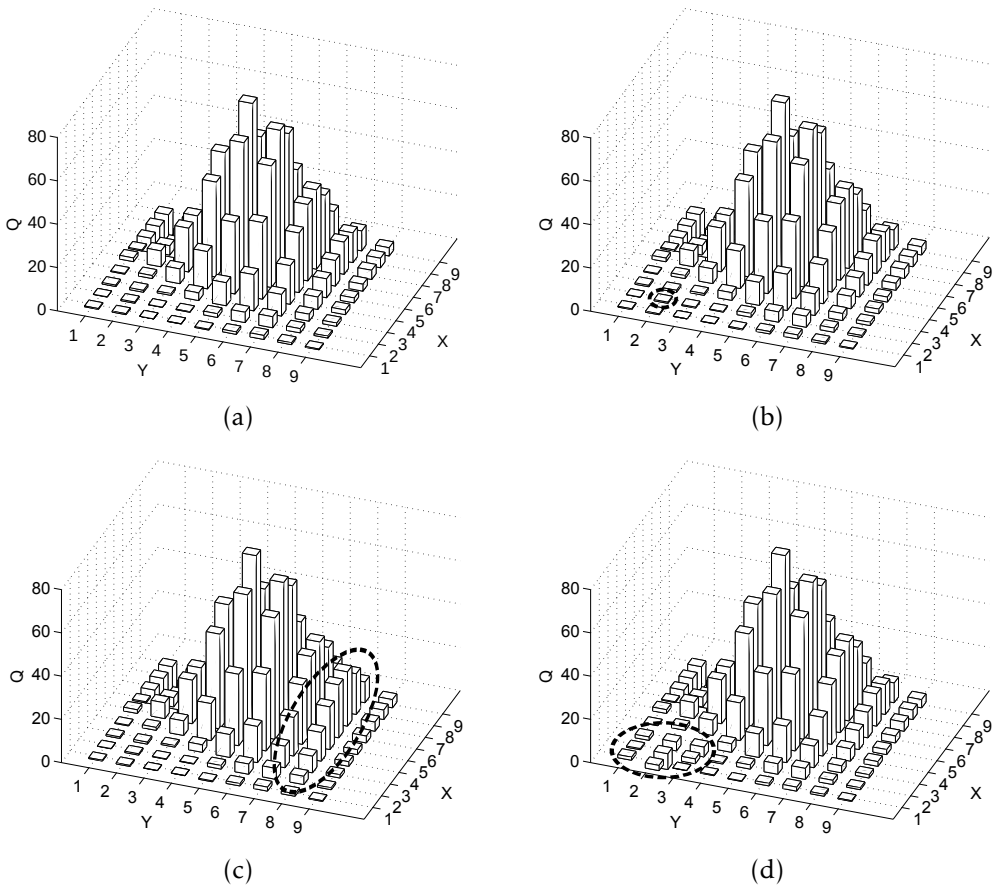
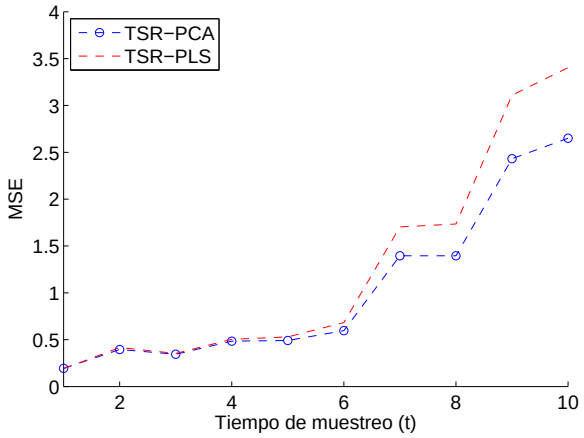


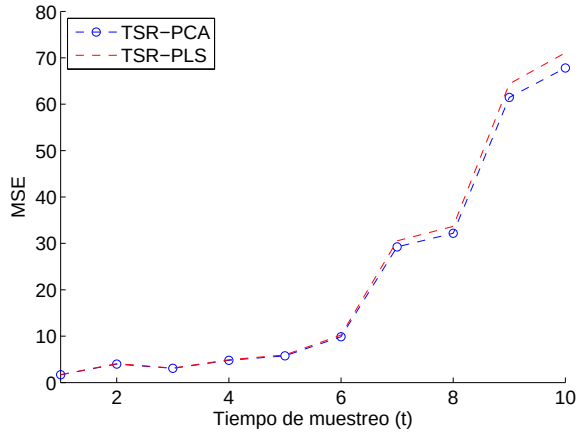
Figura 3.26: Perfil Q después de la imputación de datos TSR-PCA con modelos locales: (a) muestra el perfil original del fuego sin la presencia de ataque alguno; (b), (c) y (d) se corresponden con los resultados obtenidos después de la imputación de datos para los escenarios de ataque ADR, AMR y ALR, respectivamente. Los sensores afectados y sus valores recuperados se remarcan con línea discontinua.

sobre el MSE cometido, siendo de un 97,25 % en el caso de AMR y un 96,61 % de mejora en el caso ALR. Si observamos detenidamente los resultados obtenidos para cada uno de los métodos de recuperación, TSR-PCA y TSR-PLS, ambos proporcionan resultados similares. Esto es debido a que ambos métodos están trabajando en condiciones óptimas para la recuperación, puesto que ya se seleccionó el número adecuado de PC para cada uno.

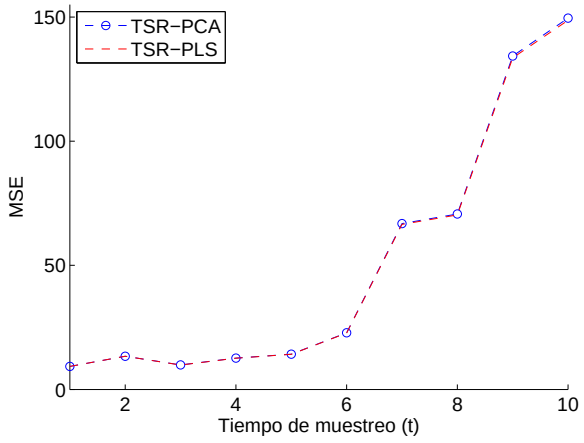
De igual forma a como se procedió en la sección anterior, se explora la evolución del MSE conforme lo hace el fuego. Este comportamiento se ilustra en la Figura 3.27 de manera similar a la Figura 3.15(a) para el caso el modelado global. Además de la tendencia incremental de nuevo observada para el MSE, se observa claramente también aquí la mejora en el rendimiento de la predicción de los datos que introducen los modelos locales, siendo escasa la diferencia entre el error MSE cometido al inicio



(a)



(b)



(c)

Figura 3.27: Evolución del MSE durante las 10 primeras observaciones obtenidas empleando modelos locales y considerando cada una de las situaciones de ataque: (a) ADR, (b) AMR y (c) ALR.

del fuego y esta misma medida en el décimo instante de muestreo a pesar de una clara acción del fuego. Este comportamiento tiene su explicación en el hecho de que los modelos globales son más sensibles al fuego. Esto es, los sensores afectados por el foco de fuego se incluyen en el modelo global junto con aquellos que no lo están, estimando los valores de los sensores con otros que difieren en sus condiciones. En su lugar, en el modelo local los sensores considerados se limitan al vecindario, tal que el valor de un sensor afectado o no por fuego se recupera a través de sensores bajo las mismas condiciones.

Finalmente, se evalúa la progresión del MSE en función del número de sensores que son atacados simultáneamente para el escenario ADR. Esta evolución se muestra en la Figura 3.28. Como ocurría con los modelos globales (ver Figura 3.15(b)), el error MSE crece conforme lo hace el número de sensores atacados. De nuevo, se observa un comportamiento lineal de crecimiento. En este caso, la constante multiplicativa es mayor que para los modelos globales, mostrando que la selección aleatoria de nodos adyacentes (desde 1 hasta 10) tiene un mayor impacto en el rendimiento de los modelos locales.

En este punto, y a partir de la discusión y resultados previos, el método propuesto de imputación junto con la estructuración adecuada de los datos en modelos locales, proporciona un alto rendimiento en la recuperación incluso en condiciones adversas, como son: un entorno cambiante y dinámico (el fuego evoluciona en el tiempo) y ante un número razonable de sensores atacados simultáneamente (alrededor de un 12% del total). Consecuentemente, nuestra propuesta robustece este tipo de redes ante ataques a la integridad de los datos y, por ende, aboga por su supervivencia.

3.9. Aplicación en entornos reales: proyecto LUCE

En esta sección se contempla el uso de un despliegue real WSN con el objetivo de corroborar la validez y aplicabilidad de los resultados obtenidos anteriormente en escenarios de simulación.

3.9.1. Descripción del entorno real

Uno de los escenarios de aplicación del proyecto *SensorScope* [128], desarrollado por la EPFL (*École Polytechnique Fédérale de Lausanne*), es LUCE (*Laussanne Urban Canopy Experiment*) [122]. El proyecto LUCE, instalado en el campus de la EPFL en 2006, consiste en el despliegue y monitorización a través de estaciones meteorológicas que forman una WSN. LUCE tiene como objetivo el estudio de las interacciones entre entornos urbanos y la más baja atmósfera, para el mejor entendimiento de la micrometeorología y el transporte atmosférico en estos entornos. Este proyecto se

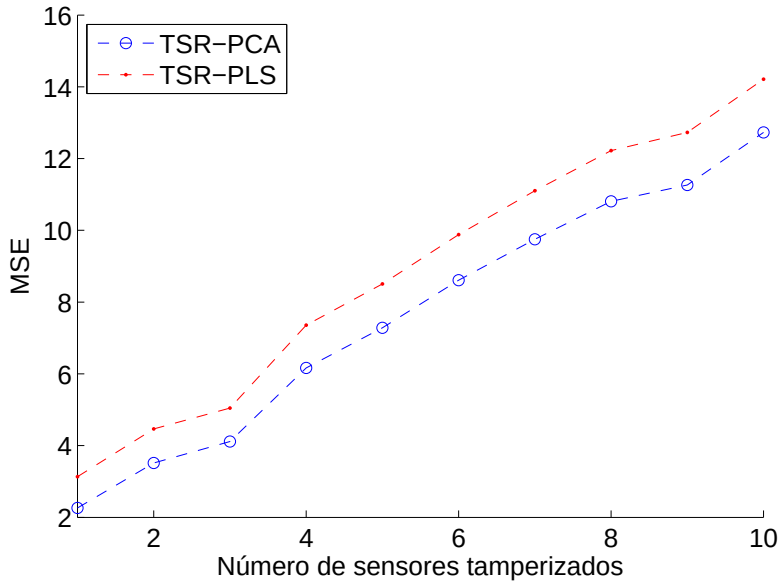


Figura 3.28: Evolución del MSE en presencia de fuego conforme crece el número de sensores afectados, empleando modelos locales. Solo se considera la situación de ataque ADR.

compone de 100 estaciones meteorológicas *SensorScope* cubriendo aproximadamente todo el área que abarca el campus de la EPFL (unos 500 m^2).

Cada una de las estaciones mencionadas posee varios sensores para la medición de parámetros clave del entorno que les rodea, proporcionando una alta densidad de medidas espaciales y temporales de dichos parámetros. Entre otros, poseen sensores de temperatura, humedad y velocidad del viento. Las mediciones de cada estación son enviadas por GPRS a la CU en intervalos de tiempo de 30 segundos.

De modo comparativo, la Tabla 3.4 expone las características principales tanto del entorno de simulación ideado con anterioridad en este capítulo como del despliegue real LUCE. Ambos cubren un área similar, con un número también similar de sensores utilizados, y son capaces tomar mediciones de temperatura cada cierto tiempo. A pesar de esta similitud, la diferencia principal es que LUCE obtiene mediciones de temperatura reales. De esta manera, este despliegue se considera un *test bed* adecuado para probar, evaluar y concluir la validez y aplicabilidad práctica del sistema propuesto.

A través del sitio web del proyecto LUCE [122] se pueden descargar los conjuntos de datos recogidos desde noviembre de 2006 hasta mayo de 2007. Tras el correspondiente estudio y análisis del contenido y estructura del conjunto de datos, se establece que los datos correspondientes a la temperatura recogida durante el periodo comprendido entre el 1 de enero y el 31 del mismo mes para el año 2007, son los más completos del conjunto, suponiendo un total de 80.000 muestras por sensor. Se consideran 61 sensores distribuidos por el área monitorizada tal y como se

Escenario	Características	Valor
Simulado	Área (m^2)	1000
	Nº de sensores	81
	Medición de la Tª ambiente	Sí
	Otras mediciones	No
	Alta densidad espacial y temporal	Sí
LUCE	Área (m^2)	500
	Nº de sensores	100
	Medición de la Tª ambiente	Sí
	Otras mediciones	Sí
	Alta densidad espacial y temporal	Sí

Tabla 3.4: Comparativa entre el escenario de simulación ideado y el despliegue real del proyecto LUCE.

muestra en la Figura 3.29. Adicionalmente, y en la misma figura, se resaltan los 8 sensores más cercanos a uno dado cuyo identificador es el 100.

3.9.2. Monitorización y detección de anomalías

Al igual que en la Sección 3.6.2, aquí se utilizará la misma metodología en cuanto a la monitorización y detección de anomalías se refiere, aplicando también modelado global. Se escogen los 20 primeros días del rango de datos seleccionado como conjunto de datos de calibración del modelo PCA. Para corregir las fluctuaciones de temperatura que se producen a lo largo del día (la temperatura comienza a subir por la mañana, se hace máxima en la mitad del día para empezar a bajar por la tarde/noche) se sustrae la media de temperatura del día concreto en todas sus mediciones.

En la Figura 3.30 se observan las muestras correspondientes al conjunto de datos de calibración o modelo de calibración (círculos oscuros) después de la eliminación de *outliers*. Una vez establecidos los límites de control, todas las subsecuentes observaciones (triángulos invertidos) que se corresponden con el conjunto de datos de test se clasifican como normales, excepto una de ellas (resaltada con línea discontinua) que se corresponde con una anomalía generada de manera artificial.

Es de notar que para este caso no existe influencia alguna de fuego. Por lo tanto, cualquier anomalía que se produzca viene motivada, muy probablemente, por la pérdida de datos en un sensor o por el mal funcionamiento del dispositivo. Ambas

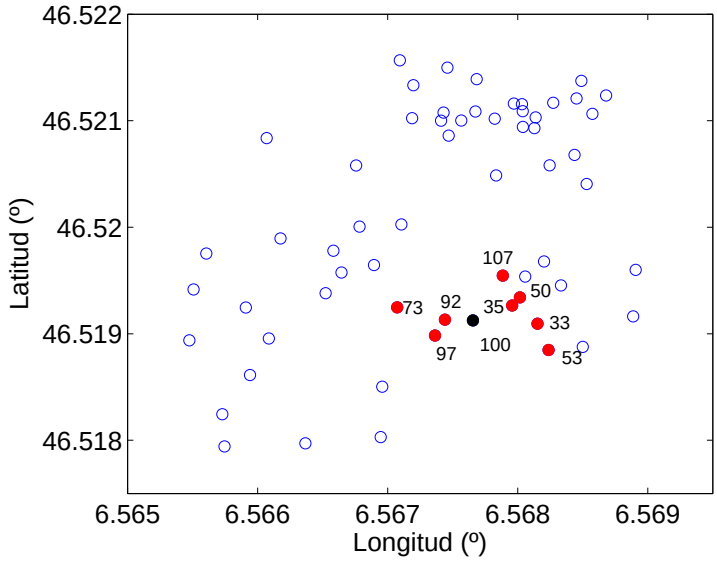


Figura 3.29: Distribución de 61 de los sensores del proyecto LUCE considerados en la experimentación real. Como ejemplo, también se resaltan los 8 sensores más cercanos a uno dado (el sensor con ID=100).

fuentes de anomalías puede tener su origen en una actuación maliciosa. Para discernir el tipo de anomalía se utilizan los gráficos de contribución Q, al igual que se hacía en la monitorización de entornos simulados. La Figura 3.31(a) muestra el perfil Q producido por el ataque ADR, presentando una significante desviación en el sensor que está siendo atacado.

3.9.3. Recuperación de datos faltantes

Después de detectar la anomalía, y al igual que se hizo en la Sección 3.6.3, se aplica el método TSR-PCA de imputación como mecanismo de respuesta. La Figura 3.31(b) muestra el perfil Q como resultado de la recuperación usando modelado global. En cuanto al error MSE cometido, en la segunda columna de la Tabla 3.5 se pueden observar los resultados.

Hemos de remarcar que en el despliegue WSN LUCE no existe algoritmo de *routing* alguno, ya que la información de los sensores se envía directamente hacia la CU mediante un enlace GPRS. Por lo tanto, no tiene sentido aquí hablar del impacto del algoritmo de encaminamiento sobre el número de sensores afectados por un ataque o en la propia imputación.

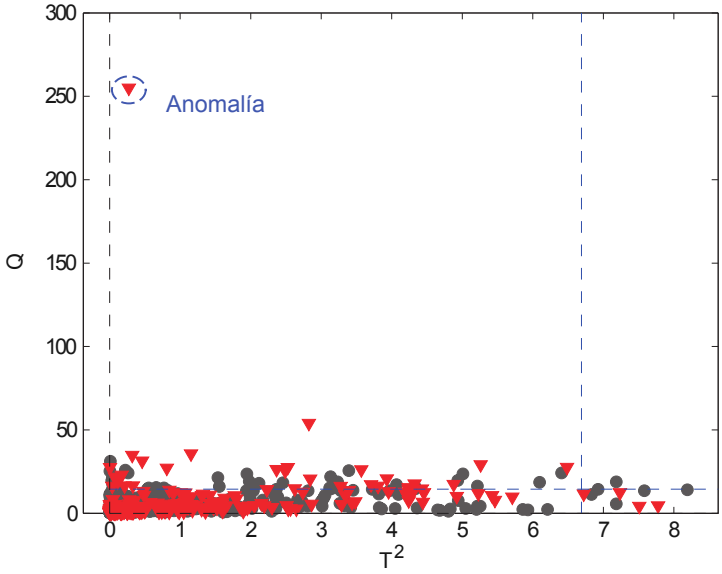


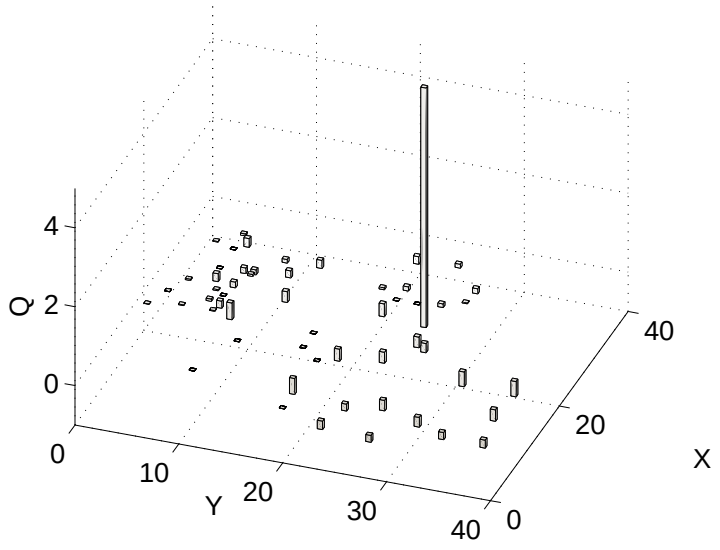
Figura 3.30: Gráfico de monitorización para el entorno real en donde se aprecian los datos iniciales de calibración (círculos oscuros), límites de control establecidos (líneas azules discontinuas) y las subsecuentes observaciones (triángulos rojos invertidos) clasificadas como eventos normales. Se resalta, en línea discontinua azul, una anomalía generada de forma artificial.

	Modelo global	**Modelo Local**	
Escenario de ataque	**MSE (TSR-PCA)**	**MSE (TSR-PCA)**	**MSE (TSR-PLS)**
ADR	0,1051	0,1081	0,1030

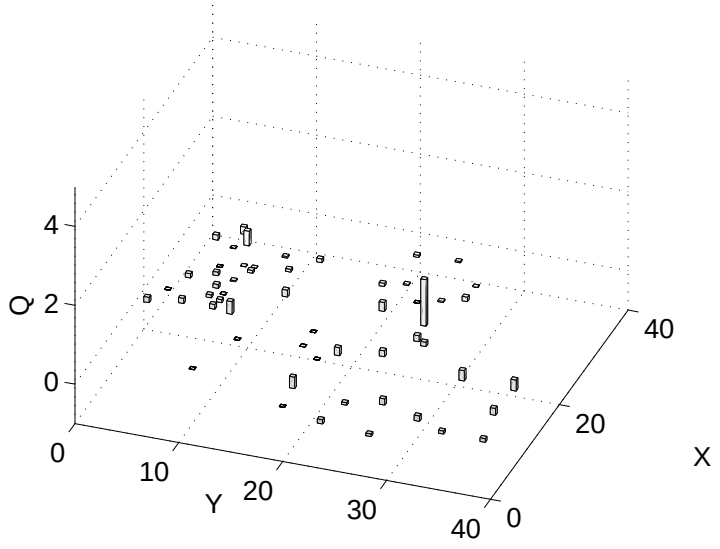
Tabla 3.5: Comparativa MSE en la imputación de datos para el ataque ADR empleando los métodos TSR-PCA y TSR-PLS con modelado local y global en el entorno real LUCE.

3.9.4. Recuperación de datos faltantes empleando modelos locales en entornos no regulares

Como ya describió en la Sección 3.8.1, es relativamente sencillo estructurar y organizar los datos provenientes de una WSN que posee una topología regular. Esto se hacía sin más que contar con los 8 sensores que rodean al afectado. Sin embargo, definir qué sensores son los más cercanos al afectado en escenarios no regulares no es tarea fácil. Por lo tanto, es necesaria la propuesta de un método adecuado para determinar dichos sensores. Para abordar esta problemática se computan los valores MSE variando el número de sensores más cercanos seleccionados a la hora de construir el modelo local. En la Figura 3.32 se puede observar que el número de



(a)



(b)

Figura 3.31: Perfiles Q para la situación ADR: (a) sensor atacado y (b) sensor recuperado.

sensores cercanos a considerar como valores válidos en la imputación está alrededor de 6 u 8, ya que seleccionar un número mayor proporciona resultados similares. En consecuencia, y con objeto de comparar resultados con el escenario regular simulado,

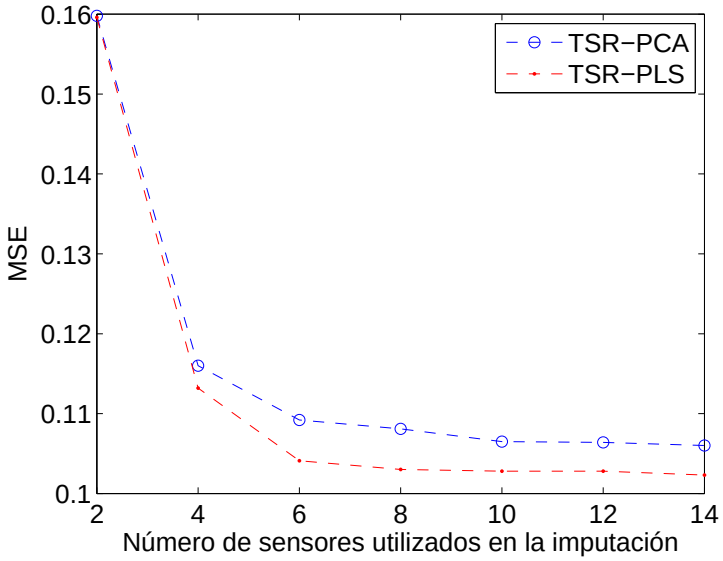


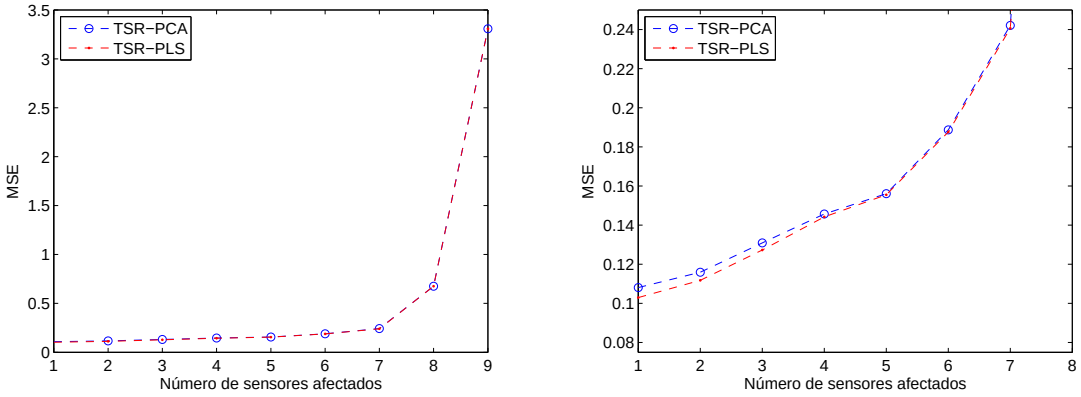
Figura 3.32: Evolución del MSE con el número de sensores más cercanos al afectado considerados como valores válidos para el método de recuperación.

se seleccionarán los 8 sensores más cercanos al afectado para llevar a cabo el proceso de recuperación en entornos WSN con topologías no regulares.

Un aspecto a tener en cuenta es la distribución de los sensores más cercanos alrededor del afectado es el hecho de cómo se distribuyen estos. Si la mayoría se distribuye de manera no homogénea, se presupone una eficacia de predicción peor que en el caso homogéneo. De esta manera se tiene una cierta incertidumbre en la predicción dependiente de dos factores principales: (*i*) la distancia de los sensores más cercanos al afectado, y (*ii*) cómo se distribuyen alrededor de él.

Los resultados de imputación que se obtienen a partir de los datos del proyecto LUCE empleando modelos locales son similares a los de la Figura 3.31(b). La Tabla 3.5 muestra los resultados numéricos MSE asociados. En este caso, son similares a aquellos obtenidos con modelos globales, principalmente debido a la existencia de una alta correlación de los datos del proyecto LUCE. Si se computan los coeficientes de correlación que muestran la relación entre variables, obtenemos que el mínimo valor se corresponde con $0,89$, indicativo de la alta correlación entre los sensores.

Otro experimento interesante y de utilidad con el fin de evaluar la robustez del método de imputación es comprobar cómo se comporta este conforme el número de sensores comprometidos aumenta. Para observar este comportamiento, se aumenta de manera secuencial el número de sensores atacados desde 1, el que originalmente se ataca, hasta alcanzar sus 8 sensores más cercanos. Por ejemplo, 1 significa que solo un sensor está siendo atacado; 2 significa que también se ataca su sensor más cercano, y así sucesivamente hasta afectar a los 9 implicados. La Figura 3.33 ilustra



(a) (b)

Figura 3.33: Evolución del MSE con el número de sensores atacados. La subfigura (a) muestra dicha evolución mientras que en (b) se presenta un *zoom* para los 7 primeros atacados.

la evolución del error MSE a medida que el número de sensores afectados crece. Se observa que dicho parámetro no varía de manera significativa cuando el número de nodos afectados es menor o igual a 7. Este dato corrobora la robustez y resistencia del método de respuesta y recuperación ante ataques de *data tampering* que, a priori, pudieran afectar de manera peligrosa al normal funcionamiento de la WSN.

3.10. Conclusiones del capítulo

A lo largo del presente capítulo se ha presentado el uso de las técnicas de análisis multivariante y su empleo en la monitorización, detección de anomalías y respuesta ante la pérdida o alteración, intencionada o no, de la información recogida en escenarios WSN. En concreto, se estudia y valida su empleo en entornos críticos de actuación como es la lucha y extinción de incendios, corroborando su aplicabilidad y validez de uso tanto en despliegues WSN simulados como reales.

Para la monitorización y detección de anomalías se utilizan esquemas basados en soluciones MSPC, proponiendo técnicas de recuperación de datos faltantes (TSR) para mitigar los efectos perniciosos ocasionados por la pérdida o modificación de información. En concreto, se considera que las anomalías acontecidas son debidas a actuaciones maliciosas específicas provenientes de ataques a la integridad de los datos, como el de *data tampering* a nivel físico; aunque este esquema podría ser aplicado frente ataques como el de *blackhole*, orientado a la perjudicar la disponibilidad de la red y los servicios que esta ofrece.

A través de variados y extensivos experimentos, sobre todo enfocados a la evaluación del método propuesto de imputación de datos, se determina que la organización de los datos empleada para la generación de modelos multivariantes tiene un impacto muy relevante sobre el rendimiento de estas técnicas. En este sentido, se diseña y experimenta con diferentes organizaciones de los datos, concluyendo que aquellos que disponen de más información correlada en relación al sensor que está siendo afectado o atacado, obtienen los mejores resultados en lo que se refiere a la efectividad en la imputación. A partir de este estudio se deduce que si se reduce la localidad ofrecida por el modelo local, se obtienen mejores resultados frente al modelado global y global dinámico.

Por otro lado, y de manera novedosa, se estudia el condicionamiento que supone el empleo de determinados algoritmos de encaminamiento de cara al rendimiento de los métodos de recuperación de datos multivariante. Para ello se proponen diferentes estrategias de *routing* que, dependiendo del modelado utilizado, contribuirán a aumentar la eficiencia de predicción. Se determina también cómo soluciones que incrementan la robustez y fiabilidad a la hora de proporcionar los servicios ofrecidos en el contexto de la seguridad, son contraproducentes de cara a la conservación y el consumo eficiente de recursos en la red, otro aspecto muy importante en este tipo de redes. Por esta razón, las soluciones de seguridad desarrolladas deberían contribuir a la consecución de entornos seguros y eficientes desde el punto de vista energético de manera que contribuyan a la adición de nuevas capacidades de supervivencia al sistema. Dicho balance no siempre es fácil de conseguir, constituyendo un verdadero reto a tener en cuenta en futuras propuestas.

Publicaciones relacionadas

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

- **R. Magán-Carrión**, J. Camacho Páez y P. García-Teodoro. "Multivariate Statistical Approach for Anomaly Detection and Lost Data Recovery in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–20, mayo 2015.
- **R. Magán-Carrión**, F. Pulido Pulido, J. Camacho Páez y 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.
- **R. Magán-Carrión**, J. Camacho Páez y P. García-Teodoro. "Supervivencia en redes de sensores mediante técnicas multivariantes," *12th Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2012)*, pp. 315–320, sept. 2012.

Capítulo 4

Optimización del posicionamiento de nodos *relay*

Contenido

En general, la planificación, la arquitectura y la distribución de cada uno de los nodos de una red está directamente determinada por el o los objetivos de rendimiento fijados. Un ejemplo claro es la optimización de la distribución de los nodos en redes WSN. Comúnmente, los nodos se emplazan de manera determinista con el fin de ofrecer la máxima cobertura sobre el área que se va a monitorizar. Por otro lado, determinados escenarios como los militares y situaciones de rescate o emergencia, no resultan tan simples en la determinación de la posición de los nodos. En estos casos, la distribución se produce de manera un tanto aleatoria, siendo solo unos pocos nodos sobre los que se tiene el adecuado control para ubicarlos correctamente [129]. Estos últimos, a los que se les denomina nodos *relay* o RN (*Relay Node*) por su función principal como retransmisores de la información, se localizan adecuadamente para

estructurar la red y conseguir el rendimiento deseado. Por ejemplo, y además de a la propia cobertura de la red, la posición de dichos nodos puede afectar a la conectividad, al retardo en las comunicaciones, al *throughput* o al consumo de energía, entre otros; aspectos todos ellos plausibles para ser considerados como objetivos de rendimiento del sistema.

El problema de la optimización del posicionamiento de nodos se considera difícil de solventar, y todavía hoy es considerado todo un reto ingenieril. Tal es así, que la mayoría de los trabajos propuestos ofrecen heurísticas que solo consiguen aproximar la solución óptima [129]. En el contexto de las redes ad hoc, la mayoría de soluciones se enmarcan dentro de entornos estáticos en los que la topología de la red no cambia con el tiempo y en donde se realiza un distribución lógica inicial en función de los objetivos de rendimiento deseados, permaneciendo así a lo largo del tiempo. Por otro lado y en menor medida, existen propuestas que abogan por la re-localización dinámica de los nodos restringida en cierta medida por el entorno en el que se integran. En este tipo de escenarios la topología de red cambia dinámicamente, motivando que la situación de optimalidad presentada por la red también varíe con el tiempo [129]. A diferencia de los escenarios estáticos, los dinámicos permiten la adaptación del sistema a situaciones cambiantes para proveer, por ejemplo, tolerancia a fallos [130–133]. La presencia de fallos o mal funcionamiento en uno o varios nodos hace que cambie la topología de red, lo que supone un impacto importante en entornos estáticos sobre todo de cara a su supervivencia, su adaptabilidad y su recuperación ante dichos eventos [88]. Es de notar que estos eventos no solo pueden deberse a un fallo fortuito de un nodo o nodos de la red, sino que, desde el punto de vista de la seguridad, pueden tener su origen en actuaciones maliciosas. Un ejemplo de escenarios ad hoc dinámicos son las redes MANET [134], en donde ataques comunes como el *blackhole* provocan efectos similares al mal funcionamiento o fallo de un nodo de la red.

Claramente, la posibilidad de re-ubicar una serie de nodos en función de los cambios producidos en el entorno es relevante a la hora de mantener o incluso mejorar el rendimiento que ofrece la red o sistema. Escenarios que requieren soluciones de posicionamiento de nodos capaces de adaptarse a las variaciones que se producen a su alrededor, implican la utilización de herramientas adicionales con respecto a entornos estáticos. Por ejemplo, sería necesaria la capacidad de evaluar el dinamismo que posee la red y/o anticiparse o inferir los próximos cambios que se producirán. Adicionalmente y no menos importante, está la necesidad implícita de la actuación coordinada en el posicionamiento de los nodos. Esto es, el movimiento de un nodo hacia una determinada posición necesariamente condicionará a su vez el movimiento y posicionamiento de los demás. Cómo evaluar y gestionar la dinámica que ofrece la red, así como el necesario control coordinado de los nodos a posicionar son, sin duda, dos aspectos relevantes e inherentes al problema de posicionamiento de nodos *relay* en redes MANET.

Son muchas las situaciones en las que el número de nodos *relay* no es un parámetro configurable; es decir, normalmente se dispone de un número específico y limitado de aquellos. Por este hecho, surgen algunos problemas prácticos en cuanto a la optimización de su posición con el objetivo de maximizar el rendimiento de la red. Este es el caso de, por ejemplo, su aplicabilidad a entornos altamente dinámicos donde el número óptimo de nodos *relay* cambia a lo largo del tiempo. De hecho, buscar la solución ideal con un número prefijado de estos supone todo un reto. Por otro lado, es común aplicar restricciones que condicionan las posibles posiciones que pueden tomar los *relays*. Sin embargo, no tener en cuenta las restricciones del entorno puede llevar a una situación en la que las posiciones resultantes de la optimización sean físicamente imposibles de implementar.

Para abordar la problemática descrita anteriormente en entornos que varían con el tiempo, se propone en este capítulo una solución cuyo fin es dar respuesta a dos preguntas intrínsecas al problema del posicionamiento de nodos *relay*: (*i*) ¿cuál es la posición optimizada de dichos nodos en un determinado momento? y (*ii*) ¿cómo han de moverse para alcanzar dicha posición? Dentro del contexto de las redes MANET, el esquema ideado solventa las cuestiones anteriores, siendo su objetivo principal maximizar dos aspectos relevantes en este tipo de redes: la conectividad y el *throughput* cuando se dispone de un número limitado y establecido de *relays*. Recuperar o mantener la conectividad es especialmente relevante en este tipo de redes ya que, debido a su inherente dinamismo, es constante la aparición de particiones o desconexiones.

La forma más sencilla de acometer el problema es haciéndolo de manera separada. Por un lado, se afronta la optimización de la posición de los nodos y por otro su movimiento controlado. Siguiendo esta filosofía, en el presente capítulo se presenta una estructura modular y flexible provista de dos bloques principales que conforman el núcleo o *core* del sistema. En concreto, la solución se basa en el trabajo realizado por Dengiz *et al.* en [131] y solventa serias deficiencias encontradas en el esquema de referencia tanto para la solución de posicionamiento como para la de control del movimiento. Como en [131], dicha solución se apoya en el uso de algoritmos de optimización basados en el conocido PSO (*Particle Swarm Optimization*) [135], así como metodologías inspiradas en MPC (*Model Predictive Control*) [136] para el control del movimiento. El algoritmo PSO, y en general aquellos enmarcados dentro del grupo de los denominados *evolutionary algorithms*, son apropiados para su uso en entornos dinámicos y cambiantes como el que se nos presenta [137]. Este tipo de heurísticas solventan la complicada tarea de la obtención de soluciones globales óptimas cuando el espacio de búsqueda es muy extenso debido a las características del entorno en donde se contextualiza el problema, como es el caso que nos atañe.

A lo largo del capítulo se describirá el problema detalladamente mediante la correspondiente formulación matemática (obviada en [131]), para después indagar en la heurística propuesta multietapa así como en los algoritmos y procedimientos

utilizados. Para corroborar la viabilidad y eficacia de la solución, se diseñará y ejecutará un extenso grupo de experimentos no solo en entornos simulados, sino también en entornos reales MANET.

Si bien es cierto que nos centraremos en entornos dinámicos, no es difícil observar que, desde el punto de vista modular que se plantea, la solución para el problema del posicionamiento optimizado de los nodos en un determinado momento temporal es aplicable a entornos estáticos. De esta manera y adicionalmente, se incluye la correspondiente experimentación sobre escenarios estáticos ad hoc, con el fin de contrastar la eficacia de la solución global en este tipo de entornos.

Lo que resta de capítulo se estructura de la siguiente manera. Primeramente, en la Sección 4.1 efectuaremos un estudio sobre las principales soluciones encontradas en la literatura que abordan el problema del posicionamiento de nodos *relay*. En la Sección 4.2 se introducen los fundamentos y limitaciones de una de las anteriores soluciones sobre la que se apoya la propuesta del presente capítulo. A lo largo de la Sección 4.3 se describen formalmente los fundamentos matemáticos en los que se basa la solución de posicionamiento propuesta. Todo este desarrollo matemático se lleva a la práctica a través de la implementación de un sistema de posicionamiento de nodos *relay* (DRNS) que se describe en profundidad en la Sección 4.4. En la Sección 4.5 se presenta el entorno de simulación propuesto así como los experimentos y resultados que prueban la viabilidad de la solución. A lo largo de la Sección 4.6 se demuestra la aplicabilidad y validez práctica del sistema DRNS cuando este se despliega en entornos reales. Finalmente, las conclusiones del capítulo se exponen en la Sección 4.7.

4.1. Ubicación de nodos *relay*: técnicas, esquemas y soluciones adoptadas

Dentro del extenso conjunto de estrategias de posicionamiento de RN, se pueden distinguir dos grupos o categorías principales que aplican al problema: aquellas propuestas que abogan por la tolerancia a fallos y las que intentan maximizar y/o recuperar la conectividad de la red. Fuera de esta clasificación se pueden encontrar soluciones con objetivos heterogéneos, como por ejemplo aquellas cuyo fin principal es acotar el retardo en las comunicaciones, otras que maximizan la cobertura obtenida, las que preservan el tiempo de vida de la red o simplemente las soluciones que limitan la localización de los RN a ciertas áreas o regiones físicas.

Siguiendo la clasificación anterior, se exponen a continuación los sistemas, soluciones y propuestas encontradas en la literatura especializada para el problema tratado.

4.1.1. Ubicación de nodos *relay* para conseguir redes conectadas y tolerancia a fallos

La mayoría de las soluciones de posicionamiento de nodos *relay* comparten un objetivo común: obtener redes k-conectadas[1] usando el mínimo número de RN. Por ejemplo, si se consideran objetivos de tolerancia a fallos, k debería ser mayor o igual a 2. Sin embargo, para conseguir una red con solo conectividad total bastaría con $k = 1$. Se dice entonces que la red está conectada, simplemente. Dentro de esta categoría de estrategias se encuentra una gran variedad de trabajos ([138–144], entre otros). Por ejemplo, en [138] los autores proponen soluciones cuyo objetivo es obtener redes conectadas tanto *single-tiered* como *two-tiered*[2] minimizando el número de RN desplegados. Para conseguir los dos objetivos anteriores, son dos los algoritmos considerados, diferenciados según el grado de aproximación a la solución óptima que presentan. Estos son los algoritmos *7-approximation* y *(5+ϵ)-approximation*, respectivamente. La extensión a este último trabajo se propone por los mismos autores en [139]. En este caso se idea un algoritmo aproximado para redes 2-conectadas que, al igual que su predecesor, se emplea tanto en esquemas *single-tiered* como *two-tiered*. De manera general, en [140] se obtienen redes k-conectadas siendo k un parámetro configurable. Cualquiera que sea el valor de k, los autores aseguran que su algoritmo llega a una solución aproximada en un tiempo polinomial. Un enfoque similar es el propuesto en [142], en donde los autores son capaces de encontrar el número mínimo de RN para conseguir redes k-conectadas. Mediante el empleo de un paso previo de selección, esta heurística mejora la propuesta anterior. Durante esta selección se retiran determinados RN de tal manera que no se pierda la k-conectividad que posee la red. En el trabajo [141] se persigue el objetivo de tolerancia a fallos en redes WSN heterogéneas, donde los sensores presentan diferentes radios de cobertura. Se obtiene una solución aproximada con el fin de conseguir redes conectadas con $k \geq 1$, teniendo en cuenta el grado de tolerancia a fallos deseado (resistencia total o parcial ante fallos) y el flujo de comunicación (en uno o varios sentidos). En la referencia [145] se propone una manera eficiente de conseguir el nivel de conectividad deseado teniendo en cuenta la longevidad de la red en lo que se refiere a la energía consumida. Al igual que en el anterior trabajo, los RN tienen restringida su ubicación en ciertos puntos candidatos preestablecidos. En aras de conseguir tolerancia a fallos, su principal objetivo es conseguir una red 2-conectada. Las referencias [146, 147] discuten soluciones similares.

[1] Se dice que una red está k-conectada si, eliminando un conjunto $k-1$ cualquiera de nodos, sigue existiendo al menos un camino que permite la conexión entre ellos, es decir, la red permanece totalmente conectada.

[2] En el contexto de redes WSN, una solución *n-tiered* se basa en el empleo de n capas de conexión o redes diferentes. Por ejemplo, es común que los RN formen una red o capa conectada que difiere de la de los propios sensores que conforman la WSN, y cuyo objetivo es proveer conectividad y/o tolerancia a fallos a esta última, disgregando así la función de cada una de ellas.

4.1.2. Ubicación de nodos *relay* para la recuperación/optimización de la conectividad

Aunque la disposición de una serie de nodos con el fin de proveer redes tolerantes a fallos, y por ende mantener la conectividad a lo largo del tiempo, son sin duda aspectos relevantes a tener en cuenta para el diseño y aplicación de esquemas de posicionamiento de RN, recuperar la conectividad perdida, cualquiera que fuera su motivo (fallos, desconexiones, acciones maliciosas, etc.), supone también un reto que se antoja difícil e interesante al mismo tiempo. Este problema se presenta especialmente en entornos dinámicos, en donde el movimiento de los propios nodos hace que aparezcan particiones en la red de manera inesperada y fortuita. Para abordar este asunto, los autores en [148] tratan de restablecer la topología de red perdida debido a los movimientos y desplazamientos de los nodos. De manera similar, en [149] se aborda cómo solventar problemas de conectividad en escenarios de índole militar. En este caso, la continuidad en la prestación de los servicios que oferta la red se soporta a través de la localización de FAP (*Flying Aerial Platform*) que hacen las veces de RN. Para ello, hacen uso de técnicas de SA (*Simulated Annealing*) con el fin de encontrar un mínimo global para el número de FAP empleado. Este último trabajo se asemeja al expuesto en [150]. Alfaqdhly *et al.* [151] investigan la reubicación óptima de nodos que recupera la conectividad ante el fallo de uno o varios de ellos. Para este fin propone una solución ILP (*Integer Linear Programming*) que trata de maximizar la cobertura a la vez que minimiza la distancia entre nodos. Los mismos autores introducen en [152] el algoritmo LDMR (*Least Distance Movement Recovery*) para mitigar los efectos adversos sobre el rendimiento de la red provocados por el fallo de uno o varios nodos. LDMR propone un enfoque distribuido que trata de movilizar uno o varios nodos hacia la zona afectada sin que ello suponga introducir más desconexiones en la red. Dentro del contexto de la teoría de grafos, a estos nodos se les denomina *non cut-vertices*. En la misma línea, la referencia [153] propone un algoritmo distribuido denominado RIM (*Recovery through Inward Motion*). RIM recupera la conectividad perdida motivada por el fallo de un nodo mediante la reubicación de sus vecinos.

La referencia [154] plantea dos enfoques distintos de relocalización de RN que, al igual que en el trabajo anterior, se utilizan para restaurar la conectividad, en este caso en redes WSN. El primero se basa en la teoría de fuerzas magnéticas. La adición de RN hace que se dispersen los nodos por todo el área de la red consiguiendo, al final, un estado de equilibrio en el que las fuerzas de atracción y repulsión implicadas permanecen estables. Con respecto al segundo enfoque, este se centra en la teoría de juegos. Se establecen líderes para cada partición, priorizando algunas particiones sobre otras de cara a la recuperación de la conectividad. El algoritmo finaliza cuando se alcanza el equilibrio Nash.

Lin *et al.* [155] abordan el problema dynRNP (*dynamic Router Node Placement*) considerando clientes *mesh* en escenarios WMN. Tanto los clientes como los *routers mesh* son móviles, agravando el problema el hecho de que los primeros pueden desactivar su interfaz inalámbrica cuando deseen. El objetivo de su propuesta es maximizar la conectividad de la red y la cobertura de los clientes, estimada la primera como el tamaño del mayor subgrafo conseguido en la red y la segunda como la capacidad de un *router* para dar servicio al mayor número de clientes posible. Se prioriza la atención a aquellos clientes cuyo peso o ponderación es mayor de acuerdo a los objetivos anteriores. Para resolver esta ponderación, los autores se apoyan en algoritmos bioinspirados; en concreto, utilizan uno basado en el comportamiento de las colonias de murciélagos o BA (de *Bat-inspired Algorithm*). Dicho algoritmo simula el método de ecolocalización de estos vertebrados para encontrar la mejor solución de acuerdo al propósito del sistema.

En [156], los autores intentan localizar el menor número de RN dentro del área que cubre una red de sensores, tal que cada uno de estos pueda comunicarse con al menos un RN. A su vez, el conjunto de RN forma una red completamente conectada. Esto se lleva a cabo resolviendo dos problemas de optimización: CRNSC (*Connected Relay Node Single Cover*) y 2CRNDC (*2-Connected Relay Node Double Cover*). Ambos consideran solo cierta información disponible en lugar de todo el plano XY completo. Los autores en [157] localizan el mínimo número de RN que preserva la conectividad global restringiendo su ubicación a un conjunto determinado de lugares. Estos lugares se corresponden con la solución al problema del *minimum Steiner tree*. Siguiendo una idea similar, en [158] se implementa una solución que recupera la conectividad perdida. Se seleccionan, de manera iterativa, aquellos puntos del árbol Steiner que conectan al menos 3 particiones de la red. Se continúan eligiendo dichos puntos de forma sucesiva, finalizando el algoritmo cuando el árbol de Steiner posee un número menor a tres puntos. Después se localizan los RN sobre la línea que une cada partición y su punto Steiner más cercano.

4.1.3. Ubicación de nodos *relay* multiobjetivo

Aunque es difícil diferenciar entre tipos de soluciones de posicionamiento de *relays* a través del objetivo perseguido, ya que son muchas las ocasiones en las que este se difumina mezclando varias metas, se pretende agrupar en esta subsección aquellas propuestas que difieren en cierta medida de las anteriores en cuanto al fin buscado. Por ejemplo, los autores en [159] (propuesta mejorada de [160]) diseñan una WSN con encaminamiento *multi-hop* localizando el mínimo número de RN adicionales que facilitan la comunicación entre cada uno de los sensores y la CU. La posición de los RN debería asegurar que el retardo de los caminos que sigue la información entre la CU y los sensores se limita a ciertos valores preestablecidos. Los autores estudian la estructura de la proyección del poliedro asociado al problema y

desarrollan desigualdades *node-cut*. Una vez definida la formulación del problema, se implementa y ejecuta un algoritmo de tipo *branch-and-cut* para posicionar los RN de forma optimizada (DCRNPP, de *Delay Constrained Relay Node Placement Problem*), teniendo en cuenta limitaciones en el máximo retardo producido. Con el fin de conseguir objetivos de conectividad y cobertura al mismo tiempo (bi-objetivo), en [161] se presenta un algoritmo genético de optimización para el posicionamiento de RN en WMN. En la referencia [162] se propone un algoritmo que emplea PSO para determinar la mejor localización de una serie de nodos dentro de un escenario industrial en términos de fiabilidad de la red, uniformidad de carga, coste total y velocidad de convergencia. Basados en la optimización de la cobertura en WSN, los autores en [163] proponen una solución basada en PSO que minimiza las zonas sin cobertura existentes a través del uso de una función de coste que utiliza regiones de Voronoi. En el trabajo [132] se expone una solución al problema del posicionamiento de CU en WSN que modifica el algoritmo PSO para minimizar el máximo retardo provisto por cada camino de la red. En [164] se propone un esquema *minimax* con el objetivo de optimizar el radio de cobertura y la uniformidad en la distribución de los sensores de una WSN en aplicaciones de vigilancia.

Por otro lado, los autores de [130] proponen un algoritmo que emula fuerzas de atracción (por ejemplo, como las que surgen cuando se estira un muelle o resorte) y de repulsión (por ejemplo, fuerzas electrostáticas) presentes en la naturaleza. Con esta filosofía, los autores maximizan la cobertura ofrecida por la red minimizando la distancia que tienen que recorrer los nodos hasta alcanzar los puntos en donde las fuerzas se equilibran. Wang *et al.* [165] abordan el problema del posicionamiento de nodos en redes inalámbricas con el fin de asegurar la cobertura en escenarios en donde estos nodos se disponen longitudinalmente. El escenario de aplicación que proponen los autores es su despliegue dentro de túneles, minimizándose la distancia entre nodos en base a la maximización de la cobertura total obtenida. Los autores muestran cómo la densidad de nodos conseguida es menor que aquella obtenida aplicando la bien conocida solución de posicionamiento *triangular-lattice*. Los autores en [166] concretan el número y despliegue de una serie de dispositivos heterogéneos tal que el coste total de la red WSN se minimiza a la vez que se satisfacen restricciones relacionadas con el tiempo de vida de la red, cobertura y conectividad. Este trabajo se extiende en [167], [168] y [169] para el despliegue de una segunda capa (*second-tier*) de RN cuyo objetivo es balancear el tráfico usando el menor número posible de estos.

Como ha quedado explícitamente constatado anteriormente, existen numerosas propuestas de posicionamiento de RN. No obstante, son pocas las que consideran de manera conjunta maximizar la conectividad y el *throughput* ofrecidos por la red. Uno de estos trabajos es el que se propone en [131], donde los autores hacen uso de un procedimiento basado en PSO que, junto con el empleo de metodologías MPC, aboga por la consecución de los objetivos antes mencionados en un entorno específico como son las redes MANET. Los autores en [170] consideran cuestiones de rendimiento

y conectividad de la red. La solución ofrecida maximiza el PDR global obtenido a través de la propuesta de un modelo realista que tiene en cuenta aspectos como las interferencias del canal o áreas congestionadas para el posicionamiento de nodos *relay*. Sin embargo, aunque el *throughput* se puede ver como una medida indirecta de la conectividad, este no se contempla explícitamente en la optimización.

Es apreciable cómo la mayoría de trabajos tratan de buscar un óptimo para el número de RN que depende en gran medida del rendimiento u objetivos perseguidos. Este hecho es especialmente importante en entornos que varían con el tiempo, ya que el número de RN óptimo también variará. Adicionalmente, es usual que en entornos reales se disponga de un número reducido y limitado de *relays*, lo que limita considerablemente el uso de las anteriores propuestas en escenarios realistas.

La propuesta de posicionamiento que se presenta y discute a lo largo del presente capítulo trata de solventar la limitación anterior proponiendo una solución cuyo objetivo es la maximización conjunta de la conectividad y el *throughput* en la red.

4.2. Mejora de la conectividad y el *throughput* a través del empleo de nodos *relay* en MANET

En la sección anterior se introdujo un extenso abanico de soluciones que abordan el problema del posicionamiento de nodos *relay*, donde el principal objetivo es la mejora o recuperación de la conectividad. La aparición de desconexiones o particiones en la red se produce principalmente debido a la propia naturaleza dinámica de esta, a fallos en el funcionamiento de los nodos o incluso a actuaciones maliciosas que presentan similar impacto sobre la conectividad de la red[3].

Mantener, recuperar e incluso mejorar la conectividad en entornos que cambian con el tiempo, y sobre todo en aquellos especialmente críticos como escenarios militares o situaciones de emergencia, constituye un importante reto. Adicionalmente, una mala gestión podría tener un importante impacto no solo material sino también humano. De igual manera, aunque un poco menos relevante, es de interés conseguir maximizar el *throughput* ofrecido por el sistema de cara, por ejemplo, a la mejora del servicio al usuario o sistema final que lo demanda.

Derivada de los motivos anteriores, en el presente capítulo se propone una solución al problema del posicionamiento de nodos *relay* en redes MANET. La propuesta realizada parte de la idea ofrecida por los autores en [131], cuyo objetivo principal es

[3]En el ámbito de la seguridad en red los ataques *blackhole*, *dropping* o *selfish* [171] son comunes en redes MANET, teniendo un impacto sobre el rendimiento de la red similar al que puede producir el fallo de uno o varios nodos.

maximizar la conectividad en este tipo de redes. Así, en primer lugar, se describirá brevemente la solución DKS (cuyas siglas se corresponden con las iniciales de los nombres de los autores del trabajo) de referencia. Esta propuesta presenta graves deficiencias en su diseño que serán discutidas y después solventadas por el esquema propuesto en lo que sigue.

4.2.1. Solución DKS para el problema del posicionamiento de nodos *relay*

Son dos los tipos de nodos involucrados en la propuesta DKS: los nodos de usuario UN (*User Node*) y los nodos RN. Los primeros demandan los servicios que ofrece la red, mientras que el principal objetivo de los últimos es garantizar que los UN reciben el mejor servicio posible. Para ello se intenta maximizar la conectividad en todo momento. En un entorno inalámbrico, como es habitual en el caso de las redes MANET, diremos que dos nodos adyacentes están conectados (es decir, existirá un enlace entre ellos) si su distancia euclídea es menor o igual que c, siendo c el radio de cobertura de los nodos.

Además de maximizar la conectividad global ofrecida por la red, el sistema DKS también considera la maximización del *throughput*. Para ello se hace uso del algoritmo PSO [135], que utiliza varias funciones de coste u objetivo. En particular, PSO considera como parámetros principales de entrada: (*i*) la predicción de la posición de los UN H instantes de tiempo en adelante y (*ii*) la mejor solución obtenida justo en el instante de tiempo anterior al actual. Con estos dos parámetros, se itera sucesivamente comparando entre diferentes soluciones posibles (partículas) para encontrar aquella que optimiza la función de coste definida. En el contexto del problema, una partícula o solución se corresponde con una distribución de red diferente en la cual se tiene, por un lado, la predicción de las posiciones de los UN en $t+H$ y, por otro, la ubicación de cada RN. La localización de estos últimos se obtiene en cada iteración del algoritmo modificando las posiciones anteriores (mediante decrementos o incrementos de velocidad y/o dirección) con el fin de encontrar aquella configuración que maximice los objetivos perseguidos. Una vez que el proceso PSO termina, el algoritmo devuelve la mejor posición para cada RN, que consigue a su vez los mejores resultados en cuanto a conectividad y *throughput*. En el Apéndice B se detallan los fundamentos del algoritmo PSO.

DKS plantea tres funciones objetivo diferentes, usadas conjuntamente durante el proceso de optimización. Con la primera de ellas, O_1, se evalúa la conectividad global de la red. O_1 se define de la siguiente manera:

$$O_1 = \frac{2 \times \sum_{i,j \in U: j>i} z_{ij}}{|U| \times (|U|-1)} \tag{4.1}$$

donde U corresponde al conjunto de nodos UN de la red, representando $|U|$ el número de UN (cardinalidad del conjunto U) y $z_{ij} = 1$ si existe un camino (directo o mediante encaminamiento *multi-hop*) que conecta los nodos i-ésimo y j-ésimo; en otro caso, $z_{ij} = 0$.

Una segunda función, O_2, se encarga de obtener el mínimo *throughput* disponible, en otras palabras, aquel enlace más débil en estos términos. Solo se considera O_2 en la optimización en caso de que dos o más posibles soluciones bajo evaluación consigan conectar la red por completo, es decir, obtengan $O_1 = 1$. De esta manera, ya que es imposible mejorar la conectividad de la red (ya está totalmente conectada), se intenta mejorar el flujo de esta. O_2 se describe a través de la siguiente ecuación:

$$O_2 = \min_{i,j \in U: j>i} \{T(G,i,j) : T(G,i,j) > 0\} \tag{4.2}$$

donde $T(G,i,j)$ es el *throughput* máximo obtenido entre el i-ésimo y j-ésimo nodos de la red, representada esta como un grafo $G = (N,E)$ donde N se corresponde con el conjunto total de nodos (RN y UN) y E integra los enlaces (aristas) existentes en la red. Es preciso mencionar aquí que el máximo *throughput* que se obtiene entre dos nodos diferentes de la red está limitado por aquel enlace que presenta menor *throuhgput* en la ruta. En la propuesta DKS los autores aproximan el *throughput* de un determinado enlace entre dos nodos adyacentes i y j (w_{ij}) como una función de la distancia d_{ij} que los separa de la siguiente manera:

$$w_{ij} = \frac{1}{1 + e^{(10 \cdot d_{ij} - 0{,}5)}} \tag{4.3}$$

Como ya se discutió, las redes MANET pueden sufrir desconexiones debido al propio movimiento de los nodos a lo largo del tiempo, fallos o mal funcionamiento de estos. Así, al evaluar diferentes soluciones al problema en donde existen particiones, y además la conectividad ofrecida es igual para todas, entra en juego una tercera función: O_3. Esta minimiza la distancia existente entre cada RN y una serie de puntos imaginarios que se calculan y ubican justo en el punto medio entre cada par de particiones. Los autores denominan a estos puntos *puntos de atracción* o AP (*Attraction Point*). Esta minimización tiene como objetivo llevar a los RN hacia dichos puntos. O_3 se obtiene como sigue:

$$O_3 = \min_{i \in R, j \in A} \left\{ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right\} \tag{4.4}$$

donde i se corresponde con el i-ésimo nodo *relay* y j es el j-ésimo punto de atracción, representando R y A el conjunto de RN y AP, respectivamente.

En resumen, la solución que obtiene el mayor valor de conectividad O_1 ($\max\{O_1\}$) es la mejor solución. En el caso de que haya varias soluciones que consigan conectar toda la red, aquella con un mayor valor de O_2 ($\max\{O_2\}$) será la elegida. En el caso de presentarse soluciones correspondientes a redes desconectados que además poseen el mismo nivel de conectividad, se escogerá aquella con un menor valor de O_3 ($\min\{O_3\}$).

El proceso de optimización se repite a lo largo del tiempo, de tal manera que el objetivo es llevar a los RN a sus mejores posiciones de acuerdo a la situación actual de la red y los objetivos de conectividad y *throughput* perseguidos. Tal como se expone a través del trabajo realizado en [172] y [173], la solución DKS tiene aplicabilidad dentro del contexto de la seguridad en redes MANET como una metodología viable para contrarrestar actuaciones maliciosas en este tipo de redes.

4.2.2. Limitaciones de la solución DKS

Prácticamente todas las soluciones de posicionamiento vistas a lo largo de la Sección 4.1 que se aplican a entornos dinámicos, tratan de resolver dos cuestiones principales: (*i*) dónde posicionar cada RN y (*ii*) cómo llevarlos hasta dichas posiciones. DKS presenta serios problemas en la manera en que afronta la resolución de ambos aspectos. Estos inconvenientes son especialmente relevantes dentro de entornos dinámicos, donde es alta la probabilidad de aparición de desconexiones y, por lo tanto, particiones.

Como ya se describió durante la subsección anterior, DKS optimiza la posición de una serie de nodos *relay* mediante el uso conjunto de tres funciones de coste diferentes dentro del proceso de optimización. Es notable el hecho de que esta metodología dista bastante de ser la mejor. Lo habitual en cualquier problema de optimización es definir correctamente una única función objetivo directamente relacionada con el problema a solucionar. Una vez definida la función de coste, será un algoritmo determinado o *solver* el que, atendiendo a la maximización o minimización de dicha función, obtendrá una solución al problema. Sin embargo, el enfoque aportado por los autores de DKS lleva al sistema a potenciales comportamientos contradictorios. Además, un agravante adicional al comportamiento de la propuesta es la utilización de la función O_1 en sí y su carácter discreto. En optimización discreta el espacio de búsqueda se divide en regiones planas, en donde no existe información diferencial alguna que

permita facilitar su solución. En general, los problemas de optimización continua con funciones de coste suaves son más fáciles de resolver [174]. En estos casos, sí que existe información diferencial o derivativa en la función de coste, siendo posible conducir la optimización hacia la solución que maximiza o minimiza (dependiendo del problema) la función objetivo.

El dinamismo inherente de las redes MANET aumenta la probabilidad de aparición de particiones en la red. En estos casos es la función O_3 la que principalmente se encarga de dirigir el movimiento de los RN hacia los puntos de atracción previamente calculados. Consecuentemente, es necesaria una buena definición para O_3 ya que prácticamente es en ella donde recae casi todo el peso de la optimización. Sin embargo, O_3 no está bien definida en la solución DKS: solo consigue aprovechar un RN de los disponibles, dejando a los demás inservibles de cara a los objetivos del sistema. Con ánimo de ilustrar y explicar el funcionamiento erróneo de dicha función, se describen a continuación dos experimentos distintos. El primero de ellos se muestra en la Figura 4.1(a). En ella se observan 12 UN cuyos identificadores van desde el 1 al 12 (círculos azules) y que se distribuyen por todo el área. Adicionalmente, en la misma figura, se aprecian 3 RN con identificadores desde el 1 al 3 (cuadrados rojos). En este punto, es conveniente recordar que el objetivo principal de estos últimos es conectar el número máximo de nodos UN entre sí. En la Figura 4.1(a) se aprecia la existencia de tres particiones formadas por los conjuntos de nodos $\{1,2,3,4\}$, $\{5,6,7,8,9\}$ y $\{10,11,12\}$. Adicionalmente, se computan necesariamente 3 puntos de atracción (representados con triángulos rojos invertidos) justo en el punto medio entre cada par de particiones. Durante todo el experimento los UN permanecerán estáticos con el objetivo de simplificar el problema y contribuir a la fácil observación del erróneo funcionamiento de la función O_3.

La Figura 4.1(b) muestra la posición final que alcanzan los RN. Se hubiese esperado que cada uno de los RN implicados se localizasen sobre cada uno de los puntos de atracción, o al menos en una zona cercana a ellos. No obstante, se observa cómo uno solo, RN2, se posiciona correctamente, permaneciendo RN1 y RN3 prácticamente en la misma ubicación inicial.

Un segundo experimento trata de corroborar el comportamiento indeseado que se produce al utilizar O_3. El objetivo de esta prueba es obtener la superficie de optimalidad de dicha función. Para ello se propone el escenario estático representado en la Figura 4.2(a), compuesto por 10 UN, 2 RN y 2 puntos de atracción. RN1 y RN2 se mueven a lo largo de la línea diagonal que une la esquina inferior derecha del escenario con la superior izquierda. A partir de este escenario en 2D se computan los valores de la función O_3 para cada par de posiciones RN1-RN2 según su movimiento a lo largo de la diagonal. La superficie resultante se observa en la Figura 4.2(b). A través de la inspección de la citada figura, se puede concluir la mala definición de la función O_3. Una correcta definición de dicha función mostraría una superficie de optimalidad con solo dos puntos mínimos, correspondientes con los casos en

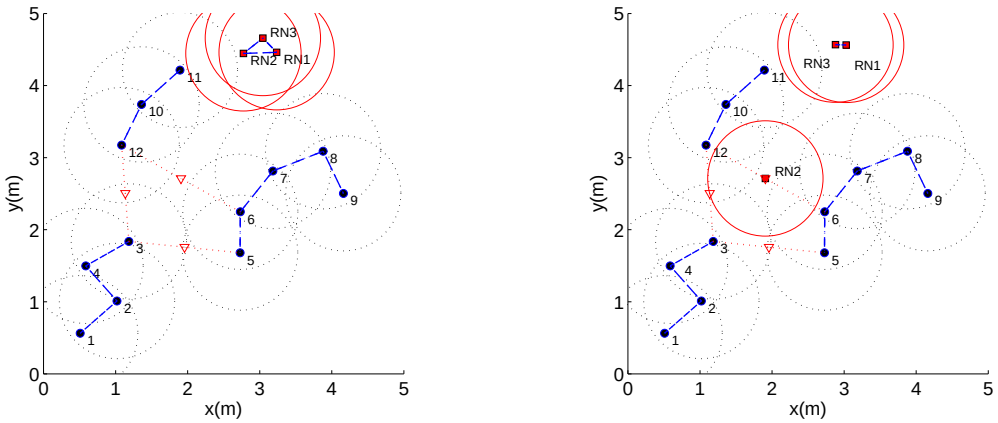
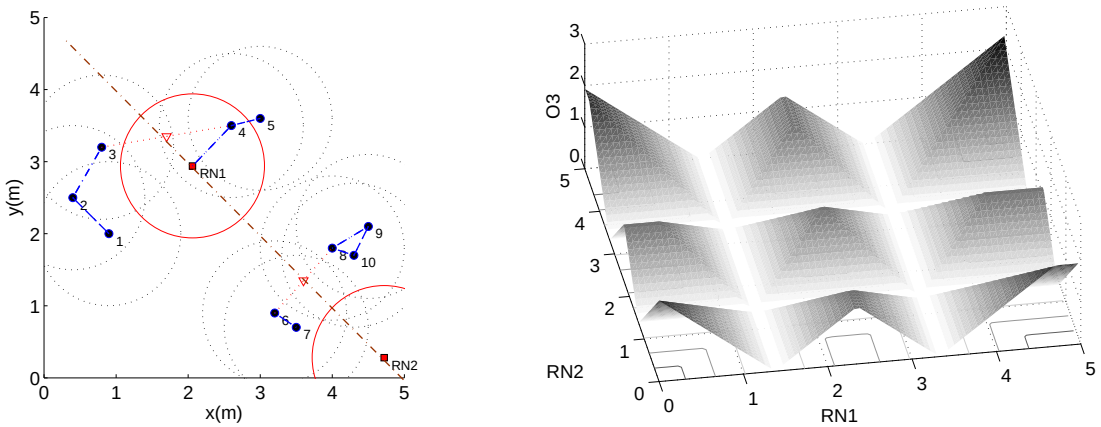


Figura 4.1: Primer experimento: posicionamiento de los RN a lo largo del tiempo. La subfigura (a) muestra las localizaciones de los 12 UN implicados (representados con círculos azules) en el experimento, así como la posición de partida de los 3 RN considerados (cuadrados rojos). En el ejemplo se establecen 3 particiones dando lugar a 3 puntos de atracción (triángulos rojos invertidos). A través de la sufigura (b) se observa la posición final que alcanza cada uno de los RN transcurridos varios instantes de simulación.

los que los RN se sitúan justo encima de cada uno de los puntos de atracción. Sin embargo, la forma mostrada en la Figura 4.2(b) presenta una superficie con valles que se corresponden con los puntos mínimos de la función. Estos valles se deben al posicionamiento de un RN sobre un punto de atracción.

Otro aspecto importante es la localización de los puntos de atracción. Se puede entrever que la mejor solución dista mucho de la que se propone en DKS: establecer dichos puntos justo en el punto medio entre cada par de particiones. Esta forma tan simple de posicionarlos hace que, cuando un *relay* alcance o se sitúe alrededor de uno de ellos, es probable que ni siquiera así pueda conectar las dos particiones. Esto ocurrirá siempre que la distancia que separa los nodos más cercanos entre dos particiones diferentes sea mayor que $2 \cdot c$, siendo c el radio de cobertura de los nodos. Teniendo en cuenta el gran dinamismo presente en este tipo de entornos, es probable que las distancias que separan las particiones sean, al menos gran parte de las veces, mayores que $2 \cdot c$, haciendo inútil el posicionamiento de un RN en dicho punto de atracción. En la Figura 4.1(b) se observa este comportamiento: RN2 es capaz de posicionarse justo en el punto de atracción requerido pero resulta inútil ya que no consigue conectar las dos particiones implicadas. Una posible solución a este problema implicaría una correcta definición de la ubicación de los AP que, junto con una buena especificación de la función O_3, llevaría al procedimiento a mover los RN hacia cada uno de los AP.



(a) (b)

Figura 4.2: Segundo experimento: obtención de la superficie de optimalidad de la función O_3. En la subfigura (a) se muestra el escenario ideado para la obtención de dicha superficie, donde RN1 y RN2 se mueven sobre la diagonal mostrada. Cuando se posicionen a la altura de los puntos de atracción mostrados (triángulos rojos invertidos) se conectarán las particiones asociadas. La subfigura (b) muestra la superficie de optimalidad obtenida para la función O_3 donde, sobre el eje Z, se representan los valores de dicha función para cada par de posiciones de ambos RN (representados como cuadrados rojos en la subfigura (a)).

En las siguientes secciones se formula, discute y prueba la alternativa aquí propuesta que solventa los defectos encontrados en DKS no solo en la definición y herramientas utilizadas en el problema, sino en la metodología que siguen sus autores para solventarlo.

4.3. Mejoras en la localización y control del movimiento de nodos *relay*

Dentro de un entorno dinámico como son las redes MANET, en donde la topología de la red cambia a lo largo del tiempo, la localización de nodos *relay* y el control de su movimiento se torna una tarea complicada. Evidentemente, el objetivo de rendimiento que se establece para el sistema condiciona totalmente el modo en que se aborda esta cuestión. En el caso que nos atañe, el objetivo final del sistema es maximizar tanto la conectividad como el *throughput* que consigue la red.

A continuación se presentan algunos aspectos y conceptos preliminares necesarios para la presentación y discusión de la formulación del problema.

4.3.1. Aspectos y conceptos preliminares

Al igual que en el esquema DKS, en nuestra aproximación se contemplan dos tipos de nodos. Estos son los UN y los RN. A modo de recordatorio, los primeros hacen uso de los servicios ofrecidos por la red que provienen de capas superiores (por ejemplo, dispositivos sensores que demandan ciertos parámetros de comunicación y calidad de servicio para llevar a cabo el envío de la información capturada en tiempo y forma), mientras que los segundos se encargan de hacer de retransmisores o *relay* de la información ayudando a que las comunicaciones se lleven a cabo. Cabe mencionar que, aunque estos nodos son móviles en el problema que se plantea aquí, la solución también se puede aplicar a esquemas en donde dicha capacidad de movimiento no es necesaria. Algunos ejemplos de nodos RN móviles son: UAV [175] o robots móviles [176], entre otros.

Para el problema de la optimización del posicionamiento de nodos se tendrán en cuenta las siguientes asunciones que, por otro lado, son ya implícitamente consideradas en el esquema DKS:

- *El proceso de optimización es centralizado.*

  Esto implica que existirá un nodo capaz de, a través de los procedimientos de comunicación y procesamiento necesarios: (*i*) capturar la información de red necesaria, (*ii*) ejecutar la rutina o algoritmo de optimización a partir de esta información y (*iii*) enviar información de control (es decir, las nuevas posiciones computadas o *target locations*) de acuerdo con los resultados de la optimización.
- *Las posiciones que adoptan los UN dentro de la red no se controlan.*

  Solo los RN serán posicionados durante la optimización. Aunque esta asunción introduce complejidad adicional, se generaliza así el problema en el que la localización de los UN es también optimizada.
- *El número de RN está limitado.*

  Esta es una asunción realista ya que, en la mayoría de las situaciones, solo se dispone de un número limitado de RN.

Adicionalmente, se consideran también las siguientes asunciones secundarias:

- *El proceso de optimización solo se limita a un espacio en dos dimensiones.*
- *Los UN son dispositivos móviles.*
- *La solución es single-tiered [147], ya que tanto los UN como los RN reenvían o retransmiten información proveniente de otros UN.*
- *El radio de cobertura para ambos tipos de nodos es de c metros.*

- *Tanto los UN como los RN tienen limitada su velocidad.*

Mientras que las primeras asunciones constituyen los cimientos de la propuesta, el último conjunto de estas se establece con el fin de limitar las posibilidades de la solución. De esta manera, se podría pensar en la natural evolución del problema a entornos en tres dimensiones, donde se considerasen redes *two-tiered* con diferentes radios de cobertura para ambos tipos de nodos.

En el contexto general que se ha descrito anteriormente, una red MANET que posee un determinado número de UN y RN se puede especificar como sigue:

$$G = (N, E) \tag{4.5}$$

con

$$N = U \cup R \tag{4.6}$$

donde U, R, N y E se corresponden con el conjunto de UN, el conjunto de RN, el conjunto completo de todos los nodos inalámbricos y el número de enlaces inalámbricos de la red (*edges* en el contexto de la teoría de grafos), respectivamente.

En una red inalámbrica, los *edges* o conjunto de enlaces inalámbricos satisfacen la siguiente expresión:

$$E := \left\{ \mathbf{e}_{ij} \mid \|\mathbf{e}_{ij}\| \leq c, \forall n_i, n_j \in N \right\} \tag{4.7}$$

donde c representa el rango o radio de cobertura de un enlace simple entre dos nodos adyacentes, siendo n_i y n_j el i-ésimo y j-ésimo nodo de la red.

La optimización de la localización de los RN se puede representar como sigue:

$$\mathbf{G}^* := \arg\max_{\mathbf{G}} \{ f(G) \mid G = (N, E) \; y \; U = U_0 \} \tag{4.8}$$

donde $f(G)$ es la función a maximizar y U_0 se corresponde con la posición actual de los UN. $f(G)$ debe estar definida de forma que tanto la conectividad como el *throughput* sean considerados bien directa o indirectamente.

Hemos de mencionar que las soluciones que se exponen en la presente sección se basan en la mejora de ciertos aspectos básicos en el posicionamiento de nodos *relay* y que no fueron correctamente diseñados en el esquema DKS. Por esta razón, para la descripción y discusión de la utilidad de la nueva propuesta se utiliza como base y a

modo de hilo conductor la solución DKS. Para contribuir al mejor entendimiento de la propuesta y adaptar conceptos y ecuaciones a la nueva formulación introducida en esta sección, se formulan y a veces repiten algunas de las fórmulas que fueron introducidas en la Sección 4.2.

4.3.2. Localización y control de movimiento optimizado para los nodos *relay*

Dado el contexto en el que se enmarca el sistema planteado, la topología de red cambia continuamente haciendo que sea necesario recalcular la trayectoria seguida por los RN prácticamente en cada instante de tiempo. De acuerdo a esos continuados cambios, es posible que los *relays* no alcancen la posición deseada antes de que se produzca el siguiente cambio. En el campo relacionado con el control automático se dice que en tal caso no se alcanzará un estado estable en el sistema. Además, los nodos verán limitado su movimiento por la definición de una velocidad máxima. En el caso de los RN se impide así que alcancen una ubicación objetivo (o AP) de manera instantánea, lo que por otro lado no sería realista desde el punto de vista físico.

Para abordar este reto, la propuesta DKS propone una solución inspirada en la metodología MPC donde se infiere el comportamiento de la red H instantes de tiempo más adelante, siendo H el horizonte de predicción. De esta manera, el sistema se anticipa a posibles cambios en la red MANET, produciendo trayectorias de movimientos más eficientes hacia una serie de puntos dados o AP.

Como ya se expuso en la Sección 4.2, DKS considera varias funciones de coste en el proceso de optimización empleado. Definamos R_{O1} como el conjunto de soluciones óptimas para las posiciones de los RN en términos de la función $O_1(G)$:

$$R_{O1} := \underset{R^{(t+H)}}{\arg\max} \left\{ O_1(G(U^{(t+H)} \cup R^t, E)) \right\} \tag{4.9}$$

donde $U^{(t+H)}$ contiene las posiciones inferidas de los UN en el horizonte de predicción H, R^t se refiere a la localización de cada RN en el instante actual, y siendo $O_1(G)$ la versión reformulada de (4.1) tal y como sigue:

$$O_1(G) = \frac{2}{|U| \times (|U| - 1)} \times \sum_{\forall u_i, u_j \in U, j > i} z(G, i, j) \tag{4.10}$$

donde $|U|$ es el número de UN considerado, correspondiéndose u_i y u_j con los i-ésimo y j-ésimo UN, respectivamente. $z(G, i, j) = 1$ si existe una camino que conecta u_i y u_j en G (bien directo o a través de encaminamiento *multi-hop*), siendo $z(G, i, j) = 0$

en otro caso. La función $z(.)$ se puede entender como una medida de si dos nodos se pueden comunicar dentro de la red, ya que dichos nodos pueden estar o no accesibles.

De acuerdo a la naturaleza de $O_1(G)$, es probable que la cardinalidad de R_{O1} sea mayor que 1 ya que, como ya se comprobó anteriormente, puede que exista más de una solución que obtenga el mismo valor de $O_1(G)$ dada su naturaleza discreta. Cuando se produce esta situación, DKS lleva a cabo el siguiente procedimiento de selección en donde hace uso de las funciones $O_2(G)$ u $O_3(R,A)$, para obtener las mejores localizaciones R^*_{DKS} de entre el conjunto obtenido en R_{O1}:

$$R^*_{DKS} := \begin{cases} \arg\max\limits_{R^t} \left\{ O_2(G(U^{(t+H)} \cup R^t, E)) \mid R^t \in R_{O1} \right\}, & \text{if } O_1(R_{O1}) = 1 \\ \arg\min\limits_{R^t} \left\{ O_3(R^t, A^{(t+H)}) \mid R^t \in R_{O1} \right\}, & \text{if } O_1(R_{O1}) < 1 \end{cases} \tag{4.11}$$

siendo $A^{(t+H)}$ la estimación de los AP en el horizonte H, y $O_2(G)$ como sigue:

$$O_2(G) = \min_{u_i, u_j \in U : j > i} \{ T(G,i,j) : T(G,i,j) > 0 \} \tag{4.12}$$

donde $T(G,i,j)$ se corresponde con el *throughput* existente entre el i-ésimo y j-ésimo UN en la red G, estando este limitado por el enlace que menor *throughput* admite. Como se expuso en la Sección 4.2, el *throughput* entre dos nodos adyacentes de la red se aproxima en función de la distancia que los separa (ver (4.3)).

$O_3(R,A)$ se define como sigue:

$$O_3(R,A) = \min_{r_i \in R, a_j \in A} \left\{ \sqrt{(r_i^x - a_j^x)^2 + (r_i^y - a_j^y)^2} \right\} \tag{4.13}$$

entendiéndose r_i como el i-ésimo RN con sus correspondientes coordenadas en el plano, r_i^x y r_i^y. Por otro lado, a_j es el j-ésimo AP y a_i^x y a_i^y sus coordenadas en el plano.

Es conveniente notar que, en este punto, los movimientos de los nodos en cada instante de tiempo se restringen a $d(r_i^t, r_i^{t+H}) \leq H \cdot l$, donde r_i es la posición (bien en el instante t o en el $t+H$) del i-ésimo RN, y $d(.,.)$ la distancia Euclídea entre dos nodos:

$$d(n_i, n_j) = \sqrt{(n_i^x - n_j^x)^2 + (n_i^y - n_j^y)^2} \tag{4.14}$$

notándose como n_i^x y n_i^y a las coordenadas (X,Y) del nodo n_i. En otras palabras, DKS limita la distancia que un RN puede recorrer en un intervalo de tiempo simple a l

unidades. l se puede derivar de la velocidad de un nodo si se define esta como l/ts, siendo ts un intervalo de tiempo determinado.

Con el objetivo de solventar la problemática que introduce DKS asociada al uso de varias funciones de coste, así como la evitación en el empleo de $O_1(G)$ dentro del proceso de optimización, alternativamente se propone contemplar dos procedimientos de optimización diferenciados. Cada uno de ellos se ha adaptado en función de la casuística asociada al problema, y en donde se empleará una única función de coste. Estos casos, que también se consideran en la solución DKS pero de manera conjunta, son: (*i*) la red esta totalmente conectada y (*ii*) la red presenta desconexiones. Una manera de discernir entre ambas situaciones es medir la conectividad que presenta la red. Para ello se utilizará la función provista por DKS en (4.10), salvo por el hecho de que G no considera ahora los RN para el cálculo de la conectividad. Al grafo correspondiente a la red que solo contempla los nodos de usuario lo denominaremos G^U.

Optimización para redes conectadas

Considerando aquellas situaciones en las que $O_1(G^U) = 1$, es decir, el estado actual de la red presenta conectividad entre todos sus nodos, se trata de incidir en la maximización del *throughput* alcanzado. Para ello se definirá una nueva y única función que, teniendo en cuenta la premisa anterior, sustituya también la evaluación de $O_1(G)$ (4.10) y $O_2(G)$ (4.12) tal y como se realizaba en DKS en esta situación.

Consecuentemente con la motivación anterior, se expone y propone la utilización de una función alternativa que considere en su definición el *throughput* de la red y sea continua y suave para evitar así los problemas asociados a la optimización de funciones discretas. Definimos de este modo la función de coste $g(G')$ tal y como sigue:

$$g(G') = \sum_{\forall u_i, u_j \in U : j > i} id(G'_i, i, j) \tag{4.15}$$

donde G'_i se corresponde con el grafo obtenido al computar el *spaning tree* que parte desde el i-ésimo UN y tal que minimiza la mayor distancia encontrada desde el nodo de partida hacia los restantes. La función $id(G'_i, i, j)$ es la inversa de la distancia del vértice más largo encontrado en G'_i desde el nodo i hacia el j. La función $g(G')$, a su vez, puede ser vista como una estimación aproximada del *throughput* global de la red, en donde el *throughput* correspondiente entre dos nodos adyacentes se aproxima por la inversa de la distancia que los separa, es decir, $w_{ij} = \frac{1}{d_{ij}}$.

La definición de $g(G')$ implica la redefinición de la red desde el punto de vista de un grafo completo G', tal que:

$$G' = (N, E') \tag{4.16}$$

$$E' := \left\{ \mathbf{e'}_{ij} \mid \forall n_i, n_j \in N \right\} \tag{4.17}$$

En comparación con la función $O_1(G)$, $g(G')$ se puede ver como una versión suavizada de esta. Para ilustrar este hecho gráficamente, en la Figura 4.3 se muestran los valores que obtienen ambas funciones dentro de un escenario ilustrativo. En el escenario ideado se colocan dos UN separados una distancia de 3 unidades y se mueve un RN progresivamente sobre la línea recta que los separa. El rango máximo o radio de cobertura de cada nodo es 2 unidades, es decir, $c = 2$. El eje de las abscisas se divide en unidades sobre las que se moverá el RN, colocándose los UN en los puntos (1,2.5) y (4,2.5) inicialmente.

Los tres nodos se conectan cuando el RN se ubica dentro del intervalo [2,3]. Si se observa la Figura 4.3 en dicho intervalo, $O_1(G)$ proporciona los mismos valores independientemente de la localización del RN dentro del anterior rango. Además, los intervalos [1,2] y [3,4] también ofrecen una región plana en términos de $O_1(G)$. En su lugar, la función $g(G')$ es continua en todo el intervalo [1,4], al tiempo que proporciona una forma suavizada en todos los puntos excepto en el óptimo, el punto 2,5. Claramente, esta función ofrece información diferencial para llevar a la optimización a su punto óptimo sin importar el punto inicial elegido.

Optimización para redes que presentan desconexiones

Considerando aquellos casos en los que $O_1(G^U) < 1$, que se esperan sean la mayoría de las situaciones cuando el contexto de uso se centra en entornos MANET, estaremos ante una red desconectada. Consecuentemente, el objetivo perseguido ahora es mover los RN hacia cada uno de los puntos de atracción para mejorar o restaurar la conectividad perdida. En el caso de DKS, los movimientos de los RN son principalmente dirigidos por la función $O_3(R, A)$ que minimiza la mínima distancia entre todos los RN y AP. De esta manera, únicamente un RN se desplazará hacia el correspondiente y más cercano AP. Este comportamiento hace que los demás RN resulten inútiles (ver Sección 4.2.2). Para solucionar este problema, se propone sustituir la dupla de funciones $O_1(G)$ y $O_3(R, A)$ evaluada dentro del proceso de optimización para este caso concreto, por una única función objetivo a la que llamaremos $p(R, A^*)$ y que presenta la expresión:

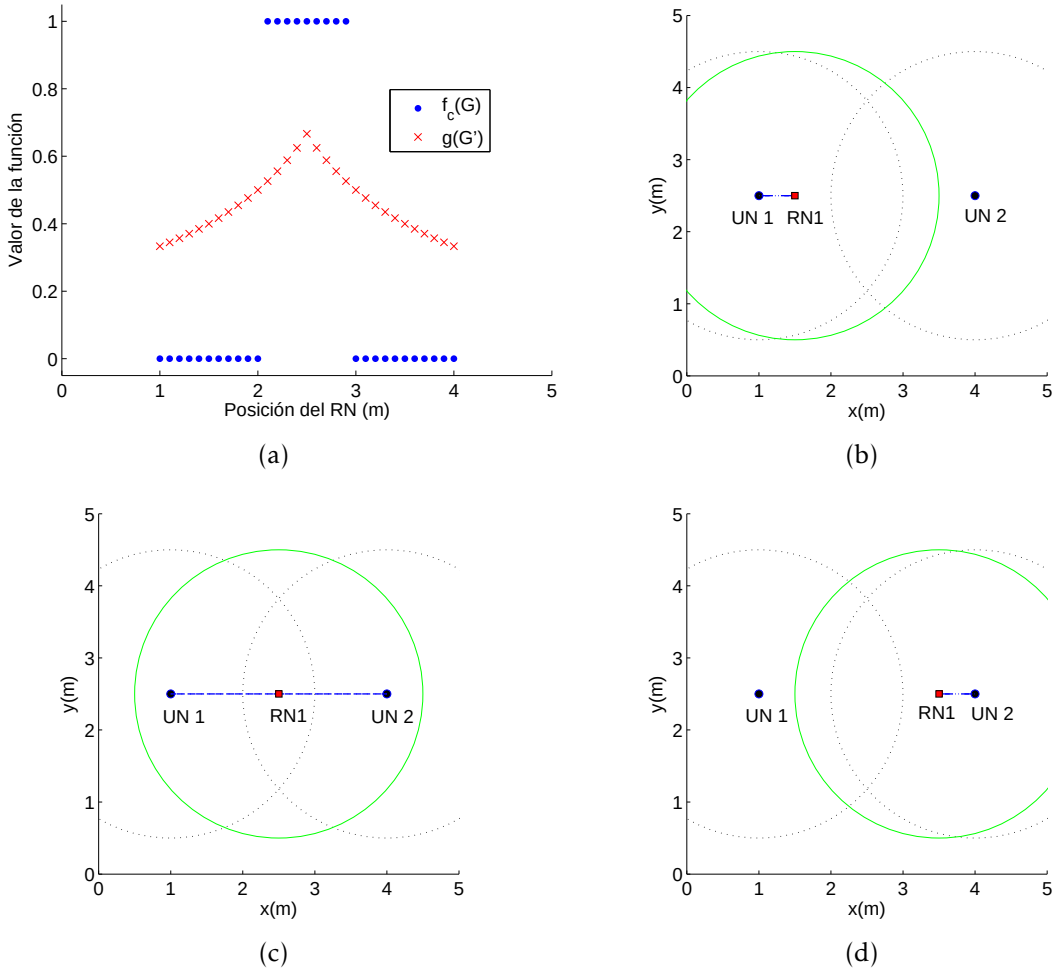


Figura 4.3: Comparativa de resultados entre las funciones $O_1(G)$ y $g(G')$. En la subfigura (a) se muestra la evolución de ambas funciones cuando el único RN presente se mueve desde UN1 a UN2. Las subfiguras (b), (c) y (d) muestran diferentes y relevantes instantes en el movimiento del RN: la red está desconectada, se conecta y se vuelve a desconectar de nuevo, respectivamente.

$$p(R,A^*) = \sum_{i=1}^{R}\sum_{j=1}^{A^*} d(r_i,a_j) - \sum_{i,j\in R:j>i} d(r_i,r_j) \tag{4.18}$$

La alternativa $p(R,A^*)$ mejora, por definición, a la función $O_3(R,A)$. Se observa cómo $p(R,A^*)$ substrae la suma de las distancias entre cada par de RN de la suma de las distancias entre RN y AP. Dicho de otra forma, $p(R,A^*)$ intenta mover los RN hacia los AP mientras mantiene a los primeros separados entre sí para evitar que se concentren en una determinada área. En [177] se define y justifica el uso de la función $p(R,A^*)$ frente a $O_3(R,A)$. Al contrario de $O_3(R,A)$, esta nueva definición

tiene en cuenta todos y cada uno de los RN en la optimización de la solución y utiliza un nuevo conjunto optimizado de puntos de atracción. Tal y como se vio en la Sección 4.2.2, DKS localiza los puntos de atracción de manera simplista y poco eficaz situándolos justo en el medio de cada una de las particiones presentes en la red. Por este motivo y como se expondrá en la siguiente sección, la nueva propuesta hará uso de un conjunto de AP, A^*, cuya ubicación se optimiza haciendo que $p(R, A^*)$ aumente su eficiencia, y por ende el rendimiento del sistema, en términos de conectividad y *throughput*.

Esquema completo

Con ánimo de aclarar conceptos y a modo de resumen, la alternativa propuesta en el presente capítulo se puede sintetizar mediante la siguiente expresión:

$$R^*_{new} := \begin{cases} \arg\max\limits_{R^t} \left\{ g(G'(U^{(t+H)} \cup R^t, E)) \right\}, & \text{si } O_1(G^U) = 1 \\ \arg\min\limits_{R^t} \left\{ p(R^t, A^{*(t+H)}) \right\}, & \text{si } O_1(G^U) < 1 \end{cases} \tag{4.19}$$

donde $g(G')$ descrita en (4.15) se utiliza como alternativa a la evaluación $O_1(G)$ y $O_2(G)$ para redes conectadas, y $p(R, A^*)$ como alternativa a la evaluación de $O_1(G)$ y $O_3(R, A)$ para redes desconectadas.

4.4. Sistema DRNS (*Dynamical Relay Node placement Solution*) y su aplicación en redes MANET

En la anterior sección se introdujo la metodología y formulación teórica para la solución propuesta en este capítulo. Principalmente, se disgrega la solución DKS para habilitar procesos de optimización separados con funciones de coste únicas, continuas y suaves, de acuerdo a los dos posibles estados de la red: totalmente conectada o desconectada. Ambas ramas o flujos de ejecución, se podría decir, pretenden mover los RN de manera optimizada y controlada hacia una localización objetivo optimizada a través del empleo de algoritmos basados en PSO y la correspondiente función de coste.

La estructura funcional de la solución de posicionamiento de RN completa se muestra en la Figura 4.4. En esta se observan los diferentes flujos de ejecución en función del estado de la red, evaluado este a través de la función $O_1(G^U)$. Adicionalmente al módulo selector que condiciona la ejecución del proceso de optimización, se presentan tres módulos adicionales. Considerando redes desconectadas ($O_1(G^U) < 1$,

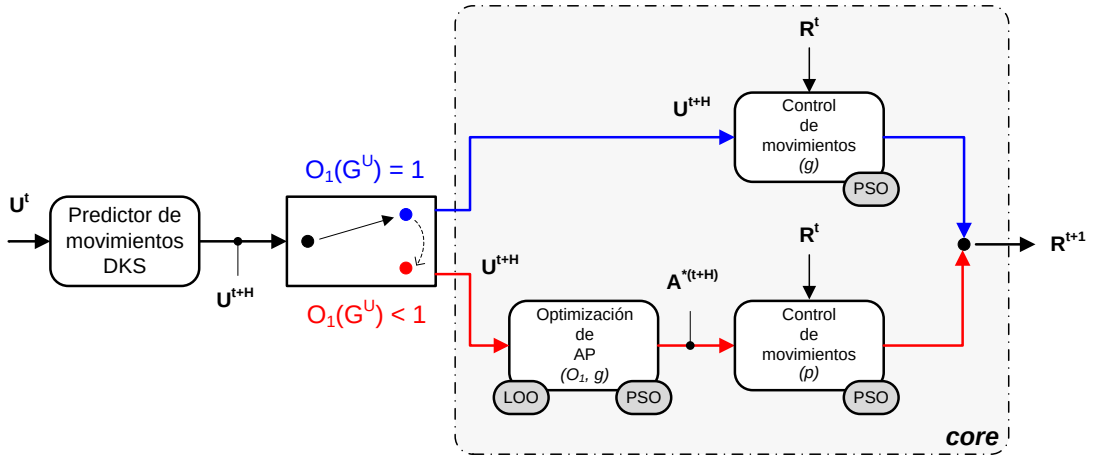


Figura 4.4: Bloques funcionales del sistema DRNS.

en rojo en Figura 4.4) es necesario computar primero los puntos de referencia hacia los que dirigir los RN para después moverlos hacia dichas localizaciones. Así, primero nos encontramos con el módulo de localización optimizada de AP que se encarga de obtener una ubicación adecuada para ellos en concordancia con los objetivos perseguidos por el sistema. Después, una optimización inspirada en MPC lleva los RN hacia el conjunto anterior de AP. Indistintamente en presencia de redes desconectadas o conectadas, este último módulo se basa en la misma filosofía de funcionamiento a excepción de la función objetivo que utiliza. Si la red presenta desconexiones, se abogará por la maximización de la conectividad para mejorarla o recuperarla. Por otro lado, para redes conectadas ($O_1(G^U) = 1$, en azul en la Figura 4.4) no tiene sentido promover la mejora de la conectividad, por lo que se considera el *throughput* como parámetro a maximizar.

Al margen del carácter conectado o no de la red, se implementa un módulo para la predicción de posiciones futuras de los UN. Este módulo promueve la adaptación eficiente de la trayectoria seguida por los *relays* durante sus movimientos, y sobre todo es útil en escenarios que varían con el tiempo. Este último módulo se hereda de la propuesta DKS [131].

A lo largo de la secciones próximas se describe en detalle cada uno de los módulos que forman el sistema.

4.4.1. Módulo para la localización optimizada de puntos de atracción

Aunque este módulo forma parte del esquema general ubicado dentro de un contexto dinámico, proporciona en sí mismo una solución al problema de posicio-

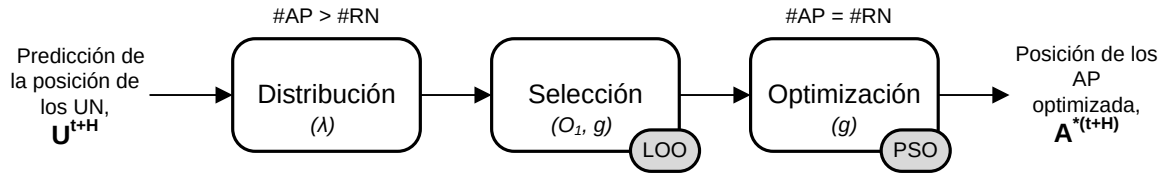


Figura 4.5: Subetapas del módulo de optimización en la localización de AP.

namiento de RN para entornos que no varían con el tiempo. En este contexto, las posiciones de los AP serían ahora la de los RN. A modo de ejemplo, este módulo se podría aplicar perfectamente para ubicar nodos *relay* WiFi con el objetivo de ampliar la cobertura total de la red. En la Sección 4.5.2 se obtienen resultados experimentales que validan el empleo de este módulo en escenarios en los que la red permanece estática a lo largo del tiempo.

A continuación, y teniendo como base el esquema general del módulo mostrado en la Figura 4.5, se describe cada uno de sus componentes: (*i*) distribución inicial de AP, (*ii*) selección de AP y (*iii*) optimización de AP.

Distribución inicial

Esta subetapa tiene como objetivo establecer un conjunto válido de AP de partida. Para ello, primero se computa el MST (*Minimum Spanning Tree*) entre las particiones existentes de la red en base a la distancia entre ellas. A continuación se distribuyen una serie de AP de manera homogénea sobre los enlaces o aristas obtenidas del cómputo del MST. Para determinar el número de AP que se distribuyen, se define el parámetro $\lambda \in (0,1]$ tal que:

$$l_{ij} = \left\lceil \frac{\|\mathbf{e}_{ij}\|}{\lambda \cdot c} \right\rceil, \ \forall u_i, u_j \in U \ s.t. \ \|\mathbf{e}_{ij}\| > c \tag{4.20}$$

donde l_{ij} representa el número de AP distribuidos sobre la arista $\mathbf{e}_{ij}$, c es el radio de cobertura y $\lceil \ \rceil$ denota la operación techo o *ceil* de acuerdo a su traducción en inglés. Por ejemplo, $\lambda = 1$ implica la utilización del mínimo número de AP de manera que se consiga conectividad total en el caso de que se ubicase un RN sobre cada AP. A medida que λ disminuye el número de AP aumenta.

Cada AP se distribuye de manera equidistante sobre un enlace en concreto $\mathbf{e}_{ij}$. Así, la distancia entre dos puntos de atracción a_z y a_q se puede obtener efectuando la operación $d_{a_z a_q} = \|\mathbf{e}_{ij}\|/l_{ij}$. Además se cumple que:

$$\lambda \cdot c \geq d_{a_z a_q}, \ \forall u_i, u_j \in U \ s.t. \ \|\mathbf{e}_{ij}\| > c \tag{4.21}$$

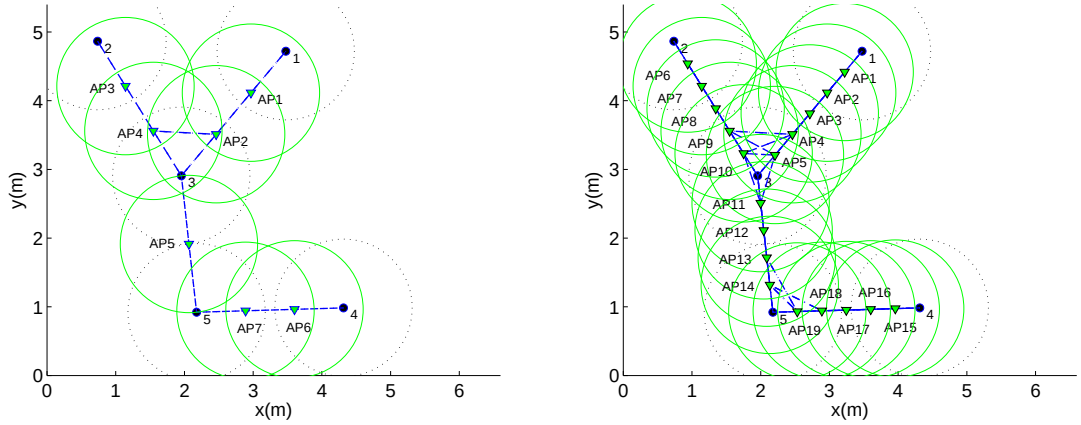


Figura 4.6: Efecto del parámetro λ en el número inicial de AP candidatos distribuidos a lo largo del MST: (a) $\lambda = 1 \rightarrow 7$ AP y (b) $\lambda = 0{,}4 \rightarrow 19$ AP.

De esta manera el número total de posiciones iniciales para los AP o número de AP candidatos iniciales será:

$$\sum_{\forall u_i, u_j \in U} l_{ij},\ s.t.\ \|\mathbf{e}_{ij}\| > c \tag{4.22}$$

A través de la Figura 4.6 se observa el efecto del parámetro λ sobre el número de AP candidatos (representados como triángulos invertidos de color verde) en un escenario de ejemplo en el que se distribuyen 5 UN (mostrados como círculos azules). Una selección adecuada de λ tendría en cuenta que valores pequeños podrían ralentizar el sistema desde el punto de vista computacional. Por el contrario, valores elevados podrían tener un impacto relevante sobre el rendimiento del sistema debido a la reducción del número de combinaciones alternativas sobre las que seleccionar el conjunto de AP adecuado en la siguiente etapa de selección. Como ya se verá durante la experimentación (Secciones 4.5.2 y 4.6.2), un valor adecuado para λ sería aquel que consiguiese que el número de AP iniciales fuese varias veces mayor que el de RN. Esta conclusión se obtiene a través de una serie de experimentos empíricos demostrando que ese rango de valores ofrece un buen balance entre el tiempo de computación y el rendimiento ofrecido por el sistema.

En términos de la complejidad computacional asociada a la ejecución de la distribución de AP, dicho coste se relaciona directamente con el número de UN considerados y el valor seleccionado para λ. Sin embargo, ya que el valor de λ se fija previamente, el tiempo de computación se ve principalmente influenciado por el algoritmo elegido para el cálculo del MST. En estos términos, y después de analizar cada una de las rutinas o algoritmos implicados, se obtiene un límite superior que,

en notación Big $\mathcal{O}$, se corresponde con $\mathcal{O}(n^2 \cdot \log n)$. Dicha expresión se obtiene de la complejidad asociada al algoritmo de Prim [178] para el cálculo del MST con n el número de UN considerados en el problema.

Selección de la solución

Una vez realizada la distribución inicial de AP candidatos, se procede a su selección teniendo en cuenta los objetivos de rendimiento del sistema. Este es un aspecto que tampoco se contempla en DKS, siendo el número de AP un condicionante para el tiempo de ejecución y rendimiento del sistema, sobre todo ante un número elevado de UN y cuando se consideran áreas de despliegue grandes con nodos dispersos. En el trabajo [177] se expone esta problemática, así como la necesidad del adecuado proceso de selección de AP.

Una vez motivada la necesidad del subsecuente proceso de selección, el procedimiento propuesto aquí se basa en la eliminación iterativa de cada AP para luego evaluar qué efecto tiene dicha eliminación sobre la conectividad y *throughput* del sistema. Para evaluar este impacto y decidir si un determinado AP se descarta o no, se implementa un procedimiento basado en LOO (*Leave-One-Out*) [179]. En cada iteración, se determina el conjunto de peores puntos de atracción, A_*, como sigue:

$$A_* := \arg\max_{\mathbf{a}_j \in A} O_1(G(U \cup \{A - a_j\}, E)) \tag{4.23}$$

Si A_* contiene un único AP en a_*, este será el que se descartará. En otro caso, a_* se obtiene de la siguiente manera:

$$a_* := \arg\max_{\mathbf{a}_j \in A_*} g(G'(U \cup \{A - a_j\}, E)) \tag{4.24}$$

Después, A se actualiza siguiendo $A = \{A - a_*\}$, repitiéndose esta operación hasta que el número de AP dentro del conjunto A se corresponda con el número de RN disponible.

Esta selección, aunque puede parecer simple, no es una tarea trivial. Su complejidad se puede aproximar por $\mathcal{O}((k^2 - m^2) \cdot n^4)$, donde k se corresponde con el número de AP correspondientes a la distribución inicial, y m es el número de RN disponibles en el sistema. Claramente es el término n^4 el que más influye en la ecuación. Este tiene su origen en la complejidad inherente y asociada al cómputo de $g(G')$ a partir de la ecuación (4.15).

Algoritmo 1: Pseudo-código para la optimización de AP basado en PSO.

Datos: Localizaciones de $U^{(t+H)}$ y $A^{(t+H)}$.
Resultado: Posiciones óptimas de los AP, $A^{*(t+H)}$ ($gbest$)

$[X, V] \leftarrow initialPopulation(\#A^{(t+H)}, \#particles)$;
$[gbest, pbest] \leftarrow evaluatePopulation(X, A^{(t+H)}, U^{t+H})$;
mientras $k \leq maxIterations$ **hacer**
  $w = w * 0{,}98$;
  para cada $i = 1 \cdots \#particles$ **hacer**
    $V(i) \leftarrow getVelocityIncrements(X, V, w, pbest(i), gbest)$; /* (B.8) */
    $V(i) \leftarrow setVelocityClamp(V(i), V_{max})$; /* (B.9) */
    $X(i) \leftarrow X(i) + V(i)$; /* (B.7) */
    $[X(i), V(i), gbest, pbest(i)] \leftarrow evaluateSolution(X(i), V(i), gbest, pbest(i), U^{t+H}, A^{(t+H)})$;
  fin
  $k = k + 1$;
fin
$A^{*(t+H)} \leftarrow gbest$;

Optimización de la solución

Una vez que los AP que ofrecen mejores resultados en función de $O_1(G)$ y $g(G')$ han sido seleccionados, la subetapa de optimización considera soluciones alternativas y que no se tuvieron en cuenta anteriormente con el fin de mejorar la localización obtenida. Como ya se verá en las Secciones 4.5.2 y 4.6.2, este paso introduce notables incrementos en el *throughput* ofrecido por la red, mientras mantiene e incluso aumenta la conectividad ofrecida por aquella.

Para llevar a cabo esta mejora se implementa un algoritmo de optimización basado en PSO. El pseudo-código del procedimiento se presenta en el Algoritmo 1. En primer lugar, y tomando como punto de partida las posiciones del conjunto de AP obtenido de la etapa de selección, se distribuye aleatoriamente un conjunto de partículas X sobre el espacio de búsqueda, estableciendo los correspondientes incrementos de velocidad inicial V también de manera aleatoria. A partir del conjunto inicial de partículas, se evalúa cada una de ellas para obtener la mejor solución global así como la mejor solución de cada partícula, *gbest* y *pbest*, respectivamente. Nótese que al inicio la mejor solución individual conseguida por cada partícula se corresponde, obviamente, con la solución aleatoria inicial. Acto seguido se actualiza la posición de cada partícula y de nuevo se evalúa. Este procedimiento se repite iterativamente un número definido de veces (*maxIterations*). Una vez que finaliza el procedimiento, se devuelve la mejor solución/posición para cada AP, A^*.

Los principales factores que afectan a la complejidad computacional del Algoritmo 1 son tanto el número de partículas consideradas como el número de iteraciones que se ejecuta el núcleo PSO para obtener la solución. Gran parte de esta complejidad tiene su origen en la ejecución de las funciones encargadas de evaluar cada

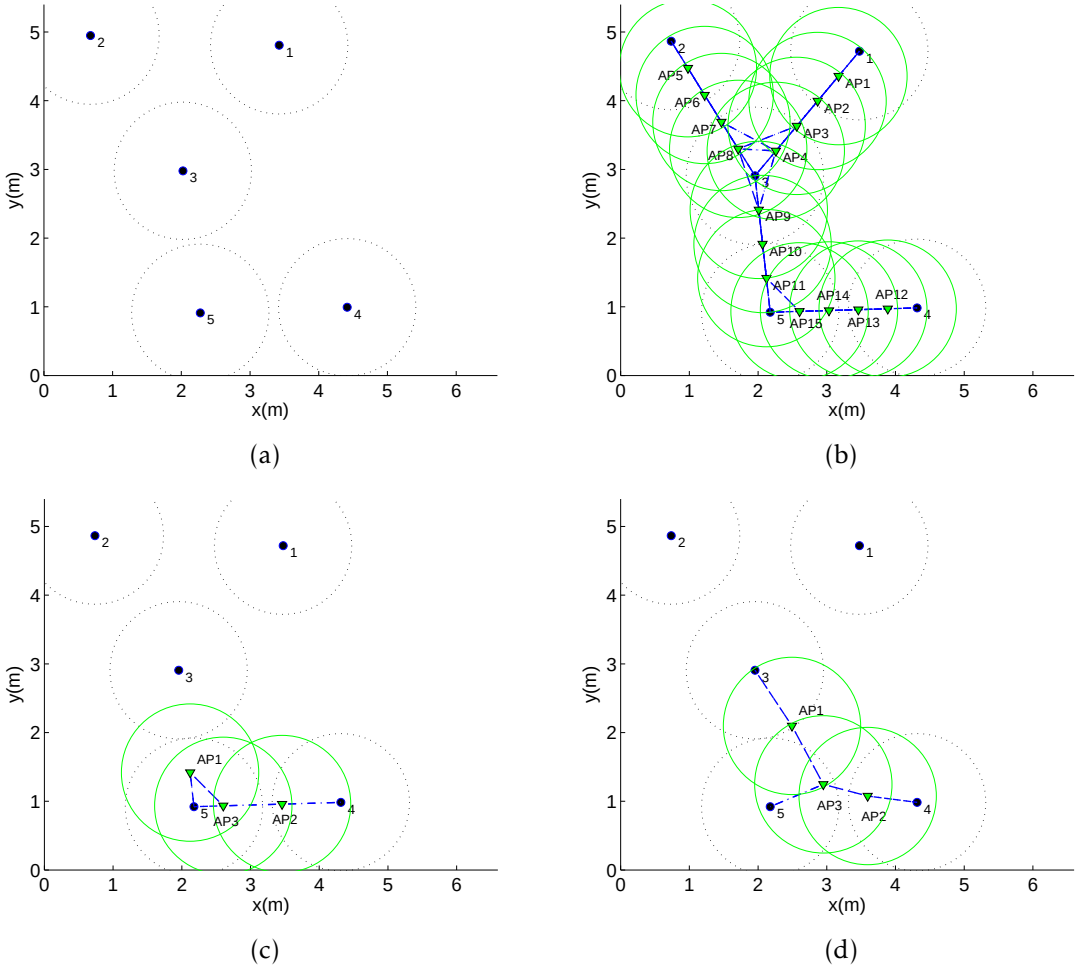


Figura 4.7: Posiciones de los AP (triángulos verdes invertidos) obtenidas tras cada una de las etapas del módulo de localización de AP: (a) distribución inicial de UN, (b) distribución inicial de AP a lo largo del MST entre particiones, (c) selección de AP y (d) localización de AP tras la etapa de optimización basada en PSO. Se consideran 5 UN (círculos azules) y 3 RN disponibles.

solución candidata. De esta manera, y tras analizar el algoritmo, se concluye que su complejidad se aproxima por $\mathcal{O}((n^4+m)\cdot p\cdot i)$ donde, como ya se intuyó, se observa la influencia del número de partículas p e iteraciones i del algoritmo. Una vez más observamos un término influyente en la expresión que proviene de la evaluación de $g(G')$ (4.15) como función de coste para la selección de la mejor solución.

La Figura 4.7 muestra el procedimiento de optimización para cada uno de los pasos o etapas establecidas en el proceso de localización de AP. Para ello se idea un escenario a modo de ejemplo, en el que se consideran 5 UN (círculos coloreados en azul) y 3 RN disponibles. En la Figura 4.7(a) se ilustra la distribución inicial de los UN. En la Figura 4.7(b) se muestra la distribución inicial de los AP sobre las aristas

obtenidas del MST previamente computado y que unen las diferentes particiones de la red. Seguidamente, en la Figura 4.7(c) se puede observar la selección de AP llevada a cabo por el procedimiento LOO de acuerdo a los requisitos principales del sistema: maximizar conectividad y *throughput*. Finalmente, el resultado de la última etapa de optimización se observa en la Figura 4.7(d). Es notable ver cómo esta última redistribuye las posiciones de los AP para lograr conectar un UN adicional. Si observamos detenidamente la figura, se puede comprobar cómo esta nueva distribución hace que aumente el *throughput* disponible en la red, no solo porque conecta otro nodo más, sino porque la distribución de los AP hace que el mínimo flujo entre todos los enlaces sea máximo. En este ejemplo específico sería necesaria la adición de más AP para conseguir conectar toda la red. Sin embargo y, a pesar de la escasa cantidad de RN y por tanto de AP, el sistema es capaz de ubicarlos en aquellas posiciones que maximizan el rendimiento del sistema.

4.4.2. Módulo para el control optimizado del movimiento de los nodos *relay*

Independientemente del caso, ya sea para mover los RN hacia los puntos de atracción en redes desconectadas o hacia aquellas posiciones que maximizan el flujo de la red ante la presencia de una red totalmente conectada, dichos movimientos han de efectuarse de manera adecuada y controlada. Para ello se implementa un módulo encargado de controlar su movimiento. Este módulo está de nuevo basado en el algoritmo PSO, tal y como se detalla en la Figura 4.8. Nótese que el algoritmo PSO utilizado para este módulo y su homólogo en el caso de la optimización de AP, tienen claras funciones diferentes. Por esta razón, los meta-parámetros asociados a cada uno de ellos se configuran adecuadamente acorde a sus objetivos. Así, en la optimización de AP, el algoritmo PSO se configura de manera que su exploración sea mayor que para el caso del módulo de control. El comportamiento de estos algoritmos se puede controlar ajustando parámetros como los *coeficientes cognitivo y social* de PSO [135, 180]. De igual manera se puede actuar sobre el *coeficiente de inercia* asociado [181, 182]. En particular, el coeficiente de inercia configurado para la optimización de AP es mayor que el utilizado en el módulo controlador de movimientos. Esta diferencia se deriva de la necesidad de ampliar el espacio de búsqueda con el objetivo de encontrar nuevas soluciones que no fueron tenidas en cuenta en la etapa de selección de AP. Junto con el uso de coeficientes de inercia elevados, ser más laxos en cuanto a la restricción máxima en los incrementos de velocidad de las partículas hace que la capacidad de exploración del algoritmo aumente, estableciéndose esta mayor para la etapa de optimización de AP.

Para conducir los RN hacia las posiciones objetivo, el algoritmo PSO usará diferentes funciones de coste: $p(R, A^*)$ (4.18) cuando existan desconexiones en la red y $g(G')$ (4.15) cuando todos los nodos estén conectados. Las soluciones obtenidas se

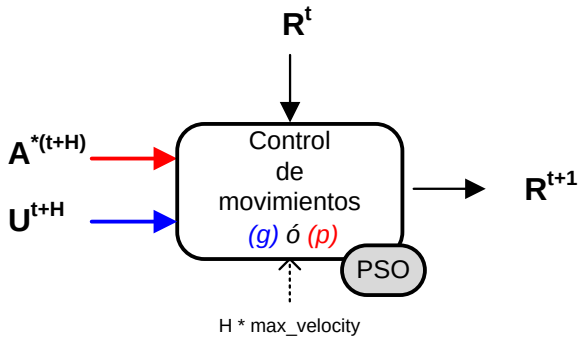


Figura 4.8: Detalle del módulo controlador de movimientos.

restringen a $H \cdot max\_velocity$, donde $max\_velocity$ se corresponde con la velocidad máxima que puede alcanzar un RN. Se puede decir que el procedimiento es capaz de obtener las posiciones optimizadas de los RN H instantes de tiempo en adelante, R^{t+H}. Siguiendo con la filosofía de trabajo MPC, aunque se obtiene una solución óptima a H pasos en un futuro, la solución final se acota a un solo intervalo de tiempo (en el caso simulado, se corresponde con un instante o tiempo de simulación). Durante este intervalo temporal un RN no podrá recorrer una distancia mayor que la que establece, obviamente, la velocidad máxima.

Aunque las posiciones de los RN se obtienen paso a paso (o, lo que es lo mismo, en cada iteración del algoritmo completo), $R^t \rightarrow R^{t+1}$, el hecho de trabajar con la predicción de la evolución de la red H instantes temporales en adelante, $(t+H)$, permite una mejor adaptación del sistema a los constantes cambios que se producen en la topología de red. Dicho de otra manera, la trayectoria que sigue cada uno de los RN es más eficiente si consideramos posiciones futuras de los UN que si se usa únicamente su posición en el instante de tiempo actual. Como ya se verá en las Secciones 4.5.2 y 4.6.2 a través de la oportuna experimentación, se observará que conforme pasa el tiempo la solución llega a un estado de estabilidad, indicando este hecho la capacidad de adaptación que proporciona el sistema.

Desde el punto de vista computacional, su complejidad está directamente influenciada por la función objetivo utilizada. Cuando se usa este módulo para contribuir a la recuperación de la conectividad su complejidad se aproxima por $\mathcal{O}(m^2 \cdot p \cdot i)$, en donde se observa la influencia de la función $p(R, A^*)$ a través del término m^2. Por otro lado, cuando se presentan redes conectadas, su complejidad viene claramente marcada por el coste computacional que añade la evaluación de la función $g(G')$. La expresión que aproxima la complejidad para este caso es $\mathcal{O}((n^4 + m) \cdot p \cdot i)$, en donde se aprecia el impacto de $g(G')$ a través del término n^4. Aunque es cierto que el mayor coste, computacionalmente hablando, lo introduce la ejecución de la función $g(G')$, también lo es el hecho de que la propia naturaleza dinámica de la red y los escenarios elegidos hacen poco probable que la red esté totalmente conectada. De esta manera,

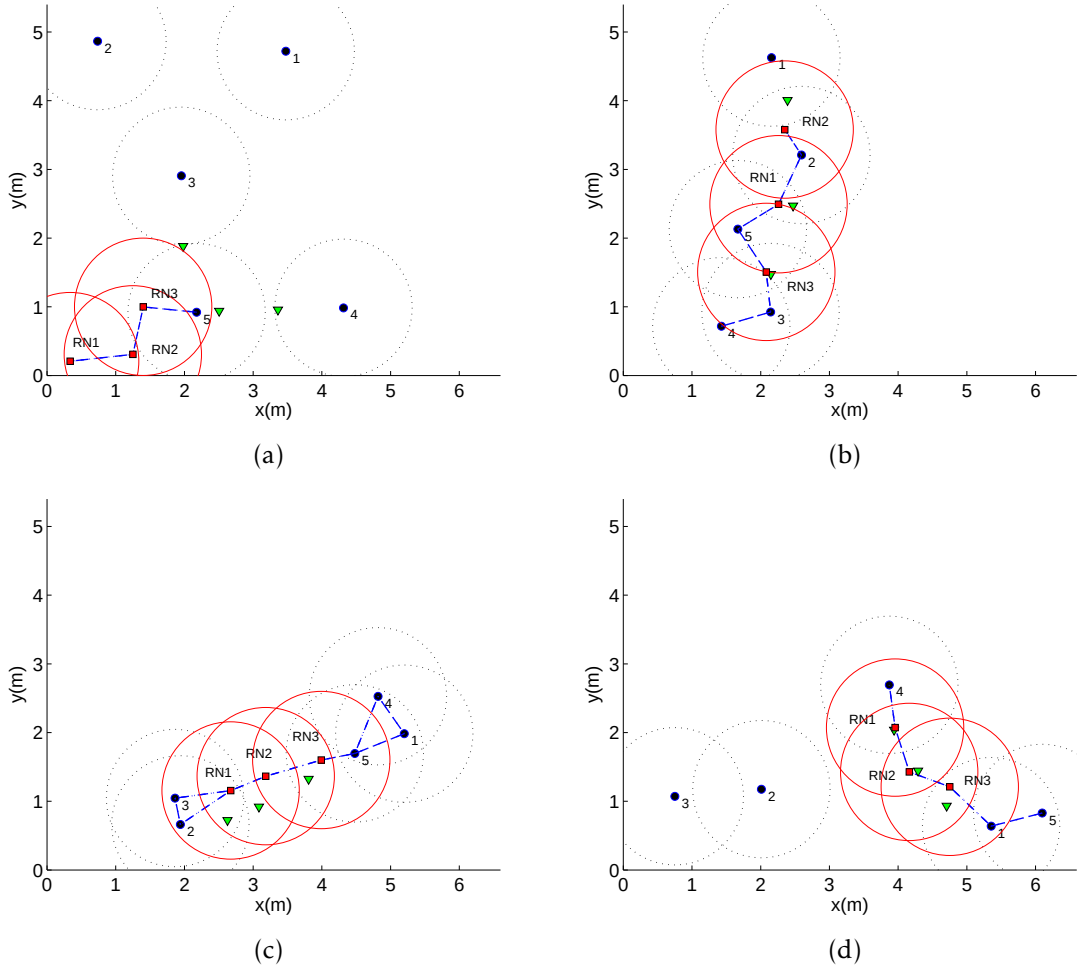


Figura 4.9: Ilustración de los movimientos de los RN (cuadrados rojos) a lo largo del tiempo cuando son dirigidos hacia a los AP (triángulos verdes invertidos). Las subfiguras (a) y (d) muestran las posiciones iniciales y finales de los nodos, respectivamente. Por otro lado, las subfiguras (b) y (c) presentan sendas instantáneas en diferentes momentos temporales intermedios.

el tiempo de ejecución del módulo vendrá marcado principalmente por aquel que se obtiene para redes desconectadas.

La Figura 4.9 muestra un ejemplo gráfico en el que se observa cómo los RN (representados con cuadrados rojos) se mueven siguiendo a los puntos de referencia, AP (triángulos invertidos verdes), para mantener e incluso incrementar la conectividad y *throughput*. Las Figuras 4.9(a) y 4.9(d) muestran las posiciones iniciales y finales de todos los nodos implicados, respectivamente, mientras que las Figuras 4.9(b) y 4.9(c) proporcionan una instantánea correspondiente a un momento intermedio entre el inicio y el final de la simulación. Las últimas tres figuras ilustran la capacidad de adaptación del sistema ante los cambios producidos en la topología de la red, tanto en

situaciones en donde los nodos de la red se dispersan (Figuras 4.9(b) y 4.9(d)) como en situaciones en las cuales se concretan dentro de un área más reducida (Figura 4.9(c)).

4.5. Evaluación en simulación

Para validar la eficacia de la solución DRNS, se propone una serie de experimentos empleando escenarios específicos diseñados al efecto. No obstante la validez de la experimentación en escenarios simulados, en la Sección 4.6.1 se llevará a cabo una experimentación en escenarios reales para corroborar las conclusiones aquí obtenidas acerca de la bondad de nuestra propuesta de re-localización de los RN.

En esta sección se describen las características y configuración del entorno de simulación elegido, así como la descripción, discusión y evaluación de los resultados obtenidos durante la experimentación.

4.5.1. Descripción del entorno de simulación

El entorno de simulación considerado en lo que sigue se ha desarrollado en Matlab de manera similar a como se propuso en la solución DKS [131]. Como allí se evitan los detalles de implementación de bajo nivel, ya que el rendimiento del sistema se medirá a través de la conectividad y *throughput* de la red. Por un lado, la conectividad se obtiene evaluando la expresión (4.10), siendo el *throughput* estimado mediante:

$$th(G') = \sum_{\forall u_i, u_j \in U : j > i} ie(G'_i, i, j) \tag{4.25}$$

donde G'_i se corresponde con el grafo obtenido al computar el MST que parte desde el i-ésimo UN. La función $ie(G'_i, i, j)$ mide el *throughput* desde el nodo i al j en G'_i. A su vez, el *throughput* de cada enlace entre dos nodos adyacentes se aproxima mediante la expresión (4.3), al igual que en la solución DKS.

Para tratar de simular una red MANET de la manera lo más realista posible, son tres los aspectos fundamentales que han de considerarse. Estos son: (*i*) el radio de cobertura, (*ii*) las velocidades de ambos tipos de nodos, UN y RN, y (*iii*) cómo se moverán los UN a través del área predefinida, es decir, qué patrón de movilidad seguirán. Hemos de recordar en este punto que los UN no son controlados por DRNS, por lo que hay que definir cómo se mueven estos.

En primer lugar, el radio de cobertura se establece en $1m$ lo que, en conjunción con el área seleccionada definida ($6{,}6m \times 5{,}4m$), asegura la existencia de desconexiones en la red la mayoría del tiempo. Ambos tipos de nodos, UN y RN, poseen el mismo radio de cobertura.

La velocidad de los RN ha de ser cuidadosamente establecida. A priori, podríamos decir que debería ser igual o mayor que la de los UN ya que, en otro caso, el sistema podría no adaptarse a los cambios producidos en el entorno. Una situación especial sería considerar la misma velocidad para ambos tipos de nodos. Más adelante durante la experimentación, se estudia el caso en que ambos nodos poseen la misma velocidad, fijándose esta a $0{,}1m/ts$. Para comprobar el efecto de la velocidad sobre el rendimiento, se obtendrán los resultados correspondientes al aumentar la velocidad de los RN con respecto a la de los UN. En este experimento la velocidad asociada a los UN será de $0{,}1m/ts$, moviéndose los RN a $0{,}15m/ts$. Ha de indicarse que, aunque efectivamente tratamos velocidades, la distancia recorrida por un nodo no está referenciada a una unidad de tiempo establecida, por ejemplo segundos, minutos u horas. En el entorno de simulación establecido, la velocidad se mide de acuerdo a la distancia recorrida durante un tiempo de simulación o *time step* (ts).

El tercer y no menos relevante aspecto es el patrón de movilidad que seguirán los UN. Dos son los elegidos: RWP (*Random Way Point*) [183] y RPGM (*Reference Point Group Mobility*) [184]. El objetivo principal por el cual se seleccionan dos patrones diferentes es estudiar el comportamiento de DRNS cuando se contemplan situaciones en las que los UN se dispersan por todo el área y permanecen la mayoría del tiempo bastante separados (RWP) o, por el contrario, cuando se consideran redes más densas en las que la separación de los nodos es menor y se contemplan movimientos en grupo (RPGM).

4.5.2. Rendimiento y discusión de los resultados

Como ya se describió en la Sección 4.4, DRNS consta de dos grandes módulos que componen su núcleo o *core*. Uno persigue la localización óptima de AP solo en redes que presentan desconexiones, mientras que el otro mueve adecuadamente los RN considerando el dinamismo que presenta la red y la existencia o no de desconexiones. Como se mencionó en la Sección 4.4.1, no es difícil aceptar la aplicabilidad del primero de los módulos dentro de entornos estáticos, simplemente sustituyendo AP por RN. De esta manera, antes de pasar a evaluar el sistema al completo sobre entornos dinámicos MANET, se proponen aquí una serie de experimentos que validan el uso del módulo de localización optimizada en escenarios en los que los nodos de la red permanecen estáticos a lo largo del tiempo.

Aplicación en escenarios estáticos

Considerando el problema de posicionamiento de RN en entornos estáticos a modo comparativo, es posible establecer como soluciones factibles al problema cada una de las subetapas de las que se compone el módulo de localización optimizada (ver Figura 4.5). Denotaremos como solución SELECTIVE a aquella que contempla la etapa de distribución inicial y selección de AP; solución OPTIMIZED a la que contempla todas las etapas del proceso, es decir, el módulo de localización optimizada de AP al completo; y por último la denominada SIMPLE. En esta, a partir de la distribución inicial de los AP, se seleccionan n de ellos de manera simplista, siendo n el número de RN disponibles. La selección se establece de manera que los AP se reparten equitativamente entre particiones.

La Figura 4.10 muestra una comparativa entre los resultados obtenidos mediante las soluciones descritas anteriormente y la proporcionada por el sistema DKS en su aplicación a entornos estáticos. Para la distribución de los UN sobre el área considerada se utiliza la ubicación inicial que se obtiene empleando uno de los patrones de movimiento de UN a utilizar en entornos dinámicos, RWP. En la parte izquierda de la figura se muestran los valores de conectividad, indicándose los correspondientes al *troughput* a la derecha. Unos y otros valores son los promedios obtenidos durante 25 repeticiones con diferentes distribuciones iniciales de UN. Con el fin de considerar un conjunto inicial de AP elevado en relación al número disponible de RN (ver Sección 4.4), se fija $\lambda = 0{,}5$. En cuanto al número de RN que intervienen en los experimentos, este varía desde 0 hasta 3 con objeto de evaluar su impacto sobre el rendimiento. Por otro lado, el número de UN se fija a 3, obteniéndose escenarios poco favorables que dificultan el mantenimiento, recuperación o mejora tanto de la conectividad como del *throughput*.

La Figura 4.10 muestra cómo las soluciones planteadas mejoran el rendimiento frente a DKS a medida que aumenta el número de RN. Tomando como hemos hecho una distribución de UN basada en RWP, el problema se convierte en todo un reto ya que solo se consiguen establecer alrededor de un 10 % de todas las conexiones posibles cuando no se consideran RN. En esta situación, conseguir una red totalmente conectada con 3 o menos RN se torna difícil. Se observa cómo, en el mejor de los casos y por término medio, solo se consigue el 50 % de las conexiones posibles de la red. DKS permanece prácticamente constante a partir de la introducción de 1 RN debido a sus ya discutidas limitaciones (ver Sección 4.2.2). Es más, considerando 3 RN, DKS se ve superada por la solución más simplista y trivial de todas, SIMPLE, que ni siquiera considera los rangos de cobertura de los dispositivos. Sobre la misma figura, los métodos SELECTIVE y OPTIMIZED obtienen mejores resultados que la propuesta DKS, ya que utilizan todos los RN disponibles. Esta conclusión se observa claramente en los resultados al utilizar más de 1 RN.

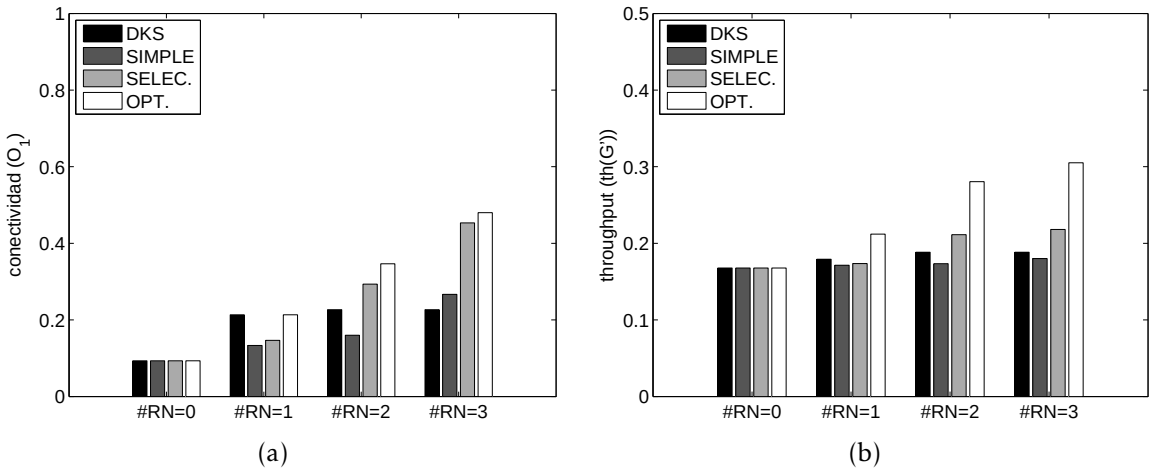


Figura 4.10: Comparativa de rendimiento para varias soluciones al problema de localización de RN aplicado a escenarios estáticos: DKS, SIMPLE, SELECTIVE y OPTIMIZED. En la figura se muestra cómo se comporta la conectividad (a) y el *throughput* (b) a medida que aumenta el número de RN, considerando una distribución inicial de UN basada en el patrón RWP.

La Figura 4.11 ilustra gráficamente las diferencias entre los métodos de localización anteriores. El bajo rendimiento alcanzado por DKS se explica a través de la Figura 4.11(a). DKS solo es capaz de localizar 1 RN sobre los puntos de atracción (triángulos rojos invertidos sobre líneas punteadas del mismo color). El resto de RN no son útiles porque no se tienen en cuenta en la optimización llevada a cabo por el método. En la Sección 4.2.2 se discutieron las deficiencias de este método. Siguiendo con el mismo ejemplo, en la Figura 4.11(b) se muestra el resultado para la opción SIMPLE. Aunque este sencillo y trivial método no consigue conectar ninguno de los 3 nodos de la red, es plausible pensar que, al contrario de DKS, la adición de RN terminase por establecer conexiones entre los nodos de la red. La solución SELECTIVE distribuye los RN de manera inteligente tal y como se muestra en la Figura 4.11(c). Esta solución, al contrario que la SIMPLE, tiene en cuenta el radio de cobertura de los nodos durante el cálculo de los valores de conectividad y *throughput* a partir de los cuales se selecciona el mejor conjunto de RN de entre aquellos distribuidos inicialmente. Finalmente, el método OPTIMIZED mejora la solución anterior especialmente en el *throughput* obtenido ya que no es posible conectar todos los nodos de la red con un número tan bajo de RN. En la Figura 4.11(d) se muestra este resultado, donde puede observarse la redistribución equidistante de RN de manera que incrementa el *throughput* máximo que se puede transmitir desde el nodo 1 al 3. Sin embargo, para la solución SELECTIVE el *throughput* máximo se verá limitado probablemente por el enlace que une el UN1 con RN1, el más largo.

La Figura 4.12 presenta una nueva comparativa de rendimiento de los métodos de localización estáticos que contemplamos anteriormente considerando ahora

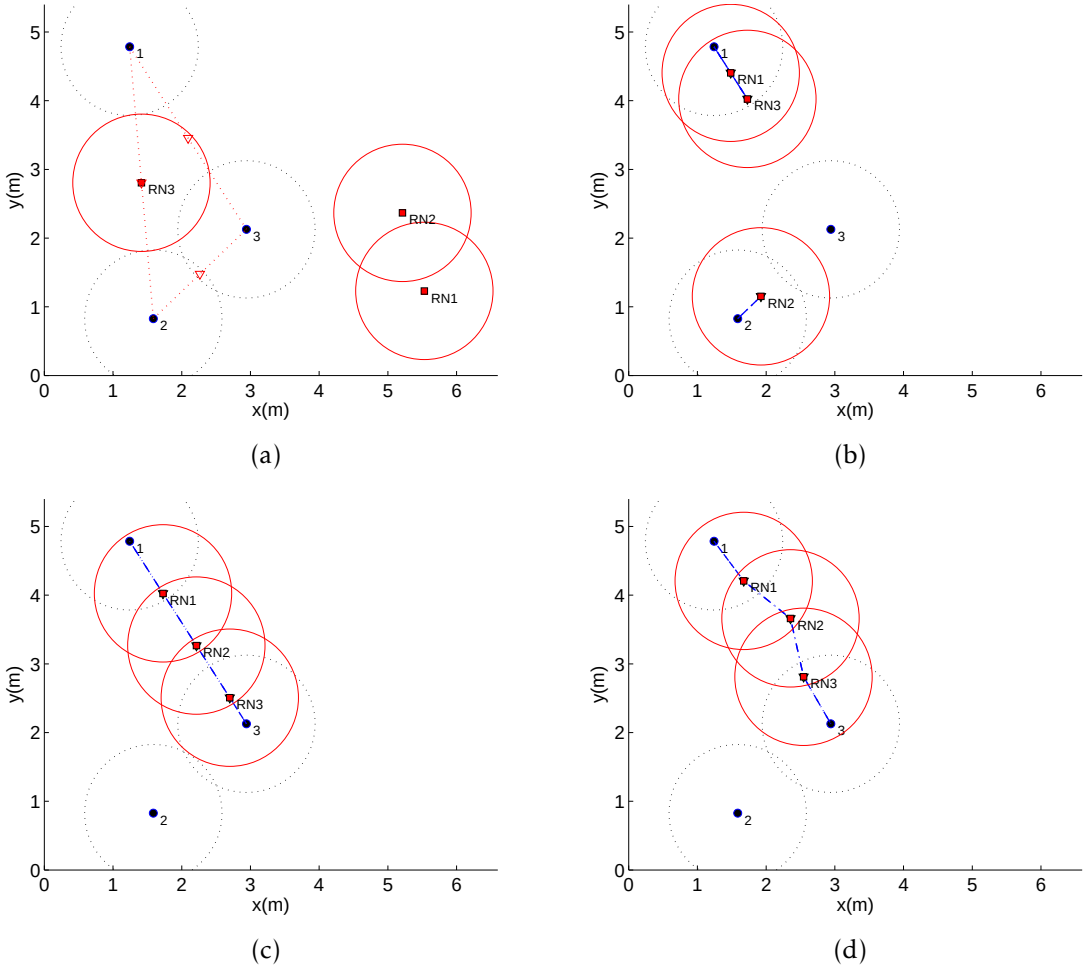


Figura 4.11: Localizaciones de los RN obtenidas para un determinado entorno de red estático de ejemplo, empleando las soluciones: (a) DKS, (b) SIMPLE, (c) SELECTIVE y (d) OPTIMIZED. Se contemplan 3 UN (círculos azules) y se dispone de 3 RN (cuadrados rojos), los primeros distribuidos siguiendo un patrón RWP. En la subfigura (a) se observan los puntos de atracción que computa DKS representados por triángulos invertidos rojos sobre líneas punteadas del mismo color que unen las diferentes particiones.

distribuciones de nodos basadas en el patrón RPGM. Este problema se antoja más simple de resolver en relación al presentado en la Figura 4.10, ya que algunos de los UN forman grupos y, por consiguiente, el número de particiones es menor. En este escenario y a medida que el número de RN se incrementa, se consigue conectar la red en la mayoría de los casos. De nuevo, los métodos SELECTIVE y OPTIMIZED destacan sobre los otros dos, siendo siempre DKS el que ofrece el menor rendimiento. En este caso se observa de manera clara cómo la solución SELECTIVE (ver Sección 4.4.1) tiende a obtener soluciones que abogan por la conectividad. Al contrario de su homóloga OPTIMIZED, que obtiene soluciones que optimizan el *throughput*. Este

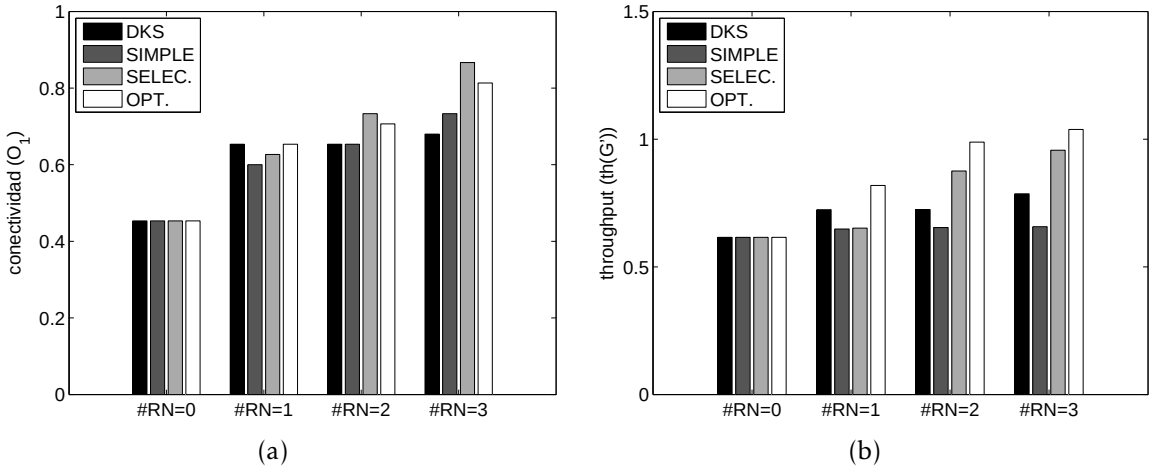


Figura 4.12: Comparativa de rendimiento para la localización de RN aplicado a escenarios estáticos considerando diferentes alternativas: DKS, SIMPLE, SELECTIVE y OPTIMIZED. En la figura se muestra cómo evoluciona la conectividad (a) y el *throughput* (b) a medida que aumenta el número de RN, considerando una distribución de UN basada en el patrón RPGM.

comportamiento está principalmente motivado por el hecho de que la solución OPTIMIZED utiliza en su última etapa de optimización $g(G')$ como función objetivo, que está estrechamente relacionada con el *throughput* de la red.

Aplicación en escenarios dinámicos

Una vez demostrada la vialidad y eficacia de la localización optimizada de AP aplicada a entornos estáticos, procedemos a la evaluación del sistema completo en entornos dinámicos. Las Figuras 4.13 y 4.14 muestran los resultados de rendimiento para el sistema DKS y DRNS cuando los nodos de usuario se mueven siguiendo un patrón RWP. Los valores de conectividad se muestran en la parte superior de las figuras, mientras que los resultados correspondientes al *throughput* obtenido se ilustran en su parte inferior. Sobre las mismas figuras, los resultados mostrados a la izquierda se corresponden con la solución ideada, DRNS, ilustrándose a la derecha aquellos correspondientes a la solución DKS. En ambas figuras se observa la evolución de la media acumulada para las métricas de rendimiento objetivo del sistema a través de 25 repeticiones para escenarios distintos. Los valores de los parámetros λ, número de UN y RN, son los mismos que en el caso estático y se utilizarán con el mismo propósito. Adicionalmente, y con objeto de evaluar el efecto de la velocidad establecida para los RN sobre el rendimiento del sistema, se efectúan experimentos estableciendo dicho valor a $0,1 m/ts$ (ver Figura 4.13) y $0,15 m/ts$ (ver Figura 4.14). La velocidad de los UN permanece constante en $0,1 m/ts$ en todos los experimentos.

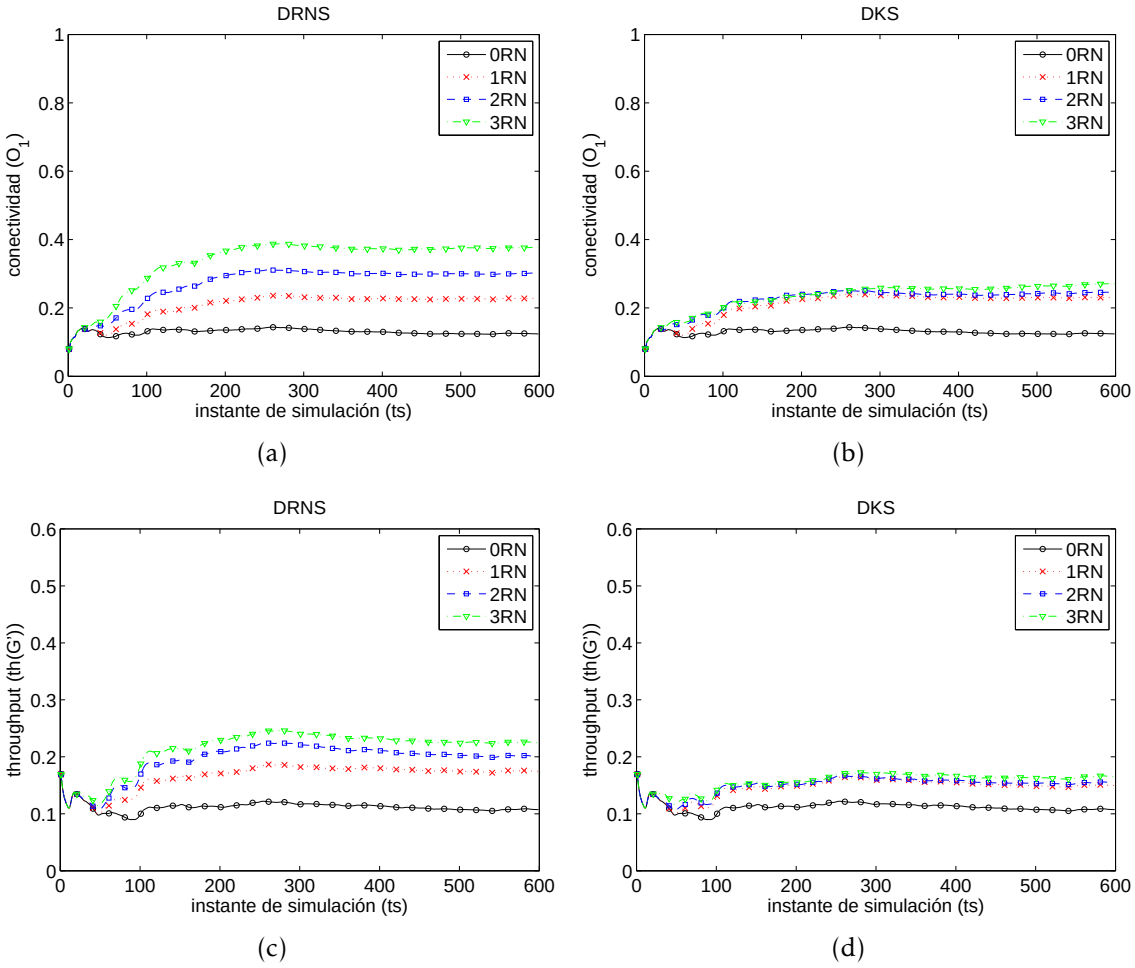


Figura 4.13: Evolución del valor de la media acumulada correspondiente a la conectividad y el *throughput* de la red cuando se consideran 3 UN y el número de RN varía desde 0 a 3. RWP es el patrón de movimiento elegido y las velocidades de los nodos implicados se fijan a $0{,}1 m/ts$. Las subfiguras (a) y (b) muestran la conectividad obtenida para la propuesta DRNS y DKS, respectivamente. En lo referente al *throughput*, las subfiguras (c) y (d) muestran los correspondientes resultados para ambas propuestas.

DRNS supera a DKS en todos los casos cuando se emplean 2 o más RN tanto en la conectividad como en el *throughput*. Se observa que el uso de más de 1 RN en DKS no tiene un impacto significativo en el rendimiento. Esta realidad evidencia la limitación de DKS a la hora de aprovechar los RN disponibles. Por el contrario, DRNS hace uso de todos los RN y mejora el rendimiento de la propuesta a medida que aumenta su número. Ambos enfoques muestran menor rendimiento cuando ambos tipos de nodos, UN y RN, poseen la misma velocidad, incrementándose aquel al aumentar esta (ver Figura 4.14). Esto se debe principalmente al hecho de que el

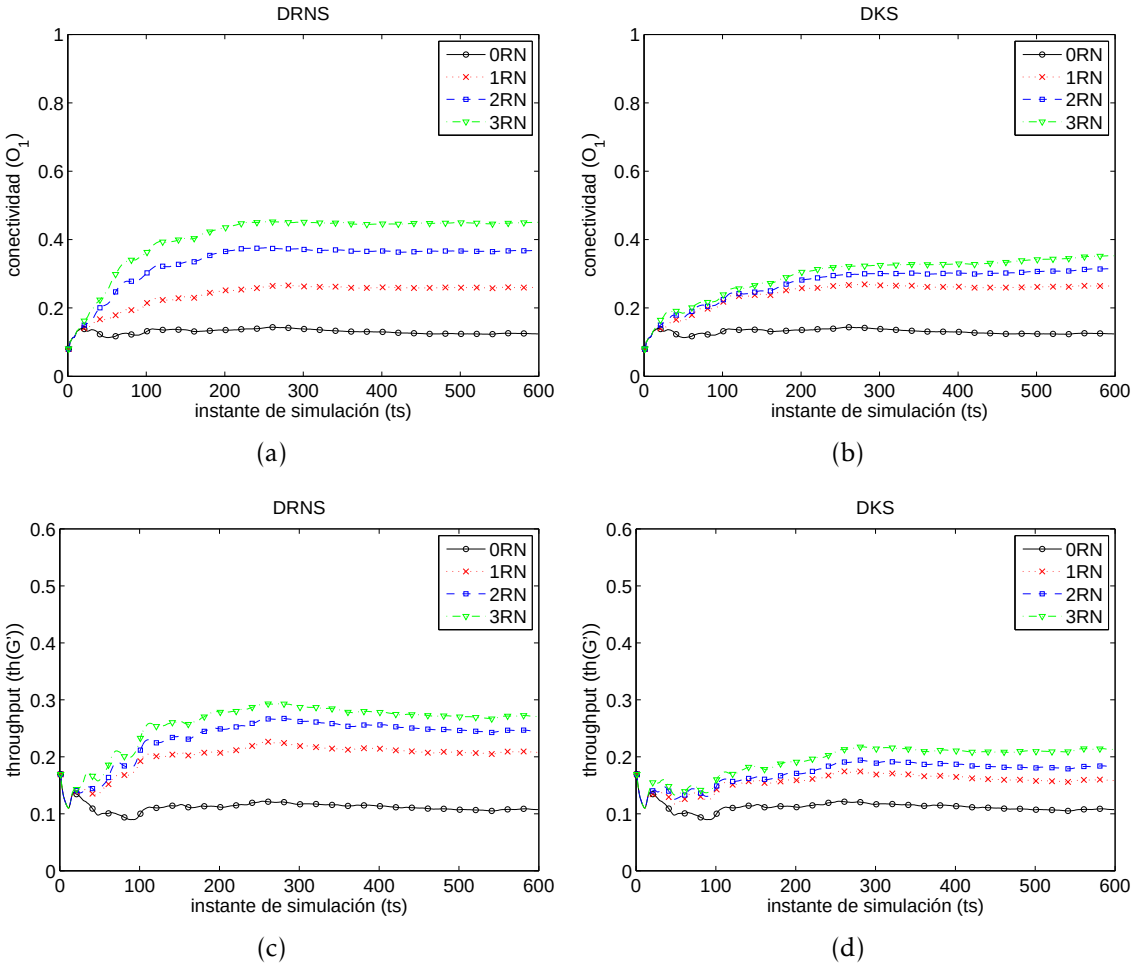
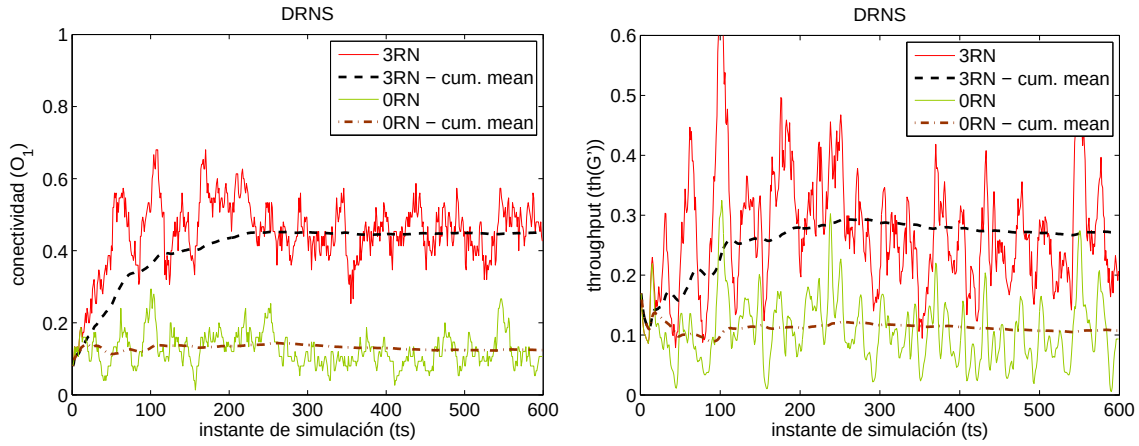


Figura 4.14: Rendimiento del sistema aumentando la velocidad de los RN a 0,15m/ts en comparación con la Figura 4.13. De igual forma, se emplean 3 UN y variamos el número de RN desde 0 a 3, moviéndose los primeros siguiendo un patrón RWP. Las subfiguras (a) y (b) muestran los resultados de conectividad para las soluciones DRNS y DKS, respectivamente, ilustrando las subfiguras (c) y (d) los correspondientes resultados para el *throughput*.

sistema no es capaz de adaptarse lo suficientemente rápido a los cambios producidos en su entorno. Para el caso de DKS este hecho es especialmente relevante ya que la red permanece prácticamente todo el tiempo particionada y, como consecuencia, DKS hace un mayor uso de la función $O_3(R,A)$ (ver (4.13)) en la optimización. A modo de recordatorio, es importante hacer hincapié en la errónea definición de dicha función, la cual no hace uso de todos los RN disponibles en el proceso; es más, solo considera uno de ellos, dejando el resto inservible de cara al rendimiento de la red.

En la evolución de las métricas de rendimiento del sistema siempre se observa un periodo de transición inicial seguido por una etapa en la que el rendimiento permanece en un estado estable. Este periodo inicial comprende el intervalo de tiempo



(a) (b)

Figura 4.15: Rendimiento de la propuesta DRNS (conectividad (a) y *throughput* (b)) empleando 0 y 3 RN, 3 UN y RWP como patrón de movimiento para los últimos. Se muestra tanto la media acumulada (líneas discontinuas) como los valores instantáneos (líneas continuas) obtenidos a lo largo de tiempo.

en el que los RN parten de su posición inicial predefinida para los experimentos (si consideramos el área rectangular utilizada, inicialmente se localizan en la esquina inferior izquierda), hasta conseguir adaptarse al dinamismo de la red a través de la optimización de su posición. De manera ilustrativa, se muestra en la Figura 4.15 tanto la evolución de la media acumulada como los valores instantáneos en cada instante de simulación para las métricas utilizadas. Un hecho indicativo del dinamismo del escenario empleado son las fluctuaciones que se observan en los resultados. A pesar del escenario altamente cambiante, se aprecia cómo el sistema es capaz de adaptarse a la dinámica impuesta por los movimientos de los nodos en la red.

Adicionalmente, en la Figura 4.16 se comparan los resultados obtenidos para DRNS frente a una solución de posicionamiento de RN básica, en la que el movimiento de los RN es totalmente aleatorio. El objetivo principal que se persigue con la introducción de dicha solución es comparar el rendimiento en simulación con el obtenido sobre el entorno real, fijando además unos valores base para la conectividad y el *throughput*. De aquí en adelante, esta solución pasará a denominarse RAND. Se observa cómo, ante un número igual de nodos en la red, la solución DRNS posiciona los RN de manera eficiente acorde con los requerimientos del sistema.

La experimentación previa se corresponde con escenarios con nodos muy dispersos dentro del área considerada. Esto hace que se produzcan desconexiones la mayor parte del tiempo. Con ánimo de evaluar el rendimiento de DRNS con otros patrones de movimiento que no abogan por la dispersión de los nodos, a continuación se utiliza RPGM. En la Figura 4.17 se muestra la comparativa con la solución RAND. La propuesta básica RAND obtiene mayor rendimiento con respecto al uso de patrones

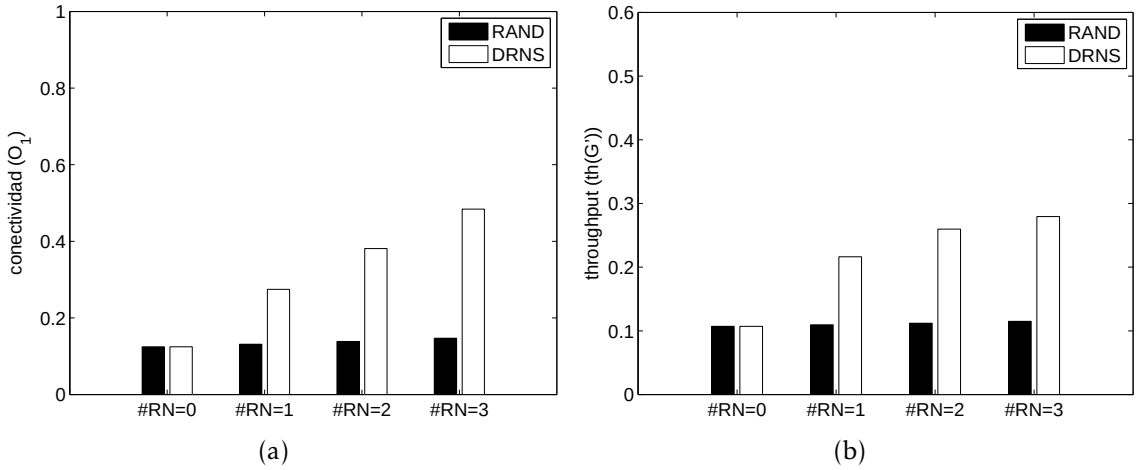


Figura 4.16: Comparativa de rendimiento para las soluciones DRNS y RAND empleando RWP como patrón de movimiento para los UN. Las subfiguras (a) y (b) muestran los valores obtenidos para la conectividad y el *throughput*, respectivamente, en función del número de RN empleado, cuando se fija el número de UN a 3.

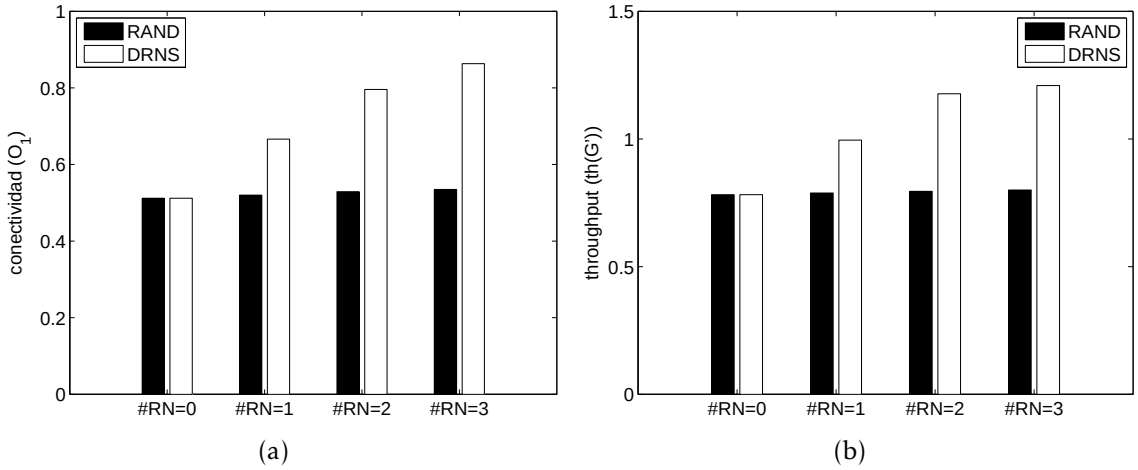


Figura 4.17: Comparativa de rendimiento para las soluciones DRNS y RAND empleando RPGM como patrón de movimiento para los UN. Las subfiguras (a) y (b) muestran los valores obtenidos para la conectividad y el *throughput*, respectivamente, en función del número de RN empleado, cuando se fija el número de UN a 3.

basados en RWP. No obstante, incluso en situaciones como estas en las que a priori sería más sencillo conectar un número de nodos más elevado y por tanto aumentar los niveles de conectividad o *throughput*, DRNS consigue mejorar a la anterior, especialmente a medida que se incrementa el número de RN utilizados.

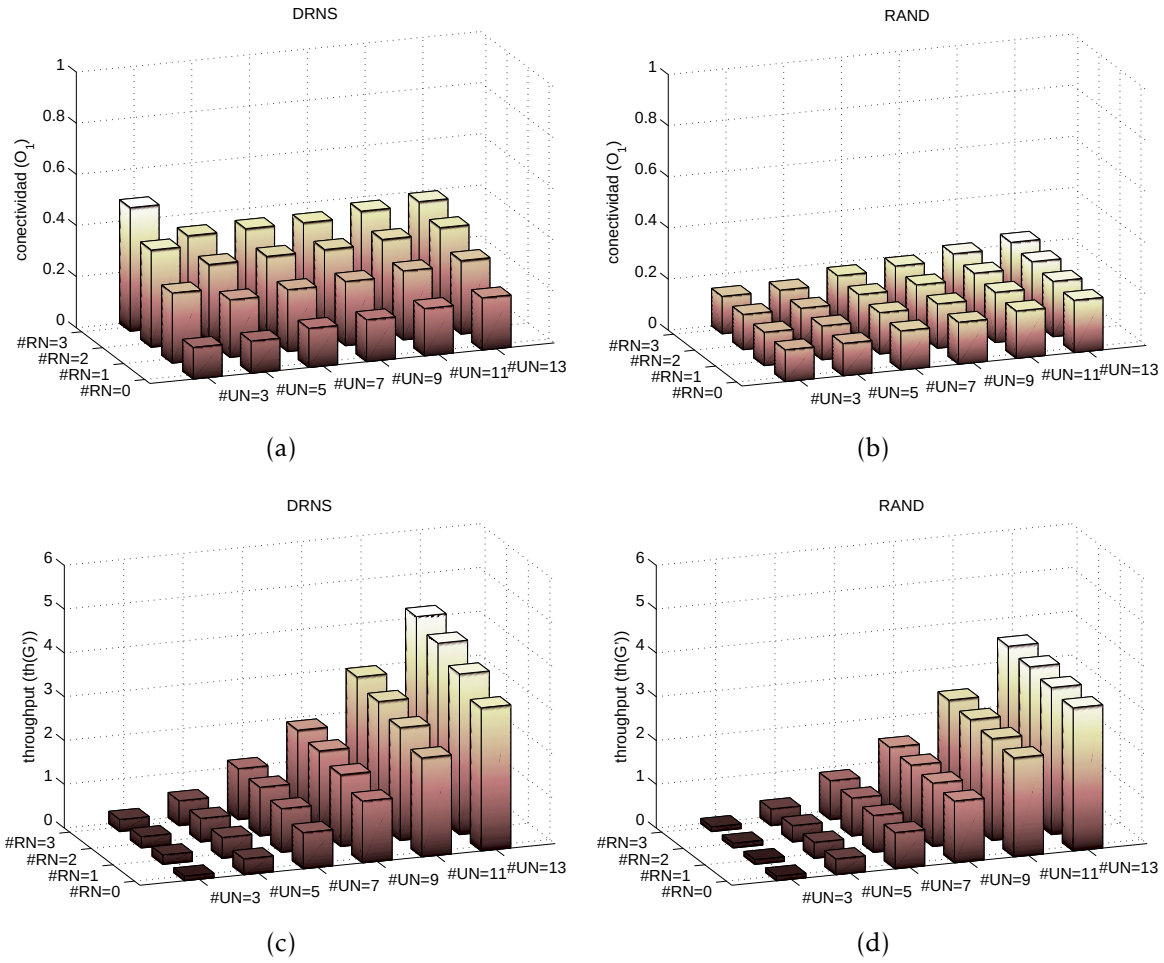


Figura 4.18: Evolución de la conectividad y del *throughput* obtenidos cuando se varía el número de UN desde 3 hasta 13 y el número de RN desde 0 hasta 3. Las subfiguras (a) y (c) ilustran los resultados para la solución DRNS, mientras que las subfiguras (b) y (d) se refieren a la solución RAND. En ambos casos se utiliza RWP como patrón de movimiento.

Finalmente, para evaluar cómo se comporta el sistema considerando un número elevado de UN, aumentaremos el número de UN hasta 13, variando el de los RN de 0 a 3. Las Figuras 4.18 y 4.19 ilustran los resultados obtenidos usando RWP y RPGM como patrones de movimiento, respectivamente. En general, DRNS se comporta como era de esperar: el rendimiento del sistema se incrementa con el número de RN utilizados.

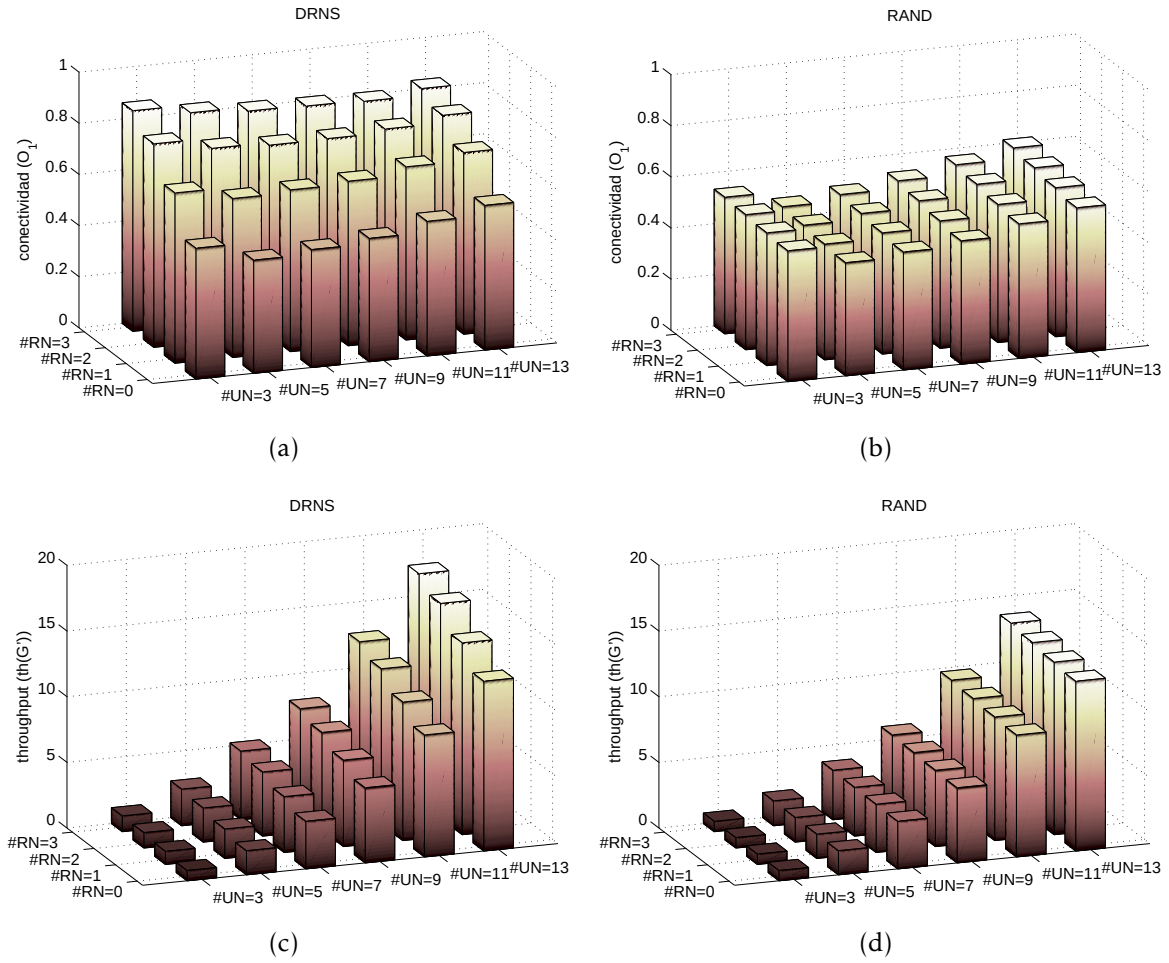
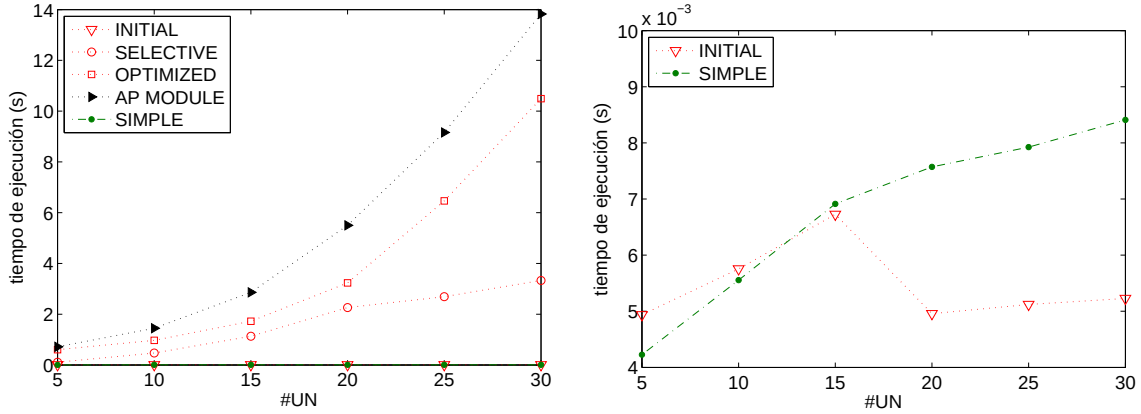


Figura 4.19: Evolución de la conectividad y del *throughput* obtenidos cuando se varía el número de UN desde 3 hasta 13 y el número de RN desde 0 hasta 3. Las subfiguras (a) y (c) ilustran los resultados para la solución DRNS, mientras que las subfiguras (b) y (d) se refieren a la solución RAND. En ambos casos se utiliza RPGM como patrón de movimiento.

Estudio del tiempo de ejecución

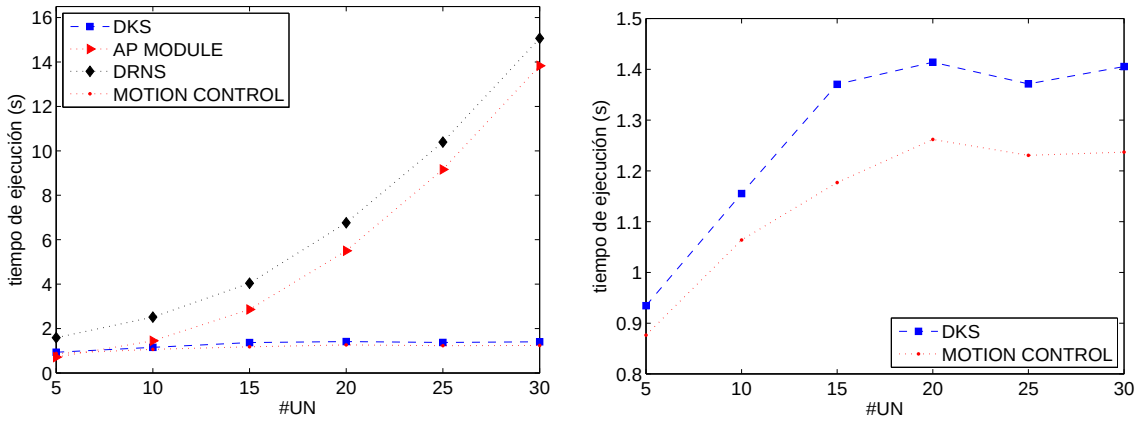
Con objeto de medir y evaluar la complejidad computacional de DRNS y establecer una comparativa con su homóloga DKS de referencia, a través de las Figuras 4.20 y 4.21 se muestra la evolución del tiempo de ejecución consumido por cada uno de los procedimientos implicados a medida que el número de UN aumenta. Dado que los escenarios planteados aquí sugieren, a priori, que en la mayoría de los casos se presenten desconexiones en la red, computaremos el tiempo de ejecución únicamente para este caso, siendo este el que tendrá un mayor impacto potencial sobre el retardo computacional de la solución.



(a) (b)

Figura 4.20: Dependencia del tiempo de ejecución con respecto al número de UN para el caso del módulo de localización optimizada de AP. La subfigura (a) muestra los tiempos del módulo en general y de las subetapas que lo componen. También muestra la evolución de la solución alternativa de localización de AP SIMPLE. A la derecha, en la subfigura (b), se presenta una inspección más cercana de la evolución para la subetapa distribución inicial y la solución SIMPLE, ya que estas dos poseen tiempos cercanos a cero.

En la Figura 4.20(a) se observa la evolución del tiempo de ejecución para el módulo de localización optimizada de AP así como para cada una de sus subetapas: distribución inicial (INITIAL), selección de AP (SELECTIVE) y optimización de AP (OPTIMIZED). Adicionalmente, se indica también el tiempo de computación empleado por la solución alternativa de localización SIMPLE que se introdujo en la Sección 4.5.2. Es patente que son las etapas SELECTIVE y OPTIMIZED las que suponen la mayor carga computacional. Atendiendo a su complejidad computacional antes discutida, $\mathcal{O}((k^2 - m^2) \cdot n^4)$ y $\mathcal{O}((n^4 + m) \cdot p \cdot i)$ respectivamente, se observa que la subetapa de optimización de AP repite $p \cdot i$ veces una operación de orden n^4, con $p = 25$ e $i = 50$. En su lugar, la selección de AP repite $(k^2 - m^2)$ veces una operación de orden n^4 que, a priori, pareciera tener un impacto mayor sobre el tiempo de cómputo. Sin embargo, su tiempo de ejecución solo crece hasta cierto número de UN ya que el número de posiciones candidatas iniciales de RN, k, también lo hace. Es más, incluso decrece rebasado un determinado número de ellos. Debido al propio aumento de UN permaneciendo constante el área de la red, se producen menos particiones y por lo tanto k disminuye. Este efecto se aprecia también en la evolución que muestra el procedimiento de distribución inicial (Figura 4.20(b)). Como consecuencia del comportamiento de la etapa de selección ante un número elevado de UN, y como se aprecia en la Figura 4.20(a), es la subetapa de optimización de AP la que introduce mayor retardo de computación, condicionando el tiempo total consumido por este módulo. Como era de esperar, la subetapa de distribución inicial de AP apenas sí introduce demora. Algo similar ocurre con la solución SIMPLE: su



(a) (b)

Figura 4.21: Dependencia del tiempo de ejecución con respecto al número de UN para el sistema DRNS completo. La subfigura (a) muestra la evolución del retardo para DRNS en comparación con el obtenido por DKS. Con fines comparativos, también se presentan los resultados para el módulo de localización optimizada de AP. A la derecha, en la subfigura (b), se presenta la tendencia del sistema DKS y el módulo de control de movimiento de los RN de manera separada con objeto de apreciar su evolución.

demora computacional es despreciable aunque, en contraposición, su rendimiento se ve superado por las soluciones SELECTIVE y OPTIMIZED.

A través de la Figura 4.21(a) y de manera similar a los resultados expuestos en la Figura 4.20(a), se compara la evolución obtenida del tiempo de ejecución del sistema DRNS completo. Es de reseñar la notable influencia del módulo de localización optimizada de AP. En comparación con el sistema DKS, DRNS introduce mayor retardo computacional a medida que aumenta el número de UN. Por otro lado, atendiendo la complejidad computacional obtenida para DKS, $\mathcal{O}((m \cdot k + n^2) \cdot p \cdot i)$ siendo $k = \frac{n \cdot (n-1)}{2}$ el número de AP computado cuando la red está totalmente desconectada (considerado el peor caso desde el punto de vista computacional), son esperables tiempos de ejecución más elevados, sobre todo con un alto número de UN. El cálculo de la conectividad de la red ($O_1(G)$) que deriva en el término n^2, así como su homólogo k, disminuyen alcanzado un cierto número de UN. Al igual que para la selección de AP, este comportamiento se debe a que el área que abarca la red permanece constante haciendo que disminuya el número de particiones conforme aumenta el número de UN. Este efecto también se observa en la Figura 4.21(b) tanto para DKS como para el módulo de control optimizado de movimientos (MOTION CONTROL). En el cómputo general del tiempo de computación invertido por el sistema DRNS, el módulo de control de movimientos añade cierta demora aunque, como ya se indicó anteriormente, sigue siendo el módulo de localización optimizada de AP el que condiciona su ejecución. A modo de resumen, la Tabla 4.1 muestra la

	$\mathcal{O}(.)$
DKS	$\mathcal{O}((m \cdot k + n^2) \cdot p \cdot i)$
INITIAL	$\mathcal{O}(n^2 \cdot \log n)$
SELECTIVE	$\mathcal{O}((k^2 - m^2) \cdot n^4)$
OPTIMIZED	$\mathcal{O}((n^4 + m) \cdot p \cdot i)$
MOTION CONTROL	$\mathcal{O}(m^2 \cdot p \cdot i)$

Tabla 4.1: Complejidad que presentan las soluciones de posicionamiento de RN estudiadas. Los parámetros m y n denotan el número de RN y UN respectivamente; k representa el número de AP candidatos; p es el número de partículas consideradas en PSO; e i el número máximo de iteraciones de dicho algoritmo.

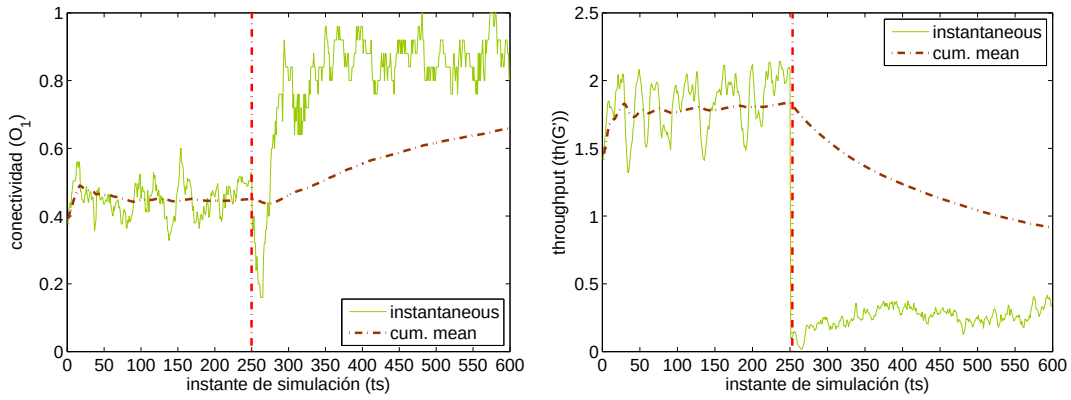
complejidad computacional de las distintas soluciones de posicionamiento de RN estudiadas.

Conclusiones preliminares

De los experimentos y resultados anteriores se pueden obtener conclusiones relevantes acerca del rendimiento y eficiencia de DRNS. Si bien es cierto que la complejidad asociada a la solución planteada es mayor en comparación con otras propuestas, también lo es su rendimiento. Un parámetro que podría condicionar su uso desde el punto de vista computacional es el número de UN considerados (n), ya que su incremento tiene un impacto directo sobre los tiempos de ejecución. Este hecho puede ser condicionante dependiendo de los requisitos funcionales del sistema, siendo DRNS de aplicación válida en escenarios donde la conectividad de la red es un requisito principal. Por ejemplo, aquellas situaciones de desastres naturales o acciones de rescate en caso de emergencia en las cuales es usual que el número de nodos a conectar es reducido. En escenarios como los anteriores, que tienen solución compleja, DRNS se presta a ser una alternativa factible frente a soluciones como DKS, ofreciendo mejor rendimiento y similar tiempo de ejecución. Durante la experimentación real veremos que el tiempo de ejecución no es un parámetro que tenga un impacto relevante sobre el rendimiento del sistema, hecho que concluye de manera definitiva el uso de DRNS.

4.5.3. Aplicación de DRNS como sistema de respuesta/tolerancia

Previamente mencionado, los continuos cambios que se producen en un entorno MANET son principalmente motivados por el inherente dinamismo de la red. No obstante, los cambios producidos en la topología de la red podría tener orígenes



(a) (b)

Figura 4.22: Rendimiento de recuperación del sistema DRNS considerando 3 nodos maliciosos, 5 UN y 3 RN. La subfigura (a) muestra cómo evoluciona la conectividad, mientras que la subfigura (b) la evolución del *throughput*. La línea roja vertical y discontinua muestra el instante de tiempo en el que se produce el ataque.

distintos como, por ejemplo, la aparición de comportamientos maliciosos o fallos en los nodos. De esta forma, el sistema DRNS se podría usar como una solución de respuesta/tolerancia ante dichos eventos. Desde el punto de vista de la seguridad en red, algunos de los ataques más perjudiciales que afectan a las MANET son, entre otros, los denominados como *dropping attacks* [5].

Con el propósito de validar el sistema propuesto aquí como solución de reacción/-tolerancia frente a este tipo de ataques (de manera similar, podría ser aplicado para solventar los efectos producidos por ataques como *route poisoning attacks*, *sinkhole* or *wormhole*), hemos realizado varios experimentos que muestran cómo nuestra solución es capaz de reaccionar para mitigar los efectos provocados por este tipo de actuaciones maliciosas. Para ello consideramos la presencia de 3 nodos maliciosos, 3 RN y 5 UN, estos últimos moviéndose acorde al patrón de movimiento RPGM. Los nodos maliciosos actúan como nodos normales hasta un determinado tiempo de simulación, después del cual se comportan como nodos *dropper*.

Partimos de la existencia de un mecanismo de detección previamente desplegado complementario a nuestra propuesta (por ejemplo [185]). En consecuencia, una vez que el ataque es detectado, el sistema de respuesta DRNS se inicia para recuperar el rendimiento de la red. La Figura 4.22 muestra cómo evoluciona la conectividad y el *throughput* a lo largo del tiempo. En la misma figura observamos el instante de tiempo en el que se produce el ataque, justo en $ts = 250$. Como se esperaba, los valores de rendimiento disminuyen cuando se produce el ataque, aunque es notable la recuperación que se produce en estos valores tras las actuación del sistema DRNS. Nótese que el máximo valor del *throughput* se reduce de acuerdo a (4.25) ya que el

número de nodos de la red disminuye debido a la exclusión de los nodos maliciosos de la red.

4.6. Aplicación a entornos MANET reales: IDSIA *Swarn Robotics Laboratory*

La gran mayoría de las propuestas para la localización de RN se validan a través de herramientas y entornos simulados. Esto está principalmente motivado por el ahorro en el esfuerzo invertido de cara a la implementación o despliegue, el tiempo (siempre un bien muy preciado) empleado y los costes implicados en la realización de los oportunos experimentos. Sin embargo, hay ocasiones en las que no es posible simular o aproximar con la suficiente exactitud aspectos relevantes de un escenario real en un entorno simulado. Así, en ocasiones, los resultados obtenidos en simulación pueden establecer de manera preliminar la eficacia y/o eficiencia de una solución, pero es posible que tras su despliegue en escenarios reales esta se comporte de manera muy diferente.

En nuestro caso, seguidamente se abordan experimentaciones en un entorno real a fin de corroborar los resultados obtenidos en simulación para nuestra propuesta de re-localización de RN. Para ello, primero se describe el entorno real utilizado basado en el empleo de nodos robotizados y después se evalúan y discuten los resultados de acuerdo a experimentos basados en los ya realizados en simulación. El rendimiento conseguido evidenciará la validez de los desarrollos realizados.

4.6.1. Descripción del entorno real

Son muchos los aspectos que han de ser considerados antes de la puesta en marcha del sistema DRNS dentro de entornos reales, especialmente en escenarios MANET cerrados o interiores. Por ejemplo, son de importante relevancia: el radio de cobertura, el movimiento de los nodos, el control de estos y las comunicaciones asociadas, aspectos físicos como la energía consumida por los nodos y cómo evitar obstáculos, entre muchos otros.

Con el fin de evaluar la propuesta usaremos un escenario de red real cuyos nodos son pequeños dispositivos móviles o robots. Es el instituto IDSIA (*Institute Dalle Molle for Artificial Intelligence*) [186], en concreto a través del laboratorio de robótica *IDSIA Swarm Robotics Laboratory* [187], el que provee dicho entorno. Este laboratorio está conformado por una habitación cerrada que posee un área útil de experimentación de $6{,}6m \times 5{,}4m$. El área disponible es realmente mayor pero, con el objetivo de establecer una zona de seguridad entre los robots y las paredes circundantes, se determinan las

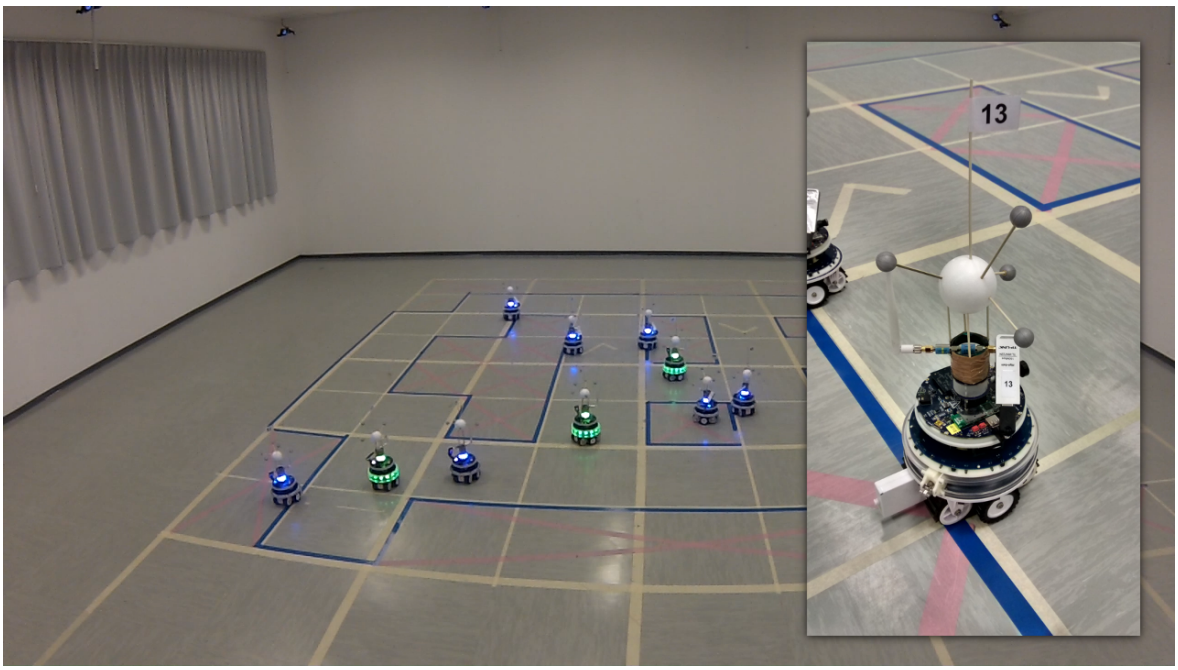


Figura 4.23: Instantánea del escenario real considerado para la experimentación y detalle de uno de los robots utilizados (*foot-bots*). En este caso se dispone de 7 UN (con luz azul) y 3 RN (con luz verde) posicionados de manera optimizada.

anteriores dimensiones rectangulares. En la Figura 4.23 se observa una fotografía del entorno durante un experimento. En la misma figura, a la derecha, se detalla la imagen de uno de los robots utilizado (los llamados *foot-bots* [188]).

Antes de empezar a utilizar el entorno real provisto son varias las cuestiones a las que hemos de dar respuesta. La primera es cómo se deberían mover los robots por el área en cuestión. El movimiento de un robot real no es nada trivial. Entre otros aspectos, es necesario conocer de antemano las dimensiones del dispositivo, su localización, su orientación, la ubicación de los demás robots, sus velocidades, cómo evitar obstáculos, etc. Tales cuestiones fueron abordadas y solventadas a través de la integración y despliegue del algoritmo de navegación y evitación de obstáculos descrito en la referencia [189], que emula el comportamiento humano ante estas situaciones. Dicho algoritmo se configura principalmente con la velocidad deseada para los nodos implicados, así como la zona o área de seguridad alrededor de ellos, con el fin de evitar colisiones con otros robots o con las paredes. El primero se selecciona de manera empírica a través del simulador ARGoS [190]. Dicho simulador es ideal para estos fines, ya que replica fielmente el entorno real que se va a utilizar. A partir de estas simulaciones iniciales se obtiene que la velocidad para los UN es $0,1 m/s$ mientras que para los RN se establece a $0,2 m/s$. Ya que cada robot tiene forma circular con un diámetro de $14 cm$, se decide que cada uno de ellos no puede posicionarse a menos de $24 cm$ de distancia de cualquier otro desde su centro geométrico.

Esto es, se establece una zona de seguridad circular de 5*cm* alrededor de cada robot. Adicionalmente, los robots que hacen las veces de UN se moverán siguiendo sendos patrones RWP y RPGM, tal y como se implementó en simulación.

Otra cuestión importante es conocer en qué posición está ubicado cada robot y, más relevante aún, dónde se encuentran los demás. Esta información es totalmente necesaria para, por un lado, el sistema DRNS que guía a aquellos *foot-bots* que actúan como RN y, por otro, para la propia navegación y evitación de obstáculos en cada uno de los robots individuales. Con estos requisitos en mente, se despliega un sistema de seguimiento compuesto por una serie de cámaras infrarrojas distribuidas homogéneamente en el techo de la habitación. En la Figura 4.24 se muestra un esquema general en el que se indican todos los elementos que actúan en el sistema en su conjunto, y que son necesarios para la implementación, despliegue y pruebas de la propuesta de posicionamiento de RN aquí planteada.

Mencionado anteriormente, la información de posición de cada robot ha de ser difundida para su conocimiento en toda la red en tiempo real, dado el dinamismo del sistema. Para esto se diseña, desarrolla e implementa un módulo software denominado *tracker interface* (ver Figura 4.24). Este módulo (que se ejecuta como un proceso más dentro de la estación central, al cargo de la gestión global del entorno) se encarga de obtener la información de localización (es decir, las coordenadas (X,Y)) de cada nodo proporcionada por el sistema de seguimiento. Después se la envía a cada uno de los robots y al módulo DRNS de manera regular.

Una cuestión adicional y relevante es la evolución temporal del sistema en su conjunto. En entornos simulados, DRNS computa y optimiza las posiciones de los RN empleando para ello un determinado tiempo al que llamamos tiempo de simulación (*ts*). Sin embargo, no existe equivalencia entre cada paso o *ts* en simulación y el tiempo tal y como transcurre en el entorno real. Así, en simulación la red no avanza hasta que transcurre un *ts* pero, en el experimento real los UN siguen moviéndose continuamente solo dependiendo de su patrón de movimiento y su velocidad. Así, es necesario que DRNS adapte su ejecución según las restricciones dinámicas y de tiempo real implícitas a este tipo de escenarios.

Adicionalmente y de manera transversal a todos los componentes implicados en el despliegue, es necesaria la existencia de un sistema de comunicaciones para el traspaso de la información de localización de los robots. Tras una extensiva búsqueda, se considera el sistema de comunicaciones LCM (*Lightweight Communications and Marchalling*) [191] como una propuesta viable para tal fin. LCM se basa en el paradigma publicador/subscriptor cuya misión es establecer comunicaciones entre los robots y la estación central. LCM ofrece un *middleware* de comunicaciones fiable, escalable, simple de utilizar y con baja latencia que lo hace adecuado para su fin dentro del esquema propuesto. Gracias a LCM se definen dos canales sobre los cuales se enviarán determinados mensajes. Estos son los canales denominados TRACK y

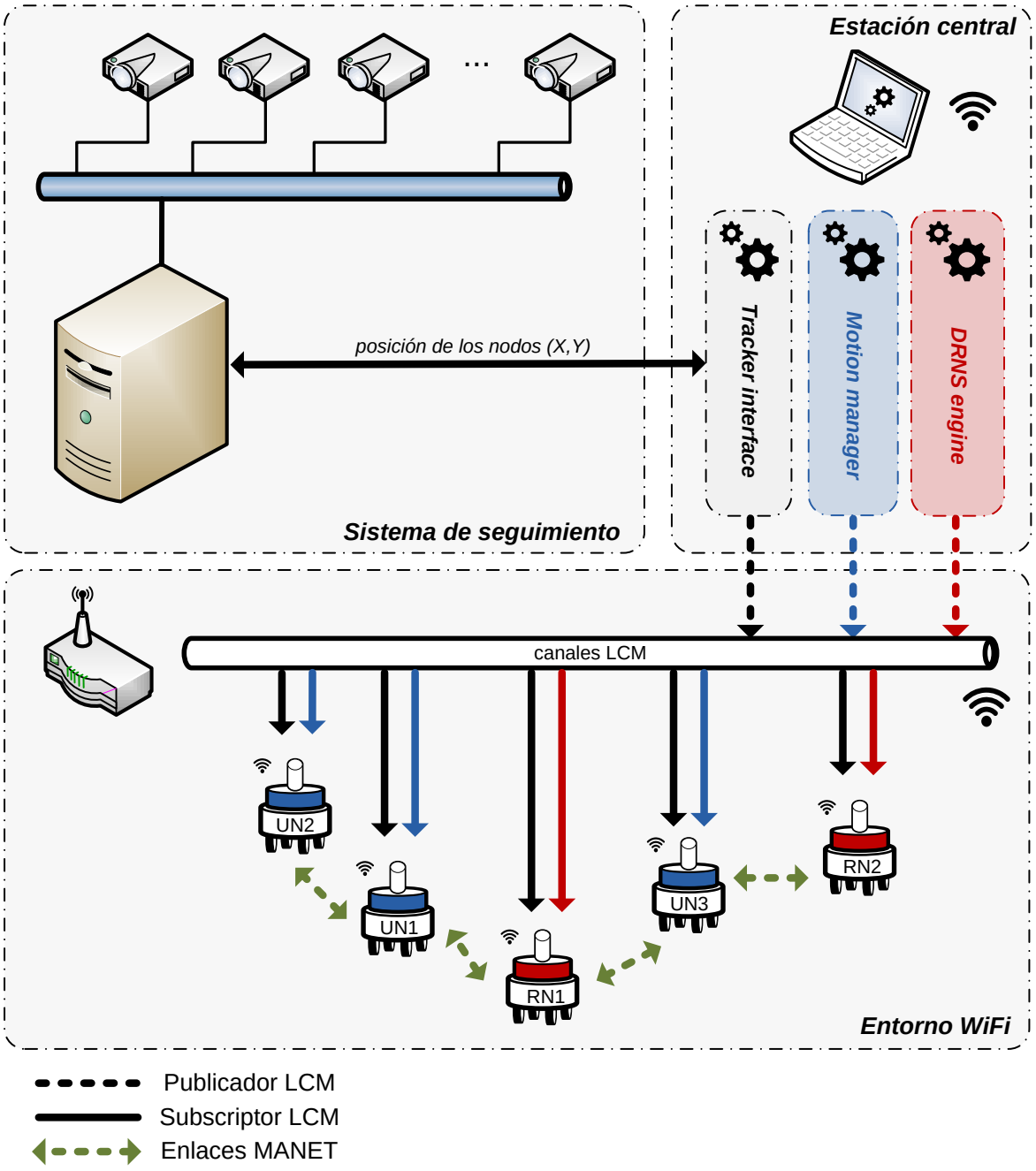


Figura 4.24: Arquitectura y elementos que intervienen en el despliegue e implementación de la solución DRNS en el entorno real seleccionado.

TARGET. El primero lo utiliza el módulo *tracker interface* para publicar mensajes que contienen las información de localización de todos los robots. De esta manera, un robot subscrito a dicho canal conocerá no solo su posición actual, sino también la de todos los demás. El canal TARGET se utiliza para informar a los UN y RN de

su próxima posición objetivo hacia la que han de dirigirse. Además de los propios robots, tanto el módulo DRNS (*DRNS engine*) como el de gestión del movimiento de los UN, *motion manager*, utilizarán dicho canal con el fin de publicar los mensajes que contendrán las nuevas posiciones para los RN y UN, respectivamente. Por supuesto, ambos tipos de nodos también habrán de suscribirse al canal TARGET para así recibir las nuevas posiciones.

4.6.2. Rendimiento y discusión de los resultados

Para evaluar el rendimiento provisto por DRNS cuando se despliega sobre el entorno real *IDSIA Swarm Robotics Laboratory* descrito previamente, se replican algunos de los experimentos realizados en simulación. La principal diferencia que existe entre ambos grupos de experimentos es que ahora se consideran 5 repeticiones del mismo experimento en lugar de las 25 realizadas en simulación, principalmente motivado por el alto coste de ejecución de un experimento en dicho laboratorio. Por otro lado, el parámetro λ se fija al valor $0,5$ al igual que en el caso simulado, siendo las velocidades de los RN establecidas a $0,2m/s$, justo el doble que las de los UN y ligeramente superior a la utilizada en simulación. Este aumento de la velocidad trata de compensar (como se apreció durante los experimentos iniciales en ARGoS) los efectos producidos por la aceleración y cambios de dirección que no fueron tenidos en cuenta en la experimentación en entornos simulados de la Sección 4.5.2.

La Figura 4.25 muestra la evolución de la conectividad y el *throughput* obtenida con la solución DRNS. Se puede observar un comportamiento similar al obtenido en simulación: el rendimiento del sistema crece con el número de RN. No obstante, son dos las principales y notables diferencias con respecto a los resultados simulados (ver Figuras 4.14 y 4.15):

1. El sistema proporciona una menor adaptación a los cambios producidos en la topología de la red.
2. Los parámetros de rendimiento son menores en general que aquellos obtenidos en simulación.

Claramente, el escenario real introduce algunos elementos inesperados que no fueron considerados en el caso de la simulación. Así, estas diferencias son principalmente motivadas por el hecho de que los robots han de evitar obstáculos como son sus propios semejantes o las paredes de alrededor. Otro aspecto que afecta a los resultados obtenidos es la velocidad de los robots. En el entorno real un nodo está sujeto a cambios de trayectoria, aceleraciones y deceleraciones cuando ha de moverse desde un punto A a un punto destino B. Esto no ocurre, sin embargo, en simulación. En resumen, las características propias del entorno real hacen que el rendimiento

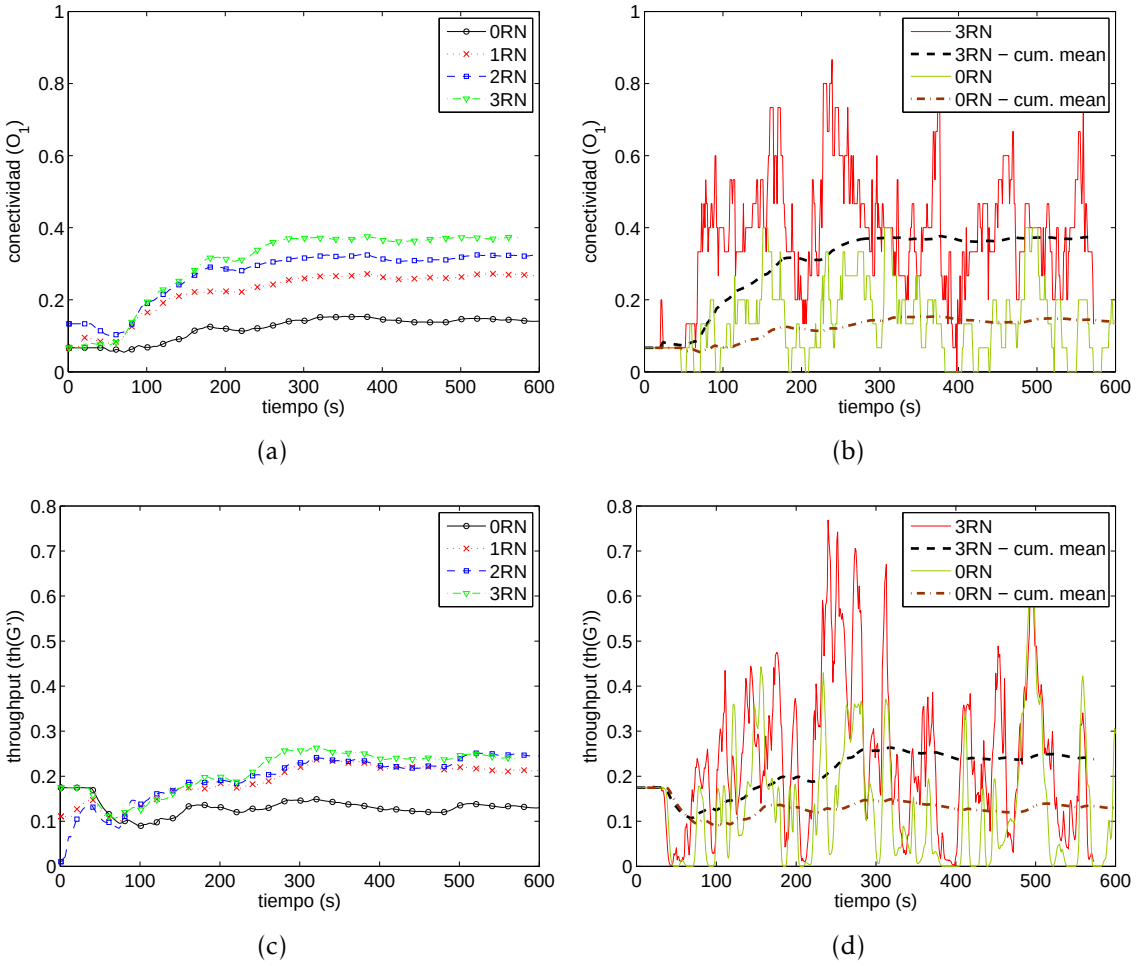


Figura 4.25: Rendimiento obtenido con el sistema DRNS desplegado en el *IDSIA Swarm Robotics Laboratory*. El número de RN varía desde 0 a 3, mientras que el número de UN se fija a 3. Las subfiguras (a) y (c) ilustran cómo evoluciona la media acumulada para la conectividad y el *throughput* en función del número de RN. A la derecha, las subfiguras (b) y (d) presentan los valores instantáneo (líneas continuas) y media acumulada (líneas discontinuas) para las mismas métricas, considerando 0 y 3 RN únicamente.

del sistema sea menor que en condiciones simuladas. Tales restricciones son especialmente relevantes en escenarios que consideran un número mayor de nodos o entornos más densos. Este comportamiento se ilustra en las Figuras 4.26 y 4.27, las cuales muestran el rendimiento del sistema DRNS en comparación con la solución RAND para los patrones de movilidad que se vienen considerando hasta el momento, RWP y RPGM. Especialmente en escenarios RPGM, es altamente probable que se crucen trayectorias de paso de diferentes robots, lo que hace que se tenga que evitar el posible choque entre ellos provocando las propias deceleraciones, aceleraciones y cambios de trayectoria. Esto causa que la mejora conseguida empleando 3 RN sea

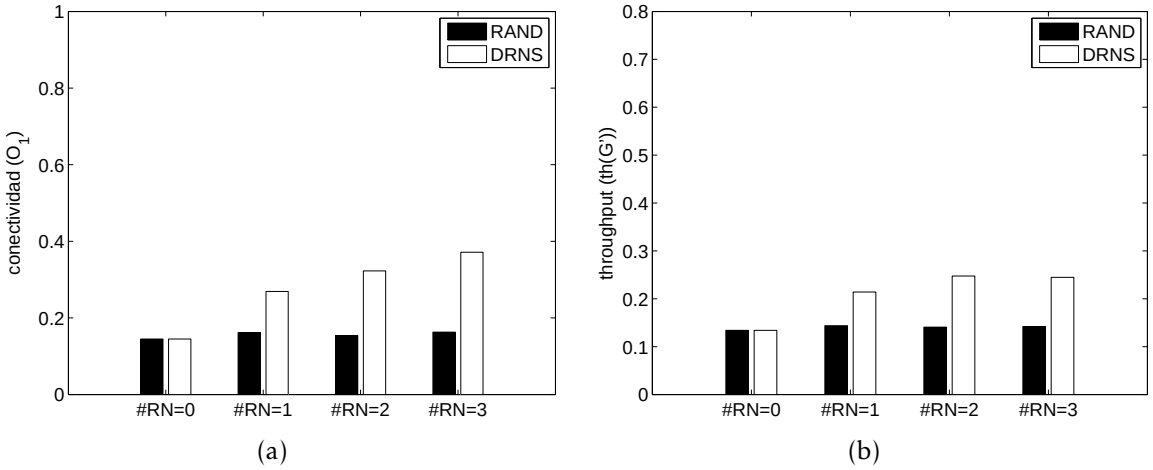


Figura 4.26: Comparativa de rendimiento entre las soluciones DRNS y RAND empleando RWP como patrón de movimiento para los UN en el entorno *IDSIA Swarm Robotics Laboratory*. Las subfiguras (a) y (b) muestran los valores obtenidos para la conectividad y el *throughput*, respectivamente, en función del número de RN empleado cuando se fija el número de UN a 3.

prácticamente igual al resultado obtenido con 2 RN. Es de esperar que este efecto se agudice a medida que aumenta el número de nodos en la red.

Mas allá de los efectos particulares asociados a entornos físicos, ha quedado comprobada la validez y viabilidad del sistema DRNS para su empleo en entornos reales. Además, a parte de las diferencias lógicas entre el rendimiento obtenido por DRNS en su aplicación en entorno reales frente a simulados, hemos de reseñar el prácticamente nulo impacto computacional de DRNS. Este hecho corrobora y aboga por su utilización en escenarios reales incluso con restricciones importantes de tiempo real.

Para apoyar los resultados experimentales obtenidos tras su despliegue en el entorno real y clarificar y consolidar conceptos de implementación y funcionamiento, se invita al lector a visualizar el vídeo correspondiente a un determinado experimento referenciado en [192].

4.7. Conclusiones del capítulo

El problema del posicionamiento de nodos *relay* continúa siendo un reto para la ingeniería a fecha de hoy. Aunque existen varios enfoques y soluciones a dicho problema, la amplia variedad de objetivos perseguidos, así como los diferentes entornos de uso, han fomentado la propuesta y desarrollo de gran cantidad de

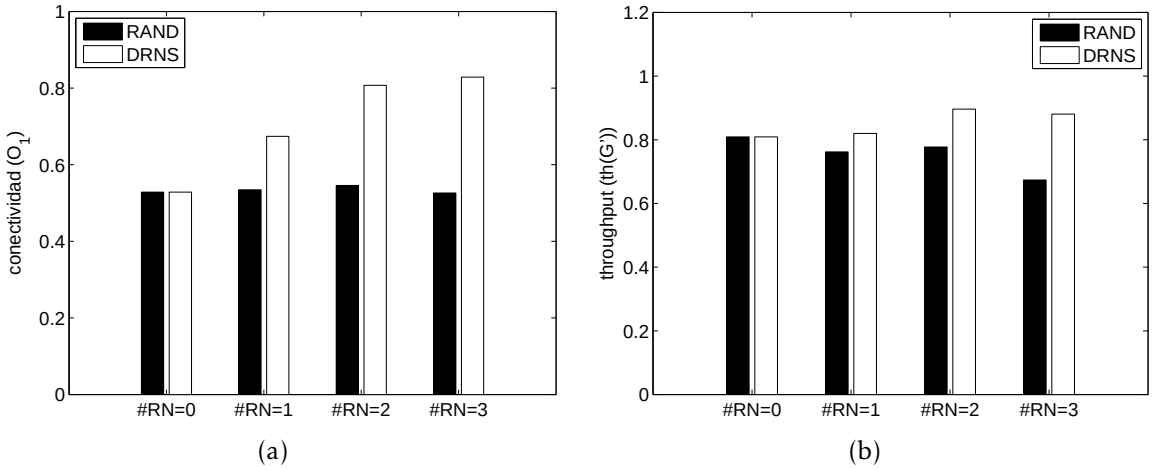


Figura 4.27: Comparativa de rendimiento entre las soluciones DRNS y RAND empleando RPGM como patrón de movimiento para los UN en el entorno *IDSIA Swarm Robotics Laboratory*. Las subfiguras (a) y (b) muestran los valores obtenidos para la conectividad y el *throughput*, respectivamente, en función del número de RN empleado, fijando el número de UN a 3.

heurísticas. A lo largo de todo este capítulo se presenta una solución propia al problema centrado en entornos dinámicos, en concreto aplicado a redes MANET. El esquema persigue maximizar la conectividad y el *throughput* de la red a lo largo del tiempo, considerando un número de nodos *relays* limitado y establecido de antemano.

A través de la exposición y propuesta de una formulación formal y coherente del problema, se presenta una heurística multietapa que trata de resolver dos aspectos fundamentales a la hora de abordar cualquier problema de posicionamiento de nodos *relays* enmarcado dentro de entornos dinámicos: (*i*) dónde han de ubicarse dichos nodos y (*ii*) cómo han de moverse estos hacia esas posiciones. Para solventar ambas cuestiones se divide el problema de manera lógica en varios módulos para afrontar, por un lado, la optimización de la posiciones de los nodos y, por otro, su movimiento hacia ellas. Esta división modular de la funcionalidad añade flexibilidad y versatilidad al sistema de posicionamiento.

Tomando como base una solución ya existente en la literatura, a través de nuestro esquema solventamos serias e importantes deficiencias de la solución de partida en relación a los dos aspectos relevantes descritos anteriormente. A través de una extensa experimentación se demuestra la viabilidad del sistema planteado no solo en entornos simulados sino también en entornos reales MANET en donde los nodos son robots.

Desde el punto de vista de la seguridad, la solución propuesta a lo largo del presente capítulo se presenta como una solución factible para contrarrestar los efectos

que producen determinados comportamientos maliciosos sobre el rendimiento de la red. Actuando sobre la maximización de métricas de rendimiento de la red se pretende conseguir sistemas más globales de respuesta/tolerancia que contemplen aquellas amenazas o ataques cuyo impacto en la red, directo o indirecto, afecte a dichas métricas en su deseo de interrumpir, principalmente, la disponibilidad de la red. A modo de ejemplo, se contrarrestan los efectos producidos en la conectividad y *throughput* de la red que conlleva la actuación de atacantes de tipo *blackhole*, siendo este esquema perfectamente aplicable para la lucha contra ataques como el de *sinkhole* o *wormhole* o frente a comportamientos *selfish*.

Publicaciones relacionadas

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

- **R. Magán-Carrión**, J. Camacho, P. García-Teodoro, E. F. Flushing y Gianni A. Di Caro. "Dynamical Relay Node placement Solution (DRNS) for MANETs," Enviado a *Ad Hoc Networks (Elsevier)*, 39 páginas, 2016.
- **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro, E. F. Flushing and Gianni A. Di Caro. "DRNS: Dynamical Relay Node placement Solution," Aceptado en *Demonstration in Advances on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, junio 2016.
- **R. Magán-Carrión**, R.A. Rodríquez-Gómez, J. Camacho y P. García-Teodoro. "Optimal Relay Placement in Multi-hop Wireless," Aceptado en *Ad Hoc Networks (Elsevier)*, 34 páginas, marzo 2016.
- **R. Magán-Carrión**, J. Camacho, P. García-Teodoro, E. F. Flushing y Gianni A. Di Caro. "Dynamical Relay Node placement Solution in MANETs," *Demonstration in 3rd International Black Sea Conference on Communications and Networking*, mayo 2015.
- **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Multiagent Self-healing System against Security Incidents in MANETs," *Highlights of Practical Applications of Heterogeneous Multi-Agent Systems*, vol. 430. pp. 321–332, junio 2014.
- **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents: Limitations and Improvements," *XI Jornadas de Ingeniería Telemática (JITEL 2013)*, pp. 445–452, octubre 2013.
- **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents: A Practical Vision,"

Demonstration in Advances on Practical Applications of Agents and Multi-Agent Systems, vol. 7879, pp. 308–311, mayo 2013.

- **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents," *Advances on Practical Applications of Agents and Multi-Agent Systems*, vol. 7879, pp. 182–191, mayo 2013.

Parte III

Integración de Soluciones de Seguridad

Capítulo 5

NETA *framework*: simulación y evaluación de ataques en redes

Contenido

Como ya se ha puesto de manifiesto a lo largo de la presente tesis, la seguridad se está convirtiendo en uno de los principales problemas a la hora de desarrollar nuevas tecnologías y servicios en redes de telecomunicaciones. Además, las técnicas empleadas en el desarrollo de ataques a la seguridad evolucionan de manera constante y rápida hacia nuevos objetivos [193][194], dificultando enormemente el desarrollo de mecanismos de defensa.

En este contexto, se han llevado a cabo numerosos esfuerzos por parte de la comunidad investigadora en el desarrollo de nuevas técnicas de defensa en la lucha frente 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.

Como resultado de esta metodología, son muchos los investigadores que contribuyen al avance en el campo con el desarrollo e implementación de sus propios

ataques. Sin embargo, no existe un marco común que en cierto modo regule estas contribuciones de manera que fuese posible la comparación objetiva de las soluciones de defensa propuestas frente a dichos ataques en igualdad de condiciones.

Por tanto, resulta deseable la existencia de un *framework* único que posibilite el desarrollo, implementación y evaluación de ataques, así como de sus respectivas soluciones de defensa. Dicho sistema permitiría combinar la ejecución de uno o varios ataques, de forma similar a como lo haría un *hacker*. A su vez, este *framework* posibilitaría el análisis del impacto de los ataques en múltiples tecnologías, protocolos y escenarios.

Motivado por las necesidades anteriores, surge NETA (*NETwork Attacks*)[1], un *framework* de ataques basado en OMNeT++ (*Objective Modular Network Test-bed in C++*) que pretende proporcionar un marco base de referencia con el que unificar el desarrollo, simulación y evaluación de ataques. NETA se diseña y crea para ser escalable y extensible, a la vez que ofrece un alto grado de versatilidad no solo para el desarrollo de nuevos ataques, sino para la integración de soluciones defensivas. Principalmente, trata de minimizar el esfuerzo realizado durante el proceso de creación de ataques que luego son utilizados para probar y evaluar distintas soluciones de seguridad. Se trata así de una herramienta muy útil para la comunidad investigadora, ya que su uso está pensado para ahorrar tiempo y esfuerzo en el desarrollo y validación final del esquema o solución de seguridad planteada.

El resto del capítulo se organiza de la siguiente forma. La Sección 5.1 proporciona un análisis del estado del arte, describiéndose distintos simuladores de redes, así como otras propuestas de seguridad similares a la presentada en este trabajo. Los principios de diseño y la arquitectura general de NETA se presentan en la Sección 5.2. A lo largo de la Sección 5.3 se describen los ataques implementados incluidos dentro del *framework* como ejemplos de uso. La Sección 5.4 detalla los escenarios de estudio, así como el entorno de experimentación, los resultados obtenidos y su evaluación. Finalmente, la Sección 5.5 expone las conclusiones y líneas de trabajo futuro en este campo.

5.1. Herramientas de simulación de redes y ataques

La simulación se usa generalmente con la intención de analizar y validar protocolos y sistemas complejos, ofreciendo de esta forma un buen compromiso entre coste y complejidad [195]. Sin embargo, la elección del mejor simulador no es una tarea

[1]NETA se desarrolla en el seno del grupo de investigación *Network Engineering & Security Group* y está disponible para su descarga en http://nesg.ugr.es/neta.

sencilla, pues requiere de un estudio previo que considere las distintas ventajas y desventajas de los mismos.

Según las referencias [196] y [197], los simuladores más utilizados en el campo de las comunicaciones son: (*i*) OPNET (*Optimized Network Engineering Tools*), (*ii*) NS-2 (*Network Simulator 2*) y (*iii*) OMNeT++. Todos ellos son simuladores de eventos discretos para redes de comunicaciones heterogéneas. Es destacable la capacidad de OPNET en la ejecución y gestión concurrente de distintos escenarios, así como la gran variedad de protocolos que ofrece NS-2. Sin embargo, OMNeT++ se está convirtiendo en la actualidad en uno de los simuladores más empleados, principalmente debido a la amplia variedad de *frameworks* (INET, MIXIM, etc.) que comprende, a su gran flexibilidad y a la inclusión de una interfaz gráfica fácil de usar, entre otras ventajas.

En cuanto al diseño y simulación de ataques, los autores generalmente implementan ataques específicos cuyo propósito es evaluar sus propias propuestas de seguridad, rendimiento de protocolos, etc. [198]. Al ser un desarrollo con carácter privado y específico, es difícil que distintas soluciones de defensa puedan compararse entre sí ya que su evaluación se hace en base a diferentes escenarios e implementaciones de un mismo ataque. Este hecho hace que dichas comparativas sean poco precisas y fiables, puesto que no se puede garantizar a priori que distintas implementaciones de un mismo ataque produzcan los mismos efectos.

Por su parte, los autores del trabajo referenciado en [199] proporcionan un *framework* basado en OMNeT++ para simular patrones de tráfico y ataques DoS sobre redes IP (*Internet Protocol*). Sin embargo, solo implementan un tipo específico de ataque y su propuesta no es extensible de cara al desarrollo de otros. En el trabajo descrito en [200] se desarrolla un *framework* de simulación de ataques aplicado a WSN. Dicho trabajo expone 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. Los autores del trabajo [201] presentan un nuevo simulador llamado NeSSi (*Network Security Simulator*), NEeSSi2 en su versión más reciente. NeSSi2 es una herramienta principalmente orientada a la integración y evaluación de soluciones de detección frente a amenazas de seguridad a nivel de aplicación. De hecho, contiene diversos ataques DDoS (*Distributed DoS*) así como virus de tipo *worm* ya implementados. Los autores no solo defienden la versatilidad de su herramienta para la implementación e integración de soluciones de detección, sino su viabilidad para el desarrollo de nuevos ataques a nivel de aplicación.

De lo anterior se deduce la necesidad de disponer de un *framework* de desarrollo, implementación y test de ataques más general, extensible y versátil que los hasta ahora disponibles. Con esta idea surge y se desarrolla NETA, cuyo funcionamiento y arquitectura se describen en la siguiente sección.

5.2. NETA: NETwork Attacks

NETA se concibe como un *framework* separado dentro del simulador OMNeT++. A su vez, se basa en el *framework* INET (*INET Framework*)[2]. De esta forma, por un lado, se pretende extender su uso entre la comunidad investigadora al ser OMNeT++ una de las herramientas de simulación más usadas en el ámbito de la simulación de redes de comunicación. Por otro lado, dotamos a la herramienta de toda la funcionalidad que oferta INET en la simulación de escenarios realistas y complejos de redes de comunicaciones.

Antes de sumergirnos en la descripción y funcionalidad de NETA, es necesario describir aquí las dos herramientas utilizadas para la correcta ejecución de nuestro *framework*.

5.2.1. Introducción a OMNeT++

OMNeT++ provee las herramientas y el entorno necesarios para llevar a cabo simulaciones de diferente índole. Posee una arquitectura modular y se basa en el empleo de eventos discretos. Entre otras ventajas, ofrece una estructura y definición versátiles que agranda su ámbito de uso a contextos y problemas diversos relacionados, entre otros, con el modelado de redes de comunicaciones (cableadas o no), definición y creación de protocolos, etc.

OMNeT++ se basa en el empleo de *módulos* de forma jerárquica. Esto añade una gran flexibilidad a la hora de implementar casi cualquier funcionalidad. Así, existen los denominados *módulos simples*, que integran una determinada funcionalidad y se implementan a través del lenguaje C++. A su vez, los *módulos compuestos* agrupan funcionalidades simples. A modo de ejemplo, pensemos en un nodo de una red MANET que necesita comunicarse con sus homólogos, que además posee una cierta movilidad, dispone de una batería que le provee de relativa autonomía, y muchos más aspectos que, juntos, posibilitan su funcionamiento. Claramente, dicho nodo se corresponde con un módulo compuesto, siendo módulos simples los demás. Llevando el ejemplo al *framework* INET, dicho nodo se corresponde con el nodo `AdhocHost` que, entre otros, posee un módulo encargado de moverlo (por ejemplo, `RandomWPMobility`) y otro para el modelado de la batería (`InetSimpleBattery`), ambos módulos simples que modelan e implementan funcionalidades concretas.

Otro aspecto importante en OMNeT++ es cómo se comunican entre sí los diferentes módulos. Esto se lleva a cabo a través de la utilización de puertas (*gates*) sobre

[2]INET ofrece un conjunto de herramientas para la simulación de redes cableadas, inalámbricas móviles o estáticas pensado para integrarse en OMNeT++. INET está disponible para su descarga en https://inet.omnetpp.org/

las que se enviarán (*out gates*) o recibirán (*in gates*) ciertos mensajes configurables. También existen puertas bidireccionales (*in-out gates*) que pueden enviar y recibir mensajes. Este paso de mensajes es muy útil para, por ejemplo, modelar el acceso a servicios entre capas de un determinado protocolo.

Hasta aquí, se puede pensar que con esta filosofía prácticamente se podría modelar cualquier sistema o funcionalidad. Efectivamente, para ello está pensado. Sin embargo, es normal que aspectos como la flexibilidad y la escalabilidad añadan dificultad a la hora de su uso. OMNeT++ solventa esta problemática separando, por un lado, el modelado de la simulación y, por otro, su configuración. Para la parte de modelado, en donde definimos qué módulos intervendrán así como sus comunicaciones y parámetros asociados, se define un lenguaje especial denominado NED (*Network Description*). Así, cada módulo, tanto simple como compuesto, posee su correspondiente archivo `.ned` que lo describe a alto nivel. Se pueden ir agrupando módulos de manera jerárquica hasta que en el nivel más alto encontramos el modelo del entorno que queremos simular, llamado *network*. A este nivel disponemos de elementos como los nodos de la red y módulos que configuran aspectos de la misma (por ejemplo, la gestión de direcciones de red). Una vez establecidos, descritos y ensamblados los elementos que componen el entorno pretendido, resta configurar ciertos aspectos propios de la simulación en sí. Para ello se utilizan ficheros `.ini` en donde se configuran parámetros como el tiempo de la simulación, número de repeticiones de las diferentes simulaciones, número de nodos, configuración de comunicaciones, etc.

INET (*INET Framework*)

Mencionado anteriormente, OMNeT++ solo provee la vía y herramientas necesarias para la simulación de modelos y escenarios complejos. Son, así, los diferentes *frameworks* existentes y diseñados según la filosofía de OMNeT++ los que aportan la funcionalidad para la simulación.

Para establecer y simular modelos que representen redes y escenarios de comunicaciones surge INET (*INET Framework*). INET implementa multitud de funcionalidades, desde protocolos de comunicaciones específicos hasta aspectos como el modelado de la batería de un nodo dentro de una red ad hoc. Otros ejemplos son la implementación del protocolo de encaminamiento AODV (`AODVUU`) para redes MANET o la aplicación *ping* (`PingApp`).

Juntos, OMNeT++ e INET, proveen el entorno y funcionalidad necesaria para la integración y desarrollo de NETA como un *framework* versátil, funcional y escalable en el contexto de la seguridad en redes de comunicaciones.

5.2.2. Principios de diseño y funcionamiento

NETA se basa en la misma idea que OMNeT++, esto es, módulos que se comunican entre sí mediante el paso de mensajes.

Principalmente, se fundamenta en el diseño e implementación de nuevos nodos que puedan ejecutar ataques, los *nodos atacantes*. Para hacerlo posible, los ataques se controlan mediante los denominados *controladores de ataque*. Dichos controladores gestionan uno o varios módulos de NETA mediante el envío de los denominados *mensajes de control*. Estos mensajes viajan desde los controladores de ataque hacia módulos específicos, convenientemente modificados para proporcionar el comportamiento del ataque. Dichos módulos adquieren el nombre de *módulos hackeados*. Para habilitar comportamientos maliciosos, los módulos *hackeados* heredan o replican el código de sus equivalentes en el *framework* INET. Posteriormente, se modifican de manera conveniente para, primero, obedecer las órdenes indicadas por los controladores y, segundo, cambiar el comportamiento normal del módulo por el del correspondiente ataque.

Los principios de diseño del presente *framework* siguen dos reglas principales:

Regla 1. *Cualquier* framework *base que sea 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 modificaciones en el comportamiento de los módulos que pasarán a ser módulos *hackeados*. Con esta regla se pretende minimizar la cuantía e impacto de dichas modificaciones tanto como sea posible.

Así, la creación de un nodo atacante puede resumirse en los siguientes pasos: (*i*) añadir al archivo `.ned` correspondiente los controladores relacionados con los ataques a ejecutar, (*ii*) crear los mensajes de control asociados y (*iii*) sustituir los módulos necesarios por parte de los controladores de ataque por los módulos *hackeados* correspondientes.

La Figura 5.1 muestra las diferencias existentes entre un nodo normal y un nodo atacante. El primero se compone de módulos simples y compuestos que se comunican entre sí mediante el paso de mensajes que podríamos denominar como normales. En cuanto al segundo, el nodo atacante, se compone de igual número de módulos, a los que se añaden los correspondientes controladores. Además, algunos de los módulos originales pueden ser reemplazados por módulos *hackeados* que permiten la

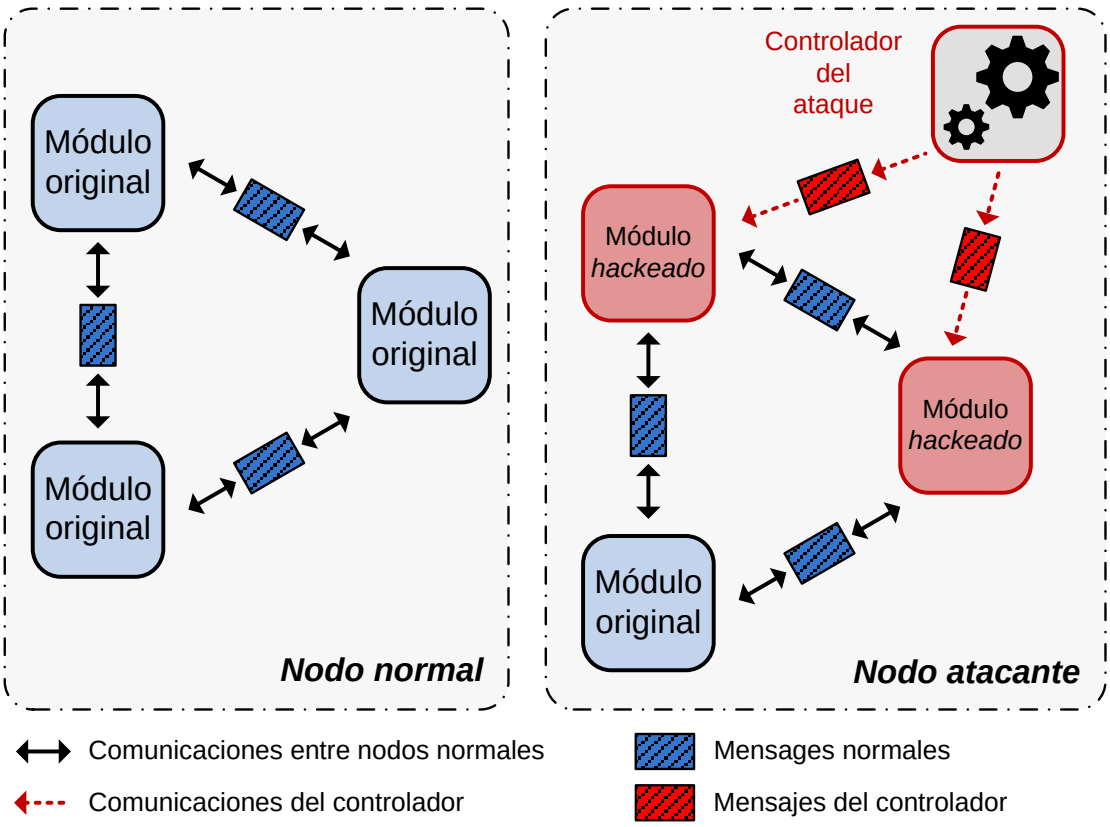


Figura 5.1: Esquema comparativo entre un nodo original y el correspondiente nodo atacante en NETA.

ejecución del ataque en cuanto así lo ordene su controlador. Como se aprecia en la figura, es el controlador el que se comunica directamente con los módulos *hackeados*.

5.2.3. Arquitectura de NETA

Como ya se introdujo en la sección anterior, son tres los componentes implicados tanto en el desarrollo e implementación de un ataque como en su ejecución. Nos referimos a los *controladores de ataque*, los *mensajes de control* y, por último, los *módulos hackeados*. A continuación, se describe en mayor detalle cada uno de ellos.

Controladores de ataque

Estos módulos, como su propio nombre indica, se encargan del control, gestión y 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`: instante de tiempo de simulación en el que el ataque se hace efectivo, es decir, cuándo comienza a ejecutarse.
- `endTime`: instante de tiempo de simulación en el que cesa el ataque.
- `customParameters`: diferentes parámetros de configuración que dependen de las funcionalidades específicas del ataque.

A continuación se resumen los principales pasos seguidos por un controlador específico a la hora de llevar a cabo un ataque en concreto:

1. Obtiene los diferentes módulos *hackeados* involucrados en la ejecución del ataque.
2. Activa aquellos módulos *hackeados* en el nodo atacante enviando mensajes de activación que también pueden contener información de configuración.
3. Desactiva los módulos *hackeados* en el nodo atacante enviando un mensaje de desactivación.

Mensajes de control

Son aquellos mensajes que se envían desde los controladores a los módulos *hackeados* involucrados en la ejecución del ataque. A través de estos, se transmite la información necesaria para la activación y desactivación de los ataques. Además, también pueden contener 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 regla 2 de nuestros principios de diseño: "Minimizar la modificación en el código original de los módulos *hackeados*".

Módulos *hackeados*

Conforman el núcleo del ataque y recae sobre ellos responsabilidad de llevarlo a cabo. Son módulos originales cuyo comportamiento normal cambia de cara a ejecutar un determinado ataque. Por ejemplo, un ataque de descarte paquetes (*dropping*) requiere usualmente la modificación del módulo encargado del reenvío de paquetes en la capa de red o IP. Por tanto, la implementación de dicho ataque implica la

modificación del módulo IPv4 original del *framework* INET en NETA. A partir de entonces será un módulo *hackeado* puesto a las órdenes de su controlador en cuestión.

Es importante resaltar que existe un único módulo *hackeado* por cada módulo modificado, en lugar de un módulo *hackeado* por cada implementación de un ataque. Esto es, si dos ataques diferentes necesitan modificar el mismo módulo, sólo existirá un módulo *hackeado* que implementará los dos ataques. Por ejemplo, y como se mostrará en la sección siguiente, tanto el ataque *IP dropping* como *IP delay* están relacionados con el módulo IPv4. Sin embargo, es solamente un módulo IPv4 el *hackeado*. Este diseño tiene como objetivo mejorar la flexibilidad del *framework*, permitiendo así la ejecución de más de un ataque simultáneamente. Por ejemplo, los ataques de *IP dropping* e *IP delay* pueden lanzarse en un mismo nodo sin más que incluir sus correspondientes controladores de ataque.

5.3. Ataques implementados

En esta sección se exponen los ataques implementados como prueba de concepto para la validación de NETA. Para cada uno de los ataques desarrollados describiremos: (*i*) cómo se comporta y (*ii*) los principales parámetros que modifican su comportamiento.

5.3.1. Ataque *IP dropping*

El comportamiento usual de un ataque *IP dropping* consiste en el descarte intencionado de los paquetes de datos recibidos de acuerdo a una determinada probabilidad. Ataques de este tipo afectan gravemente al funcionamiento y rendimiento esperados en la red, sobre todo en entornos inalámbricos *multi-hop*, ya que los nodos que lo realizan tratan de interrumpir el proceso normal de reenvío de paquetes [5]. Según la aplicación afectada, el resultado puede ser una ralentización en el envío y recepción de mensajes debido a numerosas retransmisiones, un excesivo consumo de energía en los nodos, etc. El único parámetro que modifica su comportamiento es:

- `droppingAttackProbability`: 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 normal, es decir, sin descartar paquetes.

5.3.2. Ataque *IP delay*

Este ataque se caracteriza por el retardo que introducen los nodos maliciosos durante el proceso de reenvío de los paquetes de datos IP. Principalmente, afecta a algunos parámetros de QoS de la red, como son el E2ED (*End-to-End Delay*) y el *jitter*. Dependiendo del objetivo o aplicación final de la red, este ataque puede afectar bastante al rendimiento. El ataque *IP delay* puede variar su comportamiento mediante la modificación de los siguientes parámetros:

- `delayAttackProbability`: 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, es decir, no se aplica ningún retardo extra.
- `delayAttackValue`: 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, es decir, es modificado cada vez que se accede a él. Por defecto, sigue una distribución normal con media 1 segundo y desviación típica 0,1 segundos.

5.3.3. Ataque *AODV sinkhole*

Aquellos nodos que actúan como nodos *sinkhole* envían información falsa de encaminamiento proclamando la posesión de rutas óptimas hacia el/los destinos solicitados. Para ello modifican malintencionadamente los paquetes de respuesta (`RREP`) anunciando la tenencia de dichas rutas. Este comportamiento provoca que otros nodos encaminen sus paquetes de datos a través de los nodos atacantes.

Como en los anteriores ataques, en este caso se puede alterar el funcionamiento de un nodo mediante los siguientes parámetros específicos:

- `sinkholeAttackProbability`: probabilidad de responder a un mensaje de solicitud (`RREQ`) con una respuesta falsa (`RREP`), definida 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 nodo *sinkhole* solo envía falsos `RREP` a solicitudes para las que el atacante tenga una ruta válida, es decir, rutas existentes en su tabla de encaminamiento. En caso contrario (valor *false*), el nodo envía `RREP` falsos para cualquier `RREQ` que le llegue, incluso si no tiene una ruta válida.
- `seqnoAdded`: falso número de secuencia generado por el nodo atacante. Dicho valor 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`: falso número 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.

5.4. Resultados experimentales

Con el fin de validar no solo la viabilidad de la herramienta NETA sino también la correcta ejecución de los mencionados ataques, se expone a continuación el escenario de simulación propuesto así como los diferentes experimentos llevados a cabo para tal fin.

5.4.1. Descripción del entorno de simulación

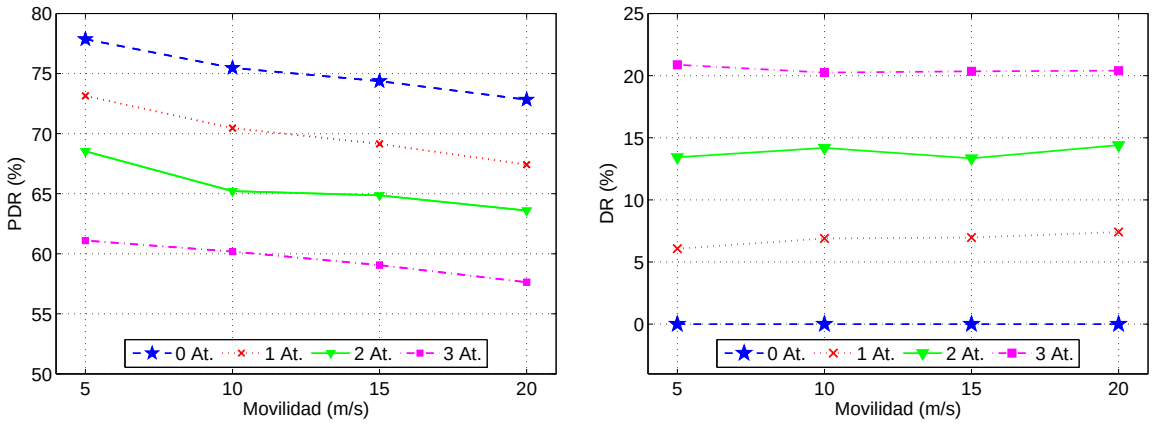
Como caso de estudio, el entorno de simulación está basado en despliegues de redes MANET. Todos los escenarios que se proponen para la ejecución de los experimentos tienen en común los aspectos que se describen a continuación.

El área de simulación se restringe a un cuadrado de 1000 m × 1000 m. Cada nodo posee un radio de cobertura de 250 m. El tiempo de simulación se fija a 300 s y los resultados obtenidos se derivan promediando (con distintas semillas) 50 repeticiones para cada simulación.

En relación a los protocolos para la capa MAC y de encaminamiento, se han elegido IEEE (*Institute of Electrical and Electronics Engineers*) 802.11g y AODV respectivamente, así como el mecanismo RTS (*Request To Send*)/CTS (*Clear To Send*) 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.

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* esta fijada al 100 %. Dicha tasa indica la probabilidad de que se realice el ataque.

El número de flujos con tráfico a nivel de aplicación está fijado a 21. Cada flujo consiste en tráfico UDP (*User Datagram Protocol*) con una tasa de envío constante (CBR (*Constant Bit Rate*)) de 4 paquetes/s, teniendo cada paquete un *payload* de 512 bytes. Para cada flujo, la dirección destino es elegida de forma aleatoria entre todos los nodos legítimos, manteniéndose el mismo destino durante todo el tiempo de la



(a) (b)

Figura 5.2: Evaluación del rendimiento del ataque *IP dropping*. Los valores obtenidos para las métricas PDR y DR en función de la movilidad de los nodos y el número de atacantes se ilustran en las subfiguras (a) y (b), respectivamente.

simulación. Los flujos comienzan de forma aleatoria entre 0,5 y 1,5 s y terminan entre 290 y 295 s.

Se utiliza el patrón establecido por RWP para simular el movimiento de los nodos. La velocidad mínima se fija a 1 m/s y la velocidad máxima varía entre 5 y 20 m/s, con un tiempo de pausa de 15 s. Es decir, una vez que el nodo alcanza el destino deseado, espera inmóvil durante el tiempo de pausa antes de elegir un nuevo destino. Este proceso se vuelve a repetir hasta el fin de la simulación.

5.4.2. Evaluación del ataque *IP dropping*

Para evaluar el funcionamiento del ataque *IP 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.

El efecto del ataque sobre las métricas de rendimiento sugeridas puede observarse en la Figura 5.2. Como cabría esperar, si el número de atacantes crece el PDR se ve deteriorado, mientras que el DR aumenta. Además, puede verse cómo el PDR decrece conforme aumenta la movilidad, mientras que el DR permanece prácticamente

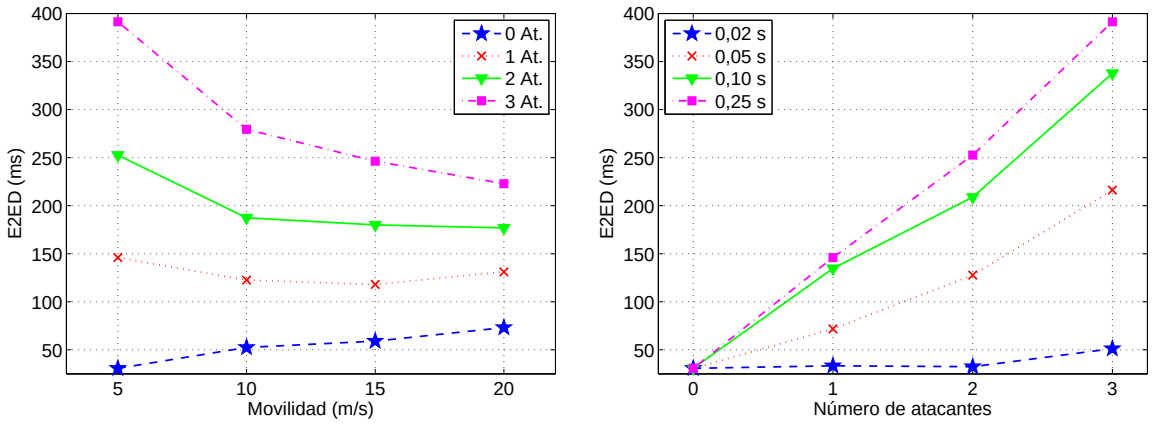


Figura 5.3: Evolución E2ED en presencia del ataque *IP delay* para, (a) distintas velocidades y número de atacantes, aplicando un *delay* de 0,25 segundos, y (b) aplicando distintos *delays*, con una velocidad fija de 5 *m/s*.

estable. Esto es debido a que incrementos en la velocidad de los nodos implican una mayor pérdida de paquetes principalmente por las colisiones y los errores del canal, mientras que el número de paquetes descartados como consecuencia del ataque permanece constante. De esta manera se corrobora, por un lado, la eficiencia del ataque y, por otro, su correcta implementación mediante NETA.

5.4.3. Evaluación del ataque *IP delay*

La métrica escogida para evaluar el rendimiento del ataque *IP delay* es la siguiente básica:

- E2ED (*End-to-End Delay*): tiempo medio empleado por un paquete de datos desde su transmisión hasta que alcanza el destino, incluyendo todos los posibles retrasos debidos a descubrimiento de rutas, permanencia en colas, tiempos de propagación, etc. Se calcula como el promedio de los E2ED de cada paquete en cada flujo, extrayéndose de esta forma el E2ED medio para toda la red.

Las pruebas realizadas evalúan el impacto del aumento del número de atacantes sobre el parámetro E2ED (Figura 5.3(a)), así como el incremento en el retardo aplicado por el atacante (Figura 5.3(b)). En el primer caso se añade un retardo de 0,25 s, correspondiente al tiempo entre llegadas de la aplicación CBR. Como puede verse en la figura, el retardo 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.

Al igual que en el ataque *IP dropping*, queda corroborado el correcto funcionamiento del ataque *IP delay* dentro del marco de NETA.

5.4.4. Evaluación del ataque de sinkhole

Para caracterizar el rendimiento de los nodos *sinkhole* se define la siguiente métrica:

- AR (*Attraction Rate*): 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. Dicho parámetro se calcula como:

$$AR = \frac{\frac{1}{N_S}\sum_{i=1}^{N_S} pkt_i - \frac{1}{N_L}\sum_{j=1}^{N_L} pkt_j}{\frac{1}{N_S}\sum_{i=1}^{N_S} pkt_i} \cdot 100 \tag{5.1}$$

  siendo N_S y N_L el número de nodos *sinkhole* y legítimos, respectivamente, y pkt_i el número total de paquetes recibidos por el nodo i.

La Figura 5.4 muestra cómo los nodos *sinkhole* atraen un tráfico superior al resto de nodos. Además, puede observarse que el parámetro AR decrece a medida que aumenta el número de atacantes. Esto es debido a que los atacantes compiten entre sí en la atracción del tráfico, dando como resultado un menor AR. Sin embargo, el número total de paquetes atraídos por todos los nodos *sinkhole* crece con el número de atacantes, como cabría esperar.

5.5. Conclusiones del capítulo

A lo largo de este capítulo se presenta y evalúa NETA, un *framework* para la simulación de ataques en redes de comunicación desarrollado en base al *framework* INET dentro del entorno de simulación OMNeT++.

NETA presenta tres componentes principales: *controladores de ataque*, que gestionan la ejecución de los ataques; *módulos hackeados*, que implementan el comportamiento del ataque; y *mensajes de control*, encargados de transmitir la información de

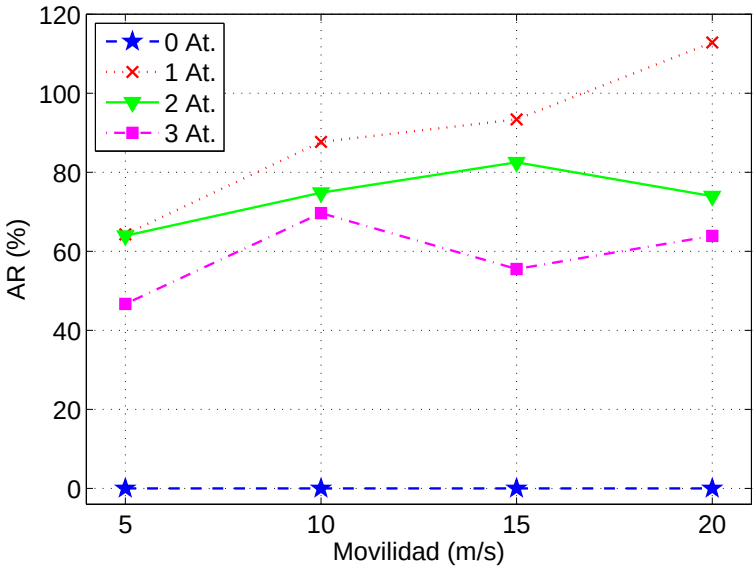


Figura 5.4: Evolución del parámetro AR para distintas velocidades, retardos y número de atacantes.

activación/desactivación, así como información de configuración desde los controladores de ataque a los módulos *hackeados*. Asimismo, y como prueba de concepto, se han implementado tres ataques: *IP dropping*, *IP delay* y *AODV sinkhole*.

Como caso de estudio se han considerado escenarios de aplicación realistas, analizando una serie de despliegues MANET. Como puede comprobarse, 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.

Sin embargo, esta primera versión de NETA es de alcance aún limitado. Por un lado, se prevé el desarrollo e implementación de nuevos ataques con una mayor complejidad junto con el desarrollo de diferentes métricas de rendimiento. También se contempla la implementación e integración de esquemas de seguridad que proporcionen una solución global de seguridad en este campo.

Tomando como base ello, el siguiente capítulo aborda la mejora de NETA en dos aspectos. El primero, la implementación de soluciones de seguridad concretas para algunos de los ataques en este capítulo ya citados. En segundo lugar, la integración y operación conjunta de líneas de defensa diferentes que permitan, en su conjunto, dotar al sistema global de capacidades adicionales de supervivencia.

Publicaciones relacionadas

Las principales contribuciones de este capítulo en forma de publicaciones científicas se listan a continuación:

- 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, sept. 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, octubre 2013.

Capítulo 6

Integración de soluciones de seguridad

Contenido

En anteriores capítulos se han diseñado y desarrollado soluciones de respuesta o reacción ante ataques a la seguridad en redes ad hoc. Aunque estas soluciones son parte imprescindible de cualquier sistema robusto y resistente ante amenazas a la seguridad, es necesaria la intervención de esquemas adicionales que las complementen de cara a proveer soluciones integrales y globales de seguridad. Para abordar esta cuestión se desarrolla e implementa una arquitectura que hace posible dicha integración y que se construye dentro del marco de NETA.

Obviamente, para llevar a cabo la integración de diferentes esquemas de seguridad es necesario contar con sus correspondientes implementaciones. En este sentido, a lo largo del presente capítulo se extiende NETA en dos aspectos. Por un lado, mediante la implementación de nuevos módulos relativos a esquemas de seguridad correspondientes a distintas líneas de defensa, como son la detección o la respuesta. Por otro lado, su ampliación mediante la ya mencionada herramienta de integración. De esta manera, NETA (y el marco de simulación que proporciona OMNeT++ e INET) permitirá simular conjuntamente: (*i*) el despliegue y ejecución de un determinado entorno de red, (*ii*) la realización de ataques y su control durante la simulación, (*iii*) la detección de comportamientos maliciosos a través de los sistemas de detección implementados, (*iv*) la generación de notificaciones ante dichos eventos, y (*v*) la

puesta en marcha del correspondiente sistema de respuesta para mitigar los efectos del ataque sobre el rendimiento de la red.

En lo que sigue, este capítulo se divide en tres secciones. A lo largo de la Sección 6.1 describiremos el desarrollo e implementación de un esquema específico de respuesta dentro del marco de NETA. Dicha solución se corresponde con una versión preliminar a la planteada en el Capítulo 4. En la Sección 6.2 se describe y despliega el *framework* de integración desarrollado que permite la obtención de soluciones de seguridad globales a través de la operación conjunta de diferentes esquemas de defensa. Para finalizar, en la Sección 6.3 se exponen las conclusiones del capítulo.

6.1. Despliegue de soluciones de respuesta/tolerancia

Como se ha comentado con anterioridad, el objetivo de la presente sección es validar la implementación de esquemas de seguridad dentro del marco de NETA. En particular, se implementará la solución de respuesta/tolerancia desarrollada en el trabajo [177] correspondiente a una versión preliminar de DRNS descrita en el Capítulo 4.

6.1.1. Funcionalidad

Debidamente detallados en la Sección 4.3.1, son varios los requisitos funcionales que se establecen para el sistema DRNS y que son directamente aplicables aquí. Así, se contemplan dos tipos de nodos: los UN y los RN. Dada la naturaleza centralizada de la solución, existirá además un nodo central o CN (*Central Node*) encargado de:

1. Obtener las posiciones de los UN y los RN.
2. Ejecutar la correspondiente rutina DRNS para la obtención de las posiciones optimizadas de los RN.
3. Enviar las nuevas posiciones de los RN para indicar así hacia dónde han de ser movidos.

A partir de los requisitos anteriores, se hace necesaria la implementación de las vías de comunicación y paso de mensajes entre el CN y los nodos implicados. De esta manera, existirán mensajes globales o *broadcast*, dirigidos a todos los nodos, o específicos (*unicast*), dirigidos hacia determinados nodos. Además, el nodo CN ha de ser capaz de obtener la posición de todos y cada uno de los nodos de la red, existiendo así un canal de comunicación de petición y respuesta desde y hacia el CN. De manera

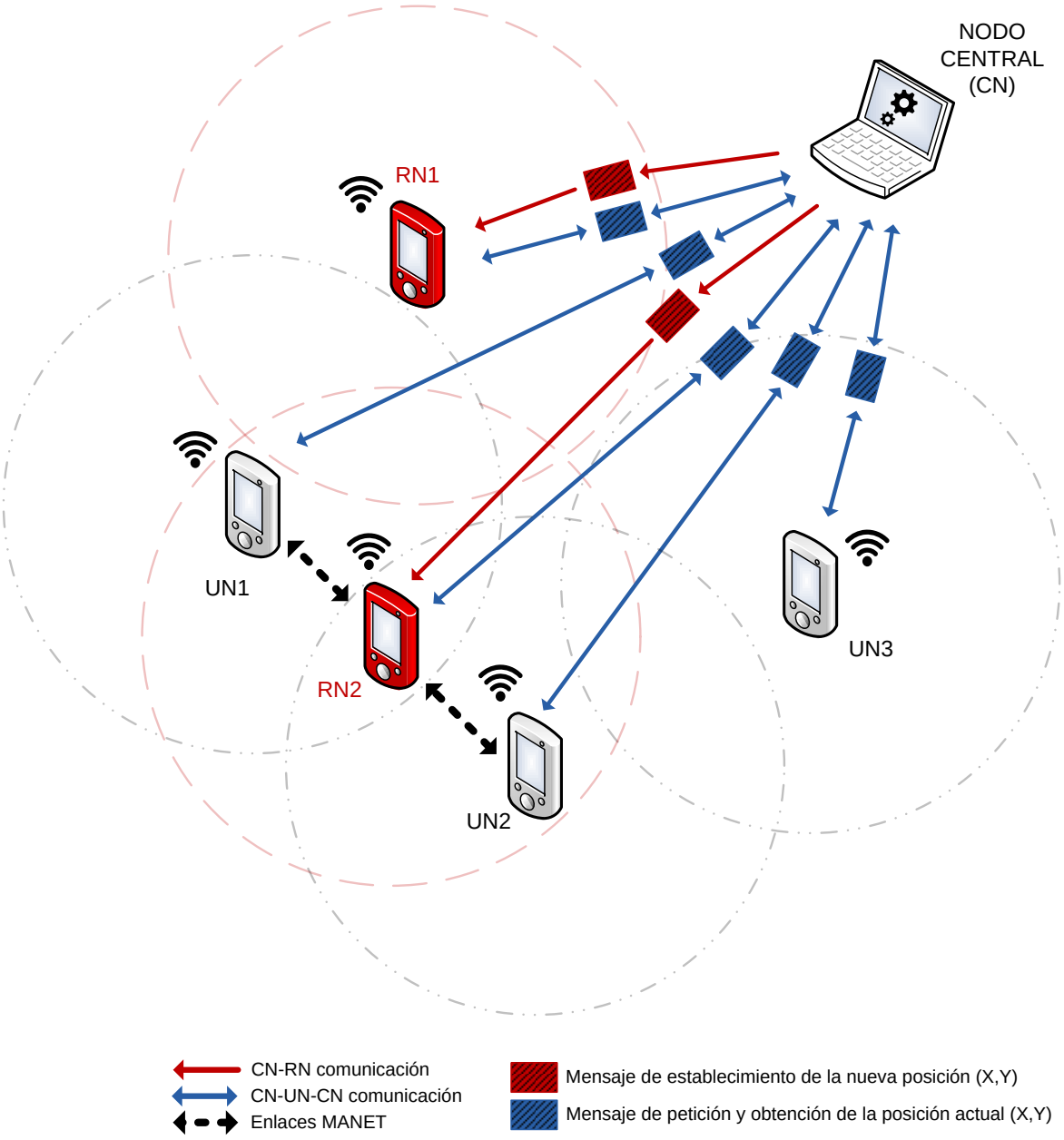


Figura 6.1: Esquema funcional de comunicaciones e intercambios de mensajes para la integración de la solución de respuesta en NETA.

similar, el CN enviará los mensajes adecuados hacia cada RN indicándoles su nueva posición. La Figura 6.1 ilustra el intercambio de información entre los nodos de la red y el CN, así como los canales de comunicaciones implicados.

Puesto que el escenario de uso se refiere a entornos en dos dimensiones, dichos mensajes transportarán información necesaria para, primero, identificar al nodo en

Figura 6.2: Formato de los mensajes de control para la petición y establecimiento de la ubicación de los nodos.

cuestión y, segundo, determinar las coordenadas en el plano (X,Y) correspondientes a su localización actual o aquellas a las que ha de moverse. Adicionalmente, cada mensaje se distinguirá mediante un identificador que determinará su tipo en función de si estamos ante una petición de coordenadas de localización (tipo 1), una respuesta a dicha petición (tipo 2) o se corresponde con el establecimiento de nuevas coordenadas (tipo 3) en el caso de los RN. En total, la longitud de cada mensaje será de 12 bytes repartidos de la siguiente forma: 2 bytes (`ushort`) para el tipo de mensaje, 2 bytes (`ushort`) para el identificador del nodo, 4 bytes (`float`) para la coordenada X del nodo en cuestión y, por último, 4 bytes (`float`) correspondientes a la coordenada Y del mismo nodo. La Figura 6.2 muestra la estructura liviana que poseen los mensajes intercambiados.

Dado el dinamismo característico del entorno sobre el que se centra nuestro sistema, las redes MANET, es necesario establecer cómo han de moverse los nodos de la red. En relación a los UN, estos se moverán de manera independiente a través del área considerada, sin que se establezca control alguno por parte de la entidad centralizada. Por el contrario, los RN se moverán siguiendo patrones de movimiento lineales desde su ubicación actual hacia las nuevas coordenadas (o *target points*) marcadas por la rutina de posicionamiento. A su vez, un parámetro importante es la velocidad asignada a los nodos, siendo necesario establecer una velocidad más elevada para los RN en comparación a aquella asignada a los UN. Esto permite, como ya se discutió en el Capítulo 4, la adaptabilidad del sistema a los continuos cambios en la topología de la red.

Finalmente, el tráfico que se transmite por la red tendrá como origen y destino nodos UN, mientras que los RN se limitarán, como no puede ser de otra forma, a reenviar el tráfico recibido hacia su destino correspondiente (ver Figura 6.1).

6.1.2. Implementación e integración

Tomando en consideración los requisitos funcionales que debe acometer nuestra propuesta para su correcto despliegue y funcionamiento, durante la presente sección

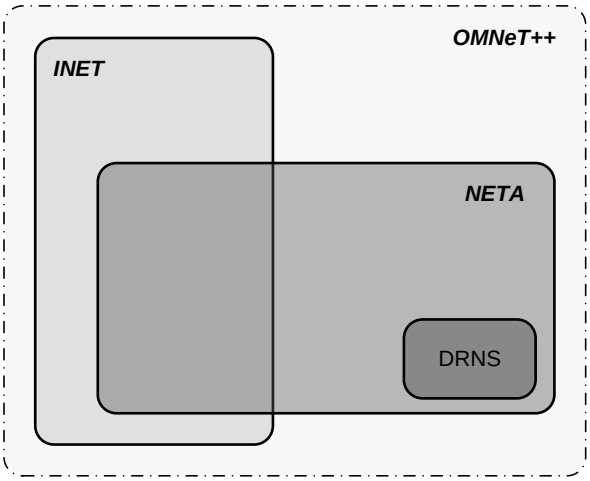


Figura 6.3: Ámbito y ubicación de los elementos necesarios para la ejecución de la propuesta DRNS dentro del marco de NETA.

se detallará cómo se ha llevado a cabo su integración e implementación dentro de NETA.

La herramienta NETA se introdujo y describió en detalle a lo largo del Capítulo 5. A su vez, vimos que NETA depende del *framework* INET y que ambos se ejecutan en el marco de simulación proporcionado por OMNeT++. Así, el proceso de integración, implementación y pruebas de nuestra propuesta dentro de NETA depende en mayor o menor medida de las herramientas sobre las que se apoya este *framework*. En la Figura 6.3 se observa la ubicación de cada una de los elementos necesarios para la ejecución de la propuesta DRNS y su ámbito de uso.

De acuerdo a la discusión anterior, la implementación de los requerimientos funcionales deseados no solo para nuestro sistema, sino para el entorno que le rodea, pueden clasificarse en base a cuál es la herramienta que proporciona dicha funcionalidad. Dos son los aspectos principales a tener en cuenta: (*i*) la comunicación entre las entidades que conforman el sistema (nodos de la red y CN), y (*ii*) el movimiento de los nodos. Con respecto al primero de ellos, NETA provee la funcionalidad necesaria para establecer la comunicación entre la entidad centralizada y los nodos de la red, ya sea para obtener la ubicación de todos ellos o para establecer las coordenadas de re-localización de los RN. A su vez, los protocolos de comunicación, transferencia de información, etc., que conforman el escenario MANET se obtienen gracias a la funcionalidad que oferta INET. En relación al movimiento de los UN, es INET quien, a través de sus múltiples implementaciones de patrones de movimiento conocidos (por ejemplo, RWP), se encarga de orquestar su movimiento. Por lo que respecta al movimiento de los RN, se desarrolla un módulo adicional de movimiento llamado `mobilityRelay` que se integra dentro de cada uno de ellos. Este módulo hereda la funcionalidad proporcionada por el módulo de INET `LineSegmentMobilityBase` para

mover cualquier nodo de forma lineal desde una posición hacia otra dada. El nuevo módulo, que se ubica dentro del marco de NETA, implementa el método `moveTo` que, una vez recibido el mensaje con la nuevas coordenadas de desplazamiento, se encarga de establecerlas y delega en la funcionalidad de `LineSegmentMobilityBase` para que conduzca a dicho nodo hacia su nueva localización.

En lo que respecta al nodo central, este implementará los métodos y rutinas necesarios para la obtención de las posiciones de los nodos, la computación de las nuevas posiciones optimizadas para los RN y el envío de dichas coordenadas a los correspondientes RN. El diagrama de flujo de la Figura 6.4 ilustra, de manera simplificada, el flujo de ejecución llevado a cabo por el nodo central. En dicha figura se observa la integración de la solución de respuesta en el marco de NETA a través de las fases de obtención y envío de información de localización desde y hacia los nodos de la red. Es la rutina de optimización de posicionamiento DRNS la que hace uso de ambas fases para llevar a cabo el posicionamiento optimizado de los RN.

6.1.3. Entorno de simulación

Son varios los aspectos que tienen un cierto impacto sobre el rendimiento de la solución de posicionamiento. Estos son, entre otros: (*i*) la frecuencia de ejecución de la rutina DRNS, que debería ser ajustada dependiendo del dinamismo de la red; (*ii*) el propio tiempo de ejecución invertido por dicha rutina; (*iii*) el rango de cobertura de los nodos, ya que con radios pequeños el número de particiones aumentaría; y (*iv*) la velocidad máxima de los nodos, especialmente la de los RN, que debería ser mayor que la de los UN. Debido a la relevancia e impacto de dichos aspectos sobre el rendimiento, se hace necesaria su modificación con objeto de adaptar el sistema en función del escenario en el que se aplica. Por esta razón, el módulo de control puede adaptar los aspectos anteriores a través de la modificación de los siguientes parámetros de configuración:

- `updateInterval`: establece el intervalo de tiempo en segundos entre ejecuciones consecutivas de la rutina de posicionamiento.
- `maxSpeed`: velocidad máxima establecida para los RN.
- `iterMax`: número máximo de iteraciones del algoritmo PSO.
- `nParticles`: número de partículas (población) utilizadas en el algoritmo PSO.
- `carrierFrequency`: frecuencia de la portadora para las señales transmitidas por el medio físico inalámbrico.
- `powerTX`: potencia de transmisión.
- `sensibility`: mínima potencia que ha de tener la señal en el receptor para que pueda ser procesada.

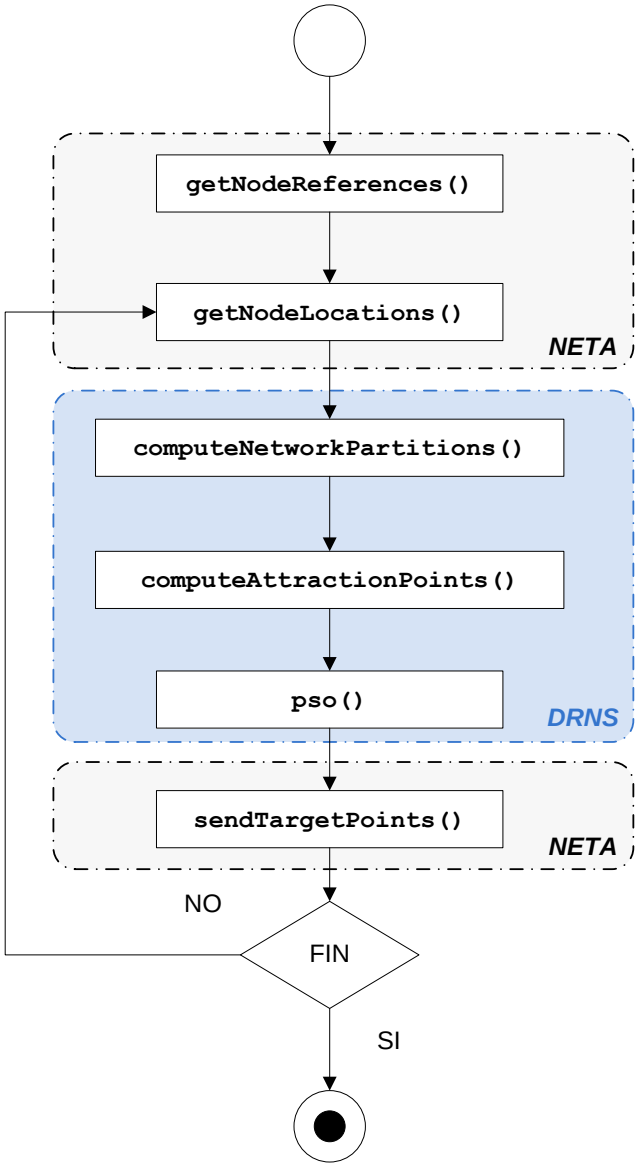


Figura 6.4: Diagrama de flujo funcional del nodo central y su integración con NETA.

Algunos de los parámetros de configuración tienen relación directa con las cuestiones planteadas anteriormente. Sin embargo, para el caso concreto del radio de cobertura, este se calcula indirectamente en base al modelo de propagación en espacio libre para comunicaciones inalámbricas que relaciona los parámetros `carrierFrequency`, `powerTX` y `sensibility`.

Para validar el correcto funcionamiento e integración del módulo de respuesta dentro de NETA, se propone el siguiente escenario de simulación. Un entorno MANET compuesto por 10 UN, variando el número de RN desde 0 hasta 3. El área contemplada es cuadrada, con dimensiones de 1000 m × 1000 m. A su vez, se con-

templa un radio de cobertura igual para todos los nodos de 250 m, de manera que sea probable la aparición de particiones en la red.

Por lo que respecta al movimiento de los nodos, los UN seguirán un patrón de movilidad RWP estableciendo un tiempo de pausa de 15 s. Adicionalmente, se establece su velocidad a 10 m/s, aumentando a 30 m/s la de los RN. Esto contribuye a la adaptación del sistema a los cambios en la topología de la red cuyo origen es el propio dinamismo de esta.

En lo referente a los protocolos y vías de comunicación empleadas para las diferentes capas, se establecen IEEE 802.11g y AODV como protocolos MAC y de encaminamiento, respectivamente. Adicionalmente, se configuran 10 flujos de datos UDP CBR de 4 paquetes de 512 bytes por segundo (`UDPBasicBurst`, en el contexto de INET).

Para cada experimento se define una duración de 100 s y se promedia el PDR de la red a partir de los resultados obtenidos durante 50 repeticiones.

6.1.4. Evaluación de los resultados

Para establecer una comparación equitativa, la Figura 6.5 presenta el rendimiento ofrecido por la red en términos del PDR medio obtenido, teniendo en cuenta el mismo número total de nodos en la red. Esto es, por un lado, contaremos con la presencia de hasta 3 RN considerando en total hasta 13 nodos (10 UN y hasta 3 RN) y, por otro, la misma cantidad de nodos UN. En la Figura 6.5 se observa cómo aumenta el PDR conforme lo hace el número total de nodos. A su vez, se aprecia una mejora en el rendimiento con la presencia de RN, como era de esperar. Es conveniente mencionar que, si bien es apreciable la mejora introducida por el sistema de posicionamiento de RN, esta no es tan importante como cabría esperar, sobre todo a partir de la intervención de 2 o más RN. Esto es principalmente debido a la versión del sistema DRNS implementado. Como ya se comentó, dicha implementación se corresponde con una solución previa a DRNS [177] en la que únicamente se contempla la mejora introducida por la función $p(R, A^*)$ encargada de atraer los RN hacia los AP (ver (4.18)). De esta forma, se obvia el cálculo y selección eficiente de AP tal y como se hace en la última propuesta de DRNS. Descrito en el trabajo [177] y de acuerdo a la definición de AP proporcionada por la solución [131], el número total de AP puede llegar a ser elevado. Consecuentemente, tal y como está definida la función $p(R, A^*)$, su primer término (el encargado de la propia atracción hacia los AP) cobra más relevancia, relegando a un segundo lugar la capacidad de mantener separados los RN. Por lo tanto, estos tenderán a permanecer juntos haciendo que el efecto sobre el rendimiento del sistema a partir de la introducción de 2 o más RN no sea relevante.

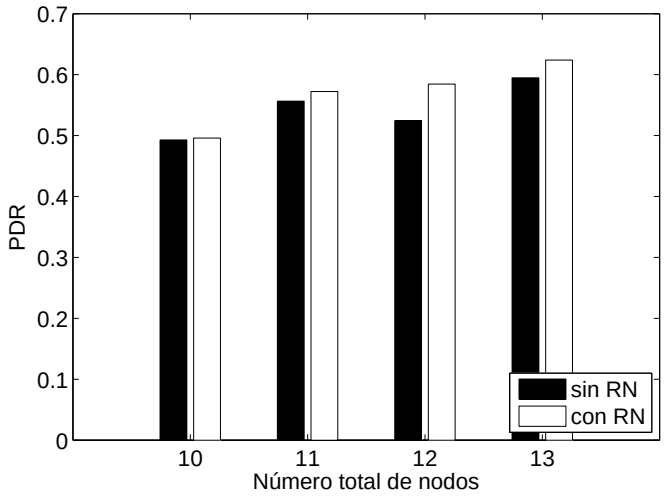


Figura 6.5: PDR obtenido utilizando RN y sin ellos. El número total de nodos se mantiene igual en cualquier caso.

Para finalizar, a través de los anteriores experimentos se valida el correcto funcionamiento e integración de la solución de respuesta dentro de NETA. A su vez, se corrobora la versatilidad y flexibilidad de dicho *framework* no solo para la implementación y pruebas de ataques en redes heterogéneas, sino también para la integración de soluciones de seguridad.

Como ya se ha indicado, a lo largo de la siguiente sección iremos un paso más allá hacia la propuesta de soluciones de seguridad integrales que agrupen en un solo sistema varios esquemas de defensa para la lucha contra los ataques y los efectos perniciosos que estos producen sobre el rendimiento de la red.

6.2. Integración de soluciones de seguridad con NETA

La puesta en marcha de la correspondiente respuesta o recuperación ante una amenaza hecha efectiva, es algo necesario pero no suficiente de cara a añadir capacidades de supervivencia al sistema. Además, carece de sentido responder o recuperar sin más, es decir, sin un mínimo conocimiento acerca de sobre qué o contra qué reaccionar.

A lo largo de esta sección se propone un nuevo *framework* o arquitectura que aborda la integración de diferentes esquemas de seguridad provistos en diferentes líneas de defensa y cuyo objetivo es posibilitar soluciones globales de seguridad. La Figura 6.6 muestra la arquitectura y elementos que componen dicho *framework* de integración, con especial énfasis en la necesaria interacción entre los módulos correspondientes a técnicas o soluciones de seguridad relativas a diferentes líneas

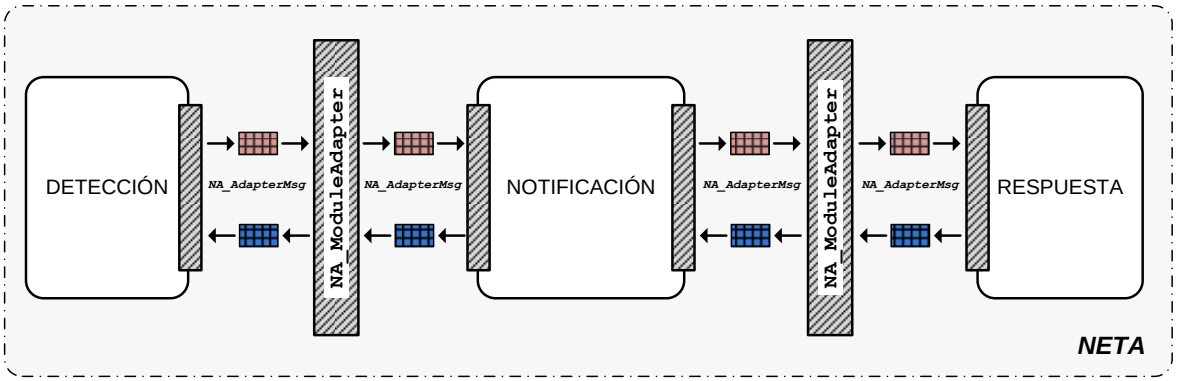


Figura 6.6: Arquitectura y elementos de comunicación para la integración de diferentes módulos de seguridad. Los módulos sombreados, así como los mensajes intercambiados, pertenecen al *framework* de integración, mientras que el resto de módulos representan cualquier esquema de seguridad desarrollado por el usuario.

de defensa. Dichos módulos representan cualquier esquema de defensa específico desarrollado por el usuario.

Como no puede ser de otra forma, se hace necesaria la interacción entre los módulos de defensa implicados. La solución propuesta se basa en el intercambio de mensajes especiales entre los módulos de defensa, siempre garantizando la aplicabilidad y flexibilidad de la solución de manera que se contemple la heterogeneidad de las soluciones de seguridad que existen en la literatura. Así, se propone la utilización de un módulo adicional al resto que haga las veces de intermediario en el intercambio de mensajes. Se podría decir que este módulo actúa a modo de interfaz estándar de comunicaciones entre módulos de defensa con el objeto de homogeneizar la interacción entre ellos. De esta manera, este módulo permite la sustitución de cualquiera de los otros (detección, notificación o respuesta) sin que afecte al funcionamiento global del sistema. El módulo en cuestión es el `NA_ModuleAdapter` que aparece en la Figura 6.6. La comunicación entre el *módulo adaptador* y los módulos de defensa se lleva a cabo de manera directa a través del intercambio de un único mensaje: `NA_AdapterMsg`.

A partir de lo explicado anteriormente, el flujo de comunicaciones que se lleva a cabo entre los módulos de defensa y el módulo adaptador se podría resumir de la siguiente manera. En primer lugar, el módulo que necesite comunicarse genera un mensaje `NA_AdapterMsg` en el que encapsula la información que considere relevante dependiendo de la funcionalidad implementada. Por ejemplo, el identificador del nodo atacante y el tipo de ataque, cuando se trate de un módulo de detección. Este mensaje se envía al módulo adaptador que se encarga de reenviarlo al módulo de destino. Una vez que el módulo destino ha recibido el mensaje, extraerá la información que considere útil del mensaje de adaptación. El envío y comunicación entre módulos forma parte de la funcionalidad que proporciona NETA.

Con objeto de validar la herramienta de integración propuesta, previamente se han desplegado e implementado en NETA algunas soluciones específicas que se enmarcan dentro del contexto de la detección, notificación y respuesta ante ataques de *dropping*. Con respecto a la solución de detección, se ha implementado el trabajo realizado en [185]. En este se propone un procedimiento para la detección de comportamientos maliciosos de tipo *dropping* en base a un determinado modelo preestablecido para el proceso de reenvío en MANET. Adicionalmente, se proponen dos modos de funcionamiento: aislado (*standalone*) y distribuido. Con el primero de ellos es el propio nodo el que, podríamos decir, detecta su comportamiento malicioso. En el segundo caso, el sistema detector se apoya en otros nodos de la red para realizar así un proceso de detección distribuido. Para simplificar el problema se utiliza aquí el modo aislado.

Una vez detectado el ataque, hemos de notificar el evento de cara a su potencial solución. Es en este momento cuando entra en juego el módulo de notificación. Para el caso que nos atañe, se ha implementado el sistema de notificación realizado a lo largo del trabajo [202]. Aunque este módulo es capaz de manejar diferentes tipos de mensajes, con ánimo de clarificar y simplificar el problema, solo uno de ellos será enviado al módulo de respuesta: un mensaje de alerta o *alert message*. Este mensaje informa de la presencia de uno o varios nodos maliciosos en la red y se transmite de forma *broadcast*.

Cuando el módulo de respuesta recibe el mensaje de alerta, los nodos maliciosos son aislados y el nodo de control (en el esquema implementado aquí y descrito en la Sección 6.1) comienza a gestionar el movimiento de los RN desplegados para mitigar los efectos del ataque en el rendimiento de la red.

A modo ilustrativo, la Figura 6.7 presenta de manera esquemática dónde se ubican realmente cada uno de los módulos anteriores dentro de los nodos de red implicados. Al utilizar el esquema de detección aislado, el módulo de detección se integra dentro del mismo nodo que hace las veces de *dropper*. A su vez, se añade el módulo de notificación y ambos se comunican, como se describió anteriormente, a través del adaptador que forma parte de la solución de integración propuesta. Por otro lado, y como destino de la notificación generada por el detector en primera instancia, nos encontramos con el nodo de control. Este integra el módulo de control que implementa la rutina de posicionamiento de los nodos *relay* y efectúa así la correspondiente reacción/respuesta ante el ataque. Al igual que para el nodo malicioso/detector, este nodo ha de integrar el módulo de notificación, encargado de recibir los mensajes de alerta. Ambos módulos se comunican entre sí a través del módulo adaptador proporcionado por el *framework* de integración.

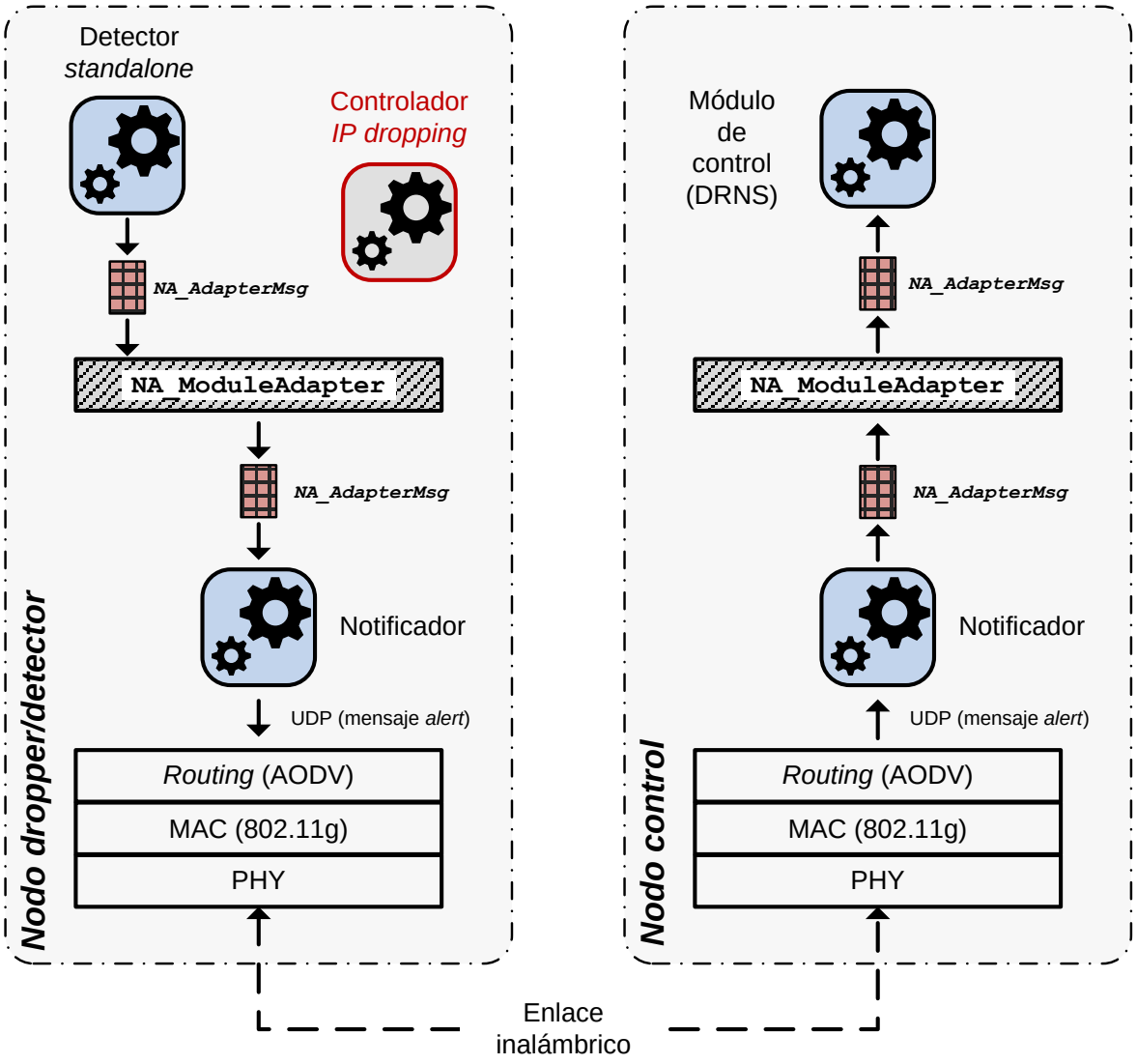


Figura 6.7: Detalle de la ubicación y comunicación de los módulos de detección, notificación, respuesta y módulo adaptador dentro del nodo malicioso/detector y de control.

6.2.1. Escenario de simulación y resultados preliminares

Una vez definidos los módulos de defensa a utilizar, así como su comunicación e integración gracias a la utilización de la herramienta desarrollada, a continuación se describen varios experimentos que evalúan el rendimiento de la red en base al PDR obtenido y contando con la presencia de ataques de *dropping*.

El escenario de simulación escogido se corresponde con el de la Sección 6.1.3, exceptuando la inclusión ahora de varios nodos maliciosos. En particular, dichos

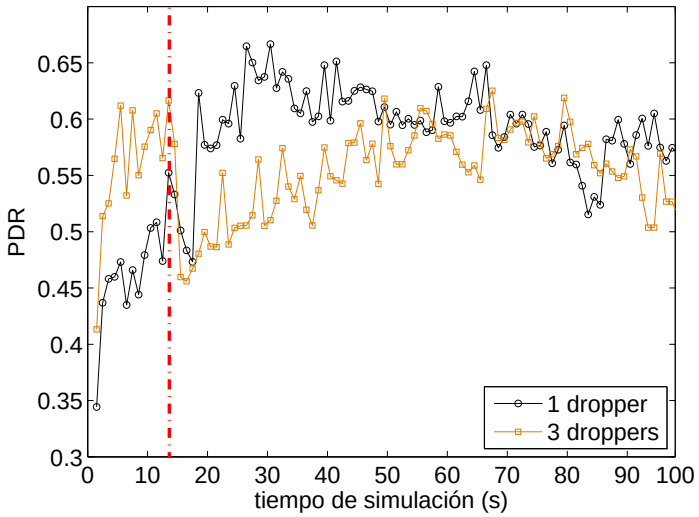


Figura 6.8: Impacto y recuperación del rendimiento de la red medido este a través del parámetro PDR considerando 10 UN, 3 UN y la presencia de 1 y 3 nodos maliciosos o *droppers*.

nodos se configuran para realizar un ataque *IP dropping* que provee el *framework* NETA (véase el Capítulo 5).

En lo que se refiere a los RN, estos permanecerán inmóviles en su posición de partida hasta que el módulo de respuesta sea consciente del ataque, es decir, hasta el momento en que recibe el mensaje de alerta. Es entonces cuando el módulo se encargará de conducir adecuadamente a los *relays* para mitigar los efectos perniciosos producidos sobre el rendimiento de la red.

Los nodos maliciosos actúan como nodos retransmisores antes del inicio del ataque, puesto que no son ni origen ni destino de ningún flujo de datos. Tras la activación del ataque, dichos nodos descartarán cualquier paquete que reciban (`droppingAttackProbability=1`), no cooperando así en el proceso de reenvío de paquetes.

En la Figura 6.8 se observa la evolución del parámetro PDR para el escenario de simulación elegido compuesto por 10 UN, 3 RN y en donde se consideran 1 y 3 nodos *droppers*. A través de dicha figura se observa el instante temporal en que comienza el ataque: justo en el segundo $t = 15$ de la simulación. Dicho momento se refleja en la figura a través de una línea vertical roja discontinua. Resulta notable la reducción del rendimiento de la red debido a la actuación de los nodos maliciosos. Además, como es de esperar, cuanto mayor es el número de nodos *droppers* mayor es el efecto producido. En estas circunstancias, incluso con un número elevado de nodos *droppers*, el módulo de respuesta es capaz de recuperar e incluso mejorar el rendimiento de la red antes del ataque. Finalmente, se observa cómo el sistema emplea más tiempo

en la recuperación del rendimiento perdido cuanto mayor es el número de nodos maliciosos, ya que el impacto sobre el rendimiento también lo es.

6.3. Conclusiones del capítulo

La consecución de sistemas invulnerables a amenazas a la seguridad es algo que se plantea complicado, por no decir prácticamente imposible. No obstante, un primer paso necesario en la consecución de este tipo de sistemas es la propuesta de sistemas robustos, resistentes y adaptables, en suma con capacidad de supervivencia. Conseguir este tipo de soluciones pasa inevitablemente por la intervención conjunta de diferentes esquemas de seguridad enmarcados dentro de distintas líneas de defensa. Con este fin en mente, se desarrolla e implementa aquí un *framework* o arquitectura de integración cuyo principal objetivo es proporcionar soluciones de seguridad globales que abarquen diferentes líneas de defensa.

Dicha herramienta se propone en el marco de NETA. Con objeto de evaluar la propuesta a través de los correspondientes experimentos, se han implementado diferentes esquemas de seguridad relativos a detección, notificación y respuesta, y se ha testado la solución global conseguida ante el despliegue de nodos maliciosos de tipo *dropper* en un entorno de red dado. Los resultados preliminares obtenidos a través de los experimentos pertinentes corroboran la viabilidad de nuestra propuesta de integración. Aún así, es necesario seguir trabajando en la evaluación de su flexibilidad y escalabilidad de cara a la posible incorporación de nuevas soluciones o esquemas de seguridad.

Publicaciones relacionadas

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

- L. Sánchez-Casado, G. Maciá-Fernández, P. García-Teodoro y **R. Magán-Carrión**. "A model of data forwarding in MANETs for lightweight detection of malicious packet dropping," *Computer Networks (Elsevier)*, vol. 87, pp. 44–58, julio 2015.
- 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," *Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, pp. 321-326, sept., 2014.
- **R. Magán-Carrión**, J. Camacho-Páez y P. García-Teodoro. "A Multiagent Self-healing System against Security Incidents in MANETs," *Highlights of Practical*

Applications of Heterogeneous Multi-Agent Systems (PAAMS), vol. 430, pp. 321–332, junio 2014.

- 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, sept. 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, octubre 2013.

A su vez, el diseño, desarrollo y puesta en marcha de las soluciones expuestas dentro de este capítulo se enmarcan dentro de varios Proyectos Fin de Carrera y Trabajos Fin de Grado llevados a cabo dentro del contexto del presente trabajo de tesis. Estos son:

- José Moreno Molina, "Integración de soluciones de seguridad en redes MANET con NETA". *Proyecto Fin de Carrera*, P. García Teodoro y **R. Magán Carrión** (Directores), Ingeniería de Telecomunicación, Universidad de Granada, Julio 2015.
- Francisco Jesús Cadenas Callejón, "Medidas de mejora y gestión de la conectividad en redes MANET mediante el empleo de agentes móviles". *Trabajo Fin de Grado*, J. Camacho Páez y **R. Magán Carrión** (Directores), Grado en Ingeniería de Tecnologías de la Comunicación, Universidad de Granada, Septiembre 2014.

Conclusiones
Y
Trabajo Futuro

Chapter 7

Conclusiones y trabajo futuro

Contenido

En el presente capítulo se resumen las principales conclusiones extraídas tras la realización del trabajo de tesis completo aquí expuesto. Estas se han ido detallando en cada uno de los capítulos previos, presentándose en este de forma unificada y resumida. Adicionalmente, se describen de manera breve las principales líneas de trabajo futuro para continuar así con la investigación iniciada durante el periodo de tesis doctoral.

7.1. Conclusiones

La versatilidad ofrecida por las redes ad hoc se debe principalmente a sus inherentes características. Sin embargo, tales cualidades motivan al mismo tiempo sus principales vulnerabilidades de seguridad. Consecuentemente, son necesarias redes más resistentes y robustas, con capacidades para la supervivencia ante las amenazas y potenciales ataques a las que están sometidas. Antes de proporcionar soluciones que solventen los problemas de seguridad de las redes ad hoc, hemos de realizar el estudio adecuado de la literatura especializada. En este sentido presentamos:

- Un estudio de las principales y relevantes amenazas actuales en el contexto de las redes ad hoc.
- Una revisión de las propuestas actuales de seguridad para la lucha frente a sus principales amenazas, especialmente enfocada en aquellas soluciones relacionadas con la respuesta y la tolerancia.
- Una novedosa organización de aquellas soluciones relacionadas con la respuesta y la tolerancia, las cuales clasificamos en tres grupos principales: *exclusión de nodos*, *exclusión de nodos y notificación* y *aislamiento de nodos*.

Una vez realizada la conveniente revisión de la literatura especializada, las principales contribuciones del presente trabajo. Estas se pueden organizar o agrupar en tres diferentes temas: la imputación de datos faltantes en WSN, el posicionamiento de nodos *relay* en MANET y la integración de soluciones de seguridad. En el contexto de las soluciones de imputación de datos faltantes en WSN, son varias las contribuciones aportadas y que exponemos a continuación:

- Desarrollamos un esquema de imputación de datos faltantes basado en el empleo de técnicas de análisis multivariante que utilizamos como sistema de respuesta/tolerancia ante fallos o ataques a la integridad de la información en escenarios como la lucha contra incendios. Adicionalmente, hemos desarrollado un sistema de monitorización y detección basado en el empleo de técnicas MSPC para discernir entre diferentes anomalías; por un lado, aquellas que provienen de cambios en el entorno (ambientales) y, por otro, las provenientes de comportamientos maliciosos derivados, por ejemplo, de ataques de *data tampering*.
- Hemos demostrado la influencia de la organización y estructuración de los datos dependiendo de la aplicación final del sistema. De esta manera, hemos desarrollado una novedosa manera de organizar los datos, los denominados *modelos locales*, que aumenta el rendimiento del procedimiento de imputación en comparación con otros tipos de organización, como por ejemplo los *modelos globales*.
- También hemos constatado el papel relevante que juega el algoritmo de encaminamiento desplegado en el rendimiento del procedimiento de recuperación de datos. Para corroborar tal influencia hemos propuesto y evaluado varias estrategias de encaminamiento, las cuales, dependiendo de qué sensores son atacados, podrían llevar al sistema a la obtención de diferentes resultados.
- Desarrollamos un simulador y entorno específicos para evaluar la solución propuesta. A través de este somos capaces de reproducir un hipotético escenario de lucha contra incendios donde las variaciones de temperatura son recogidas por los sensores desplegados en el área monitorizada.

- Hemos realizado un amplio número de experimentos para evaluar la capacidad y rendimiento del sistema de imputación de datos faltantes, no solo en entornos simulados, sino también en entornos reales como el despliegue LUCE.

A continuación se exponen las contribuciones aportadas en relación al posicionamiento de nodos *relay* en entornos MANET. Estas son:

- Hemos desarrollado e implementado un esquema de posicionamiento de nodos *relay* que maximiza la conectividad y el *throughput* en este tipo de escenarios, donde son continuos los cambios en la topología de la red.
- Para llevarlo a cabo, primero proponemos una formulación matemática del problema abordando dos cuestiones principales comunes en este tipo de problemas y entornos: (*i*) cuál es la mejor posición de los nodos *relay* en un determinado momento, y (*ii*) cómo han de ser movidos hacia dichas localizaciones.
- A partir de la formulación anterior, hemos desarrollado e implementado un sistema flexible y versátil de posicionamiento de nodos *relay* al que llamamos DRNS (*Dynamical Relay Node placement Solution*). Esta solución se basa en un trabajo previo, en donde primero, se solventan graves deficiencias encontradas en su diseño y, segundo, se mejora notablemente su rendimiento. DRNS se basa principalmente en el uso del algoritmo PSO y metodologías inspiradas en MPC para la obtención de las posiciones optimizadas de los nodos *relay* y el control de sus movimientos.
- Este problema se aborda considerando un número fijo de nodos *relay* que, aunque complica la resolución del problema, lo hace más realista y aplicable en la práctica.
- Aunque el sistema propuesto se enfoca en entornos variables con el tiempo, hemos validado la utilidad del módulo encargado de la optimización de la posición de los AP, como una solución factible al problema del posicionamiento de nodos *relay* en entornos estáticos.
- A través de un extensivo conjunto de experimentos en simulación, concluimos la eficacia de nuestra propuesta en términos de los objetivos propuestos de rendimiento: conectividad y *throughput*. Además, concluimos la capacidad de adaptación de nuestro sistema frente a los cambios producidos en entornos dinámicos.
- Desde el punto de vista de la seguridad, hemos corroborado la aplicación de nuestro sistema como solución válida de respuesta/tolerancia ante la presencia de nodos maliciosos. En concreto, hemos probado la solución como medida de respuesta ante ataques de *dropping* de manera satisfactoria.
- Hemos llevado a cabo un estudio sobre la complejidad y tiempo de ejecución de la propuesta. Aunque es cierto que se añade cierta complejidad computacional

con respecto a otras soluciones, también lo es la mejora que se introduce en el rendimiento.

- Hemos desplegado nuestro esquema en entornos reales como el proporcionado por *IDSIA Swarm Robotics Laboratory*. Para ello, hemos diseñado e implementado una arquitectura específica que solventa la problemática asociada a este tipo de entornos. A través de una experimentación similar a aquella realizada en simulación, comprobamos la viabilidad práctica de nuestra propuesta. Si bien el tiempo de ejecución de la solución podría ser un factor determinante en entornos reales, hemos constatado que el incremento en tiempo computacional no tiene un impacto significativo en su rendimiento.

Finalmente, con respecto a la integración de soluciones de seguridad, se destacan las siguientes contribuciones:

- Hemos desarrollado e implementado un *framework* de integración e interacción de soluciones de defensa. Este presenta una arquitectura modular y versátil que contempla la amplia y heterogénea variedad de soluciones de seguridad existentes.
- A modo de prueba de concepto, se han integrado varios esquemas de defensa propuestos en la literatura dentro del marco de la detección, notificación y respuesta. Además, para corroborar la completa integración y operación del conjunto completo como solución de seguridad global, se simuló el ataque de *dropping attack* y se expusieron los resultados de recuperación del conjunto completo ante este tipo de ataques.

7.2. Líneas de trabajo futuro

La línea de investigación iniciada durante el trabajo de tesis realizado deja abiertas una serie de ideas de trabajo futuro. Estas son, entre otras:

- Estudiar nuevos métodos de organización de los datos con objeto de mejorar el rendimiento de la recuperación de datos faltantes en WSN, principalmente orientados al incremento de la correlación en la información utilizada por el proceso de imputación. Podríamos pensar en desarrollar modelos locales para cada sensor específico, particularizando dicho modelo.
- Evaluar la influencia en el rendimiento de la recuperación de datos en WSN cuando se emplean modelos locales en conjunción con modelado dinámico y estrategias de encaminamiento también dinámicas.

- Abordar sistemas de detección que disciernan de manera autónoma y automática cuál fue la causa de la anomalía presentada. Esta solución formaría parte de una solución global de seguridad para la lucha frente a ataques a la integridad de la información en WSN. Para ello se podría pensar en utilizar técnicas basadas en análisis discriminante.
- Diseñar y construir la versión descentralizada y distribuida del esquema DRNS. Esta nueva versión añadiría la capacidad de auto-gestión, proporcionaría escalabilidad y resistencia ante fallos.
- Diseñar e implementar un nuevo módulo de adaptación ante cambios en el entorno para el sistema DRNS. Tal módulo debería ser capaz de re-computar dinámicamente los parámetros y meta-parámetros del algoritmo PSO en concordancia con el dinamismo de la red.
- Desarrollar funciones de coste más elaboradas, para ser utilizadas en el sistema DRNS, que contemplen aspectos más realistas y relevantes en redes MANET. Por ejemplo, parámetros importantes son: el PDR de la red, las interferencias en las comunicaciones, contemplar áreas densas en lo que se refiere al número de nodos, la energía de los nodos, el *goodput* de la red, etc.
- Analizar sistemáticamente cómo medir la capacidad o capacidades de supervivencia de un sistema de forma objetiva y cuantitativa mediante un conjunto de métricas presentes en este tipo de redes. Por ejemplo, a través de la consideración de aspectos como la energía, el ancho de banda, la conectividad, el *throughput*, el tiempo de vida de los enlaces, etc.
- Emplear la información anterior para proponer esquemas de seguridad globales con capacidad de supervivencia que consideren los anteriores y otros aspectos para, primero, ser capaces de medir cuantitativamente dicha capacidad y, segundo, abogar por su optimización a través del empleo de técnicas de análisis multivariante o enfoques provenientes del análisis superviviente o *survival analysis*.
- Extender la solución DRNS a escenarios más complejos no limitados a dos dimensiones. Por ejemplo, su aplicación en entornos FANET, donde se contempla una tercera dimensión.
- Evaluar algoritmos de optimización alternativos para su comparación con el que actualmente usa DRNS.
- Desarrollar e implementar nuevos ataques y soluciones de defensa dentro del marco de NETA y del *framework* de integración provisto en la tercera parte del presente trabajo de tesis.

Bibliografía

[1] Real Academia Española, "Procedencia y significado del término ad hoc." [Online; Accessed 30-March-2016] `http://lema.rae.es/dpd/srv/search?key=hoc`.

[2] D. G. Reina, M. Askalani, S. L. Toral, F. Barrero, E. Asimakopoulou, and N. Bessis, "A Survey on Multihop Ad Hoc Networks for Disaster Response Scenarios," *International Journal of Distributed Sensor Networks*, vol. 2015, p. 16 pages, October 2015.

[3] M. Azer, S. El-Kassas, and M. El-Soudani, "Security in Ad Hoc Networks: From Vulnerability to Risk Management," in *Third International Conference on Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09*, pp. 203–209, June 2009.

[4] R. Hekmat, "Introduction to Ad-hoc Networks," in *Ad-hoc Networks: Fundamental Properties and Network Topologies*, pp. 1–8, Springer Netherlands, 2006.

[5] 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.

[6] R. Di Pietro, S. Guarino, N. V. Verde, and J. Domingo-Ferrer, "Security in wireless ad-hoc networks – A survey," *Computer Communications*, vol. 51, pp. 1–20, September 2014.

[7] M. O. Pervaiz, M. Cardei, and J. Wu, "Routing Security in Ad Hoc Wireless Networks," in *Network Security* (S. C.-H. Huang, D. MacCallum, and D.-Z. Du, eds.), pp. 117–142, Springer US, 2010.

[8] Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan, and S. Anwar, "Intrusion response systems: Foundations, design, and challenges," *Journal of Network and Computer Applications*, vol. 62, pp. 53–74, February 2016.

[9] 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. S. Shen, and D.-Z. Du, eds.), Signals and Communication Technology, pp. 103–135, Springer US, 2007.

[10] M. Lima, A. dos Santos, and G. Pujolle, "A survey of survivability in mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 11, pp. 66–77, March 2009.

[11] A. Azni, R. Ahmad, Z. Mohamad Noh, F. Hazwani, and N. Hayaati, "Network survivability analysis modeling approach for MANETS: A systematic review," in *2014 Fourth World Congress on Information and Communication Technologies (WICT)*, pp. 74–79, Dec. 2014.

[12] 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, April 2014.

[13] H. Aldabbas, H. Janicke, R. AbuJassar, and T. Alwada'n, "Ensuring Data Confidentiality and Privacy in Mobile Ad Hoc Networks," in *Advances in Computer Science and Information Technology. Networks and Communications* (N. Meghanathan, N. Chaki, and D. Nagamalai, eds.), no. 84 in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 490–499, Springer Berlin Heidelberg, January 2012.

[14] S. Misra, I. Woungang, and S. Chandra Misra, eds., *Guide to Wireless Ad Hoc Networks*. Computer Communications and Networks, Springer London, 2009.

[15] S. Abbas, M. Merabti, and D. Llewellyn-Jones, "On the evaluation of reputation and trust-based schemes in mobile ad hoc networks," *Security and Communication Networks*, vol. 8, pp. 4041–4052, Dec. 2015.

[16] 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, October 2007.

[17] 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.

[18] L. Sánchez Casado, *Anomaly-based multi-layer intrusion detection for MANET environments*. PhD thesis, ETSIIT, University of Granada, October 2014.

[19] B. Mokhtar and M. Azab, "Survey on Security Issues in Vehicular Ad Hoc Networks," *Alexandria Engineering Journal*, vol. 54, pp. 1115–1126, Dec. 2015.

[20] A. Ahmed, K. A. Bakar, M. I. Channa, K. Haseeb, and A. W. Khan, "A survey on trust based detection and isolation of malicious nodes in ad-hoc and sensor networks," *Frontiers of Computer Science*, vol. 9, pp. 280–296, Dec. 2014.

[21] X. Liao, D. Hao, and K. Sakurai, "Classification on attacks in wireless ad hoc networks: A game theoretic view," in *2011 7th International Conference on Networked Computing and Advanced Information Management (NCM)*, pp. 144–149, June 2011.

[22] 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, January 2012.

[23] 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, April 2011.

[24] 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, September 2002.

[25] 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.

[26] 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.

[27] H. Deng, W. Li, and D. P. Agrawal, "Routing Security in Wireless Ad Hoc Networks," *IEEE Communications Magazine*, vol. 40, pp. 70–75, October 2002.

[28] 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, August 2002.

[29] 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.

[30] S. Khan, K.-K. Loo, N. Mast, and T. Naeem, "SRPM: Secure Routing Protocol for IEEE 802.11 Infrastructure Based Wireless Mesh Networks," *Journal of Network and Systems Management*, vol. 18, pp. 190–209, October 2009.

[31] S. Khan, N. A. Alrajeh, and K.-K. Loo, "Secure route selection in wireless mesh networks," *Computer Networks*, vol. 56, pp. 491–503, February 2012.

[32] 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.

[33] 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.

[34] 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.

[35] 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, March 2004.

[36] 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, July 2012.

[37] 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, January 2010.

[38] 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, March 2010.

[39] 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, March 2003.

[40] 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.

[41] R. Mitchell and I.-R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, pp. 1–23, April 2014.

[42] 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.

[43] 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, October 2005.

[44] D. Djenouri and N. Badache, "On eliminating packet droppers in MANET: A modular solution," *Ad Hoc Networks*, vol. 7, pp. 1243–1258, August 2009.

[45] 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, March 2005.

[46] 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.

[47] S. Buchegger and J. 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, June 2002.

[48] 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, January 2007.

[49] 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.

[50] 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, August 2009.

[51] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*. The Morgan Kaufmann Series in Data Management Systems, Boston: Morgan Kaufmann, 2011.

[52] Y. Zhang, W. Lee, and Y. A. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," *Wireless Networks*, vol. 9, pp. 545–556, Sep. 2003.

[53] 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.

[54] 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, February 2007.

[55] 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*, Springer Berlin Heidelberg, September 2004.

[56] 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, June 2009.

[57] 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)*, June 2012.

[58] 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.

[59] 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.

[60] A. Ahmed, K. A. Bakar, M. I. Channa, and A. W. Khan, "A Secure Routing Protocol with Trust and Energy Awareness for Wireless Sensor Network," *Mobile Networks and Applications*, pp. 1–14, January 2016.

[61] 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.

[62] 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.

[63] 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, January 2012.

[64] 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, January 2002.

[65] 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, March 2010.

[66] H. Jadidoleslamy, M. R. Aref, and H. Bahramgiri, "A fuzzy fully distributed trust management system in wireless sensor networks," *AEU - International Journal of Electronics and Communications*, vol. 70, pp. 40–49, Jan. 2016.

[67] D. Kukreja, S. K. Dhurandher, and B. V. R. Reddy, "Enhancing the Security of Dynamic Source Routing Protocol Using Energy Aware and Distributed Trust Mechanism in MANETs," in *Intelligent Distributed Computing* (R. Buyya and S. M. Thampi, eds.), no. 321 in Advances in Intelligent Systems and Computing, pp. 83–94, Springer International Publishing, 2015.

[68] 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.

[69] M. Su, "Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems," *Computer Communications*, vol. 34, pp. 107–117, January 2011.

[70] V. Laxmi, C. Lal, M. S. Gaur, and D. Mehta, "JellyFish attack: Analysis, detection and countermeasure in TCP-based MANET," *Journal of Information Security and Applications*, vol. 22, pp. 99–112, June 2015.

[71] 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, February 2011.

[72] 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, January 2012.

[73] A. Shabut, K. Dahal, S. Bista, and I. Awan, "Recommendation Based Trust Model with an Effective Defence Scheme for MANETs," *IEEE Transactions on Mobile Computing*, vol. 14, pp. 2101–2115, October 2015.

[74] 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, June 2009.

[75] 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, October 2009.

[76] 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, January 2010.

[77] 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, July 2010.

[78] K.-L. Tsai, M. Ye, and F.-Y. Leu, "Secure Power Management Scheme for WSN," in *Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats*, MIST '15, (New York, NY, USA), pp. 63–66, ACM, 2015.

[79] M. Balakrishnan, H. Huang, Y. Jaradat, S. Pawar, S. Misra, and R. Asorey-Cacheda, "Null Frequency Jamming of Dynamic Routing in Wireless Ad Hoc Networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pp. 1 –5, Dec. 2011.

[80] A. Hamieh, "POWJAM: A power reaction system against jamming attacks in wireless ad hoc networks," in *Wireless On-demand Network Systems and Services (WONS), 2012 9th Annual Conference on*, pp. 9 –15, January 2012.

[81] J.-H. Cho, A. Swami, and I.-R. Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks," *IEEE Communications Surveys Tutorials*, vol. 13, pp. 562–583, Nov. 2011.

[82] 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, August 2012.

[83] 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, August 2011.

[84] 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, March 2008.

[85] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, pp. 6–28, Dec. 2004.

[86] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, pp. 2292–2330, August 2008.

[87] M. D. Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Transactions on Sensor Networks*, vol. 8, pp. 1–7, August 2011.

[88] M. Lima, A. dos Santos, and G. Pujolle, "A survey of survivability in mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 11, pp. 66–77, 2009.

[89] K. M. X. Chen, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, pp. 52–73, June 2009.

[90] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, vol. 35, pp. 54–62, Dec. 2002.

[91] I. T. Jolliffe, *Principal Components Analysis*. Series in Statistics, Springer New York, 2 ed., 2002.

[92] J. E. Jackson, *A User's Guide to Principal Components*. Wiley Series in Probability and Statistics, Wiley, 2004.

[93] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Analytica Chimica Acta*, vol. 185, pp. 1–17, July 1986.

[94] S. Wold, M. Sjöström, and L. Eriksson, "Pls-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, pp. 109–130, October 2001.

[95] F. Arteaga and A. Ferrer, "Dealing with missing data in mspc: several methods, different interpretations, some examples," *Journal of Chemometrics*, vol. 16, pp. 408–418, August 2002.

[96] F. Arteaga and A. Ferrer, "Framework for regression-based missing data imputation methods in on-line mspc," *Journal of Chemometrics*, vol. 19, pp. 439–447, Dec. 2005.

[97] J. Camacho and J. Picó, "Multi-phase principal component analysis for batch processes modelling," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, pp. 127–136, April 2006.

[98] J. Flores-Cerrillo and J. F. MacGregor, "Control of batch product quality by trajectory manipulation using latent variable models," *Journal of Process Control*, vol. 14, pp. 539–553, August 2004.

[99] M. H. Bharati, J. J. Liu, and J. F. MacGregor, "Image texture analysis: methods and comparisons," *Chemometrics and Intelligent Laboratory Systems*, vol. 72, pp. 57–71, June 2004.

[100] M. Xie, H. Song, T. Biming, and P. Sazia, "Anomaly detection in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 34, pp. 1302–1325, July 2011.

[101] Y. Li and L. E. Parker, "A spatial-temporal imputation technique for classification with missing data in a wireless sensor network," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Nice), pp. 3272–3279, IEEE, Sept. 2008.

[102] K. Smarsly and K. H. Law, "Decentralized fault detection and isolation in wireless structural health monitoring systems using analytical redundancy," *Advances in Engineering Software*, vol. 73, pp. 1–10, July 2014.

[103] L. Gruenwald, M. S. Sadik, R. Shukla, and H. Yang, "Dems: a data mining based technique to handle missing data in mobile sensor network applications," in *Proceedings of the Seventh International Workshop on Data Management for Sensor Networks (DMSN)*, (NY, USA), pp. 26–32, ACM New York, Sept. 2010.

[104] J. C. Lim and C. J. Bleakley, "Robust data collection and lifetime improvement in wireless sensor networks through data imputation," in *Fifth International Conference on Systems and Networks Communications (ICSNC)*, (Nice), pp. 64–69, IEEE, August 2010.

[105] E. W. Dereszynski and T. G. Dietterich, "Spatiotemporal models for data-anomaly detection in dynamic environmental monitoring campaigns," *ACM Transactions on Sensor Networks*, vol. 8, pp. 1–3, August 2011.

[106] Y. Li and L. E. Parker, "Nearest neighbor imputation using spatial-temporal correlations in wireless sensor networks," *Information Fusion*, vol. 15, pp. 64–79, January 2014.

[107] H. Yu, Y. Zhuang, and W. Wang, "Distributed h_∞ filtering in sensor networks with randomly occurred missing measurements and communication link failures," *Information Sciences*, vol. 222, pp. 424–438, February 2013.

[108] M. A. Livani and M. Abadi, "A pca-based distributed approach for intrusion detection in wireless sensor networks," in *International Symposium on Computer Networks and Distributed Systems (CNDS)*, (Tehran), pp. 55–60, IEEE, February 2011.

[109] N. Chitradevi, K. Baskaran, V. Palanisamy, and D. Aswini, "Designing an efficient pca based data model for wireless sensor networks," in *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief (ACWR)*, (NY, USA), pp. 147–154, ACM New York, 2011.

[110] K. H. Esbensen, *Multivariate Data Analysis - in practice*. CAMO, 5 ed., 2009.

[111] S. Wold, "Cross-validatory estimation of the number of components in factor and principal components models," *Technometrics*, vol. 20, no. 4, pp. 397–405, 1978.

[112] J. Camacho and A. Ferrer, "Cross-validation in pca models with the element-wise k-fold (ekf) algorithm: theoretical aspects," *Journal of Chemometrics*, vol. 26, pp. 361–373, May 2012.

[113] W. Ku, R. H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 30, pp. 179–196, Nov. 1995.

[114] J. Chen and K. Liu, "On-line batch process monitoring using dynamic PCA and dynamic PLS models," *Chemical Engineering Science*, vol. 57, pp. 63–75, January 2002.

[115] H.-J. Abdulnasser, "Multivariate tests for autocorrelation in the stable and unstable VAR models," *Economic Modelling*, vol. 21, pp. 661–683, July 2004.

[116] H. Hotelling, *Multivariate Quality Control. Techniques of Statistical Analysis*. MacGraw-Hill, 1947.

[117] T. Kourti and J. F. MacGregor, "Multivariate spc methods for process and product monitoring," *Journal of Quality Technology*, vol. 28, no. 4, pp. 409–428, 1996.

[118] J. Camacho, J. Picó, and A. Ferrer, "Bilinear modelling of batch processes. Part II: a comparison of pls soft-sensors," *Journal of Chemometrics*, vol. 22, pp. 533–547, July 2008.

[119] J. E. Jackson, *A User's Guide to Principal Components*. Series in Probability and Statistics, Wiley, 2008.

[120] R. Bro, K. Kjeldahl, A. Smilde, and H. Kiers, "Cross-validation of component models: A critical look at current methods," *Analytical and Bioanalytical Chemistry*, vol. 390, pp. 1241–1251, January 2008.

[121] J. Camacho and A. Ferrer, "Cross-validation in pca models with the element-wise k-fold (ekf) algorithm: Practical aspects," *Chemometrics and Intelligent Laboratory Systems*, vol. 131, pp. 37–50, February 2014.

[122] EPFL (École Polytechnique Fédérale de Lausanne), "Luce: Laussanne urban canopy experiment." [Online; Accessed 30-March-2016] `http://lcav.epfl.ch/page-86035-en.html`.

[123] K. A. Kellner and D. H. K. Behrends, "Simulation environments for wireless sensor networks," Tech. Rep. IFI-TB-2010-04, Institute of Computer Science, Georg-August-Universitat Gottingen, Germany, June 2010.

[124] E. S. Manolakos and D. V. Manatakis, "Temperature field modeling and simulation of wireless sensor network behavior during a spreading wildfire," in *European Signal Processing Conference (EUSIPCO 2008)*, (Laussanne), pp. 1–5, IEEE, August 2008.

[125] J. Camacho, J. Picó, and A. Ferrer, "Bilinear modelling of batch processes. Part I: theoretical discussion," *Journal of Chemometrics*, vol. 22, pp. 299–308, May 2008.

[126] B. M. Wise, N. B. Gallagher, J.S.W.W, and R. Koch, "PLS_toolbox - advanced chemometrics software for use with matlab." [Online; Accessed 30-March-2016] `http://www.eigenvector.com/software/pls_toolbox.htm`.

[127] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 1992.

[128] EPFL (École Polytechnique Fédérale de Lausanne), "Sensorscope: Sensor networks for environmental monitoring." [Online; Accessed 30-March-2016] `http://lcav.epfl.ch/cms/site/lcav/lang/en/sensorscope-en`.

[129] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, pp. 621–655, June 2008.

[130] H. Liu, X. Chu, Y. W. Leung, and R. Du, "Simple movement control algorithm for bi-connectivity in robotic sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 994–1005, Sept. 2010.

[131] O. Dengiz, A. Konak, and A. E. Smith, "Connectivity management in mobile ad hoc networks using particle swarm optimization," *Ad Hoc Networks*, vol. 9, pp. 1312–1326, Sept. 2011.

[132] H. Safa, W. El-Hajj, and H. Zoubian, "Particle swarm optimization based approach to solve the multiple sink placement problem in wsns," in *Proceedings of the IEEE International Conference on Communications (ICC) - Wireless Network Symposium*, (Ottawa), pp. 5445–5450, June 2012.

[133] M. Azharuddin and P. Jana, "A ga-based approach for fault tolerant relay node placement in wireless sensor networks.," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, (Hooghly), pp. 1–6, February 2015.

[134] M. Ayyash, Y. Alsbou, and M. Anan, "Introduction to Mobile Ad-Hoc and Vehicular Networks," in *Wireless Sensor and Mobile Ad-Hoc Networks* (D. Benhaddou and A. Al-Fuqaha, eds.), pp. 33–46, Springer New York, 2015.

[135] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the the 6th. International Symposium on Micro Machine and Human Science*, (Nagoya), pp. 39–43, October 1995.

[136] E. F. Camacho and C. Bordons, *Advanced Textbooks in Control and Signal Processing*. Series in Statistics, Springer London, 2 ed., 2007.

[137] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, pp. 303–317, June 2005.

[138] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 56, pp. 134–138, January 2007.

[139] W. Zhang, G. Xue, and S. Misra, "Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms," in *Proceedings of the 26th. IEEE International Conference on Computer Communications (INFOCOM)*, (Anchorage), pp. 1649–1657, May 2007.

[140] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed fault tolerance," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 216–228, February 2010.

[141] X. Han, X. Cao, E. L. Lloyd, and C. S. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 643–656, May 2010.

[142] A. Kashyap, S. Khuller, and M. Shayman, "Relay placement for fault tolerance in wireless networks in higher dimensions," *Computational Geometry*, vol. 44, pp. 206–215, May 2011.

[143] B. Hao, J. Tang, and G. Xue, "Fault-tolerant relay node placement in wireless sensor networks: Formulation and approximation," *Workshop on High Performance Switching and Routing (HPSR)*, vol. 2004, pp. 246–250, 2004.

[144] H. Liu, P. J. Wan, and X. Jia, "Fault-tolerant relay node placement in wireless sensor networks," in *Computing and Combinatorics*, pp. 230–239, Kunming, China: Springer-Verlag Berlin Heidelberg, August 2005.

[145] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 1399–1411, June 2012.

[146] S. Misra, S. D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements," in *Proceedings of the 26th. IEEE International Conference on Computer Communications (INFOCOM)*, pp. 879–887, 2008.

[147] S. Misra, S. D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks: Formulation and approximations," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 434–447, Nov. 2010.

[148] A. Kashyap and M. Shayman, "Relay placement and movement control for realization of fault-tolerant and hoc networks," in *41st. Anual Conference on Information Sciences and Systems*, (Baltimore), pp. 783–788, March 2007.

[149] S. Perumal, J. Baras, C. Graff, and D. Yee, "Aerial platform placement algorithms to satisfay connectivity, capacity and survivability constraints in wireless ad-hoc networks," in *Military Communications Conference (MILCOM)*, (San Diego), pp. 1–7, Nov. 2008.

[150] I. Rubin and R. Zhang, "Placement of uavs as communication relays aiding mobile ad hoc wireless networks," in *Military Communications Conference (MILCOM)*, (Orlando), pp. 1–7, Oct. 2007.

[151] A. Alfaqdhly, U. Baroudi, and M. Younis, "Optimal node repositioning for tolerating node failure in wireless sensor actor network," in *Proceedings of the 25th. Biennal Symposium on Communications*, (Kingston), pp. 67–71, May 2010.

[152] A. Alfaqdhly, U. Baroudi, and M. Younis, "Least distance movement recovery approach for large scale wireless sensor and actor networks," in *7th. International Conference on Wireless Communications and Mobile Computing Conference (IWCMC)*, (Istanbul), pp. 2058–2063, July 2011.

[153] M. Younis, S. Lee, and A. Abbasi, "A localized algorithm for restoring internode connectivity in networks of moveable sensors," *IEEE Transactions on Computers*, vol. 59, pp. 1669–1682, August 2010.

[154] I. F. Senturk, K. Akkaya, and S. Yilmaz, "Relay placement for restoring connectivity in partitioned wireless sensor networks under limited information," *Ad Hoc Networks*, vol. 13, pp. 487–503, February 2014.

[155] C. Lin, Y. Li, and D. Deng, "A bat-inspired algorithm for router node placement with weighted clients in wireless mesh networks," in *9th. International Conference on Communications and Networking in China*, (Maoming), pp. 139–143, August 2014.

[156] J. Tang, B. Hao, and A. Sen, "Relay node placement in large scale wireless sensor networks," *Computer Communications*, vol. 29, pp. 490–501, February 2006.

[157] X. Cheng, D. Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, pp. 347–355, June 2008.

[158] S. Lee and M. Younis, "Optimized relay node placement for connecting disjoint wireless sensor networks," *Computer Networks*, vol. 56, pp. 2788–2804, August 2012.

[159] A. Nigam and Y. K. Agarwal, "Optimal relay node placement in delay constrained wireless sensor network design," *European Journal of Operational Research*, vol. 233, pp. 220–233, February 2014.

[160] A. Nigam and Y. K. Agarwal, "Optimal relay placement in wireless sensor networks using node cut inequalities," in *Proceedings of the 4th. International Conference on Communication Systems and Networks (COMSNETS)*, (Bangalore), pp. 1–8, January 2012.

[161] A. Barolli, F. Xhafa, and M. Takizawa, "Optimization problems and resolution methods for node placement in wireless mesh networks," in *Proceedings of the 14th. International Conference on Network-Based Information Systems (NBiS)*, (Tirana), pp. 126–134, Sept. 2011.

[162] L. Wang, X. Fu, J. Fang, H. Wang, and M. Fei, "Optimal node placement in industrial wireless sensor networks using adaptive mutation probability binary particle swarm optimization algorithm," in *Proceedings of the 7th. International Conference on Natural Computation (ICNC)*, (Shanghai), pp. 2199–2203, July 2011.

[163] N. Aziz, A. W. Mohemmed, and M. Y. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram," in *Proceedings of the International Conference on Networking, Sensing and Control (ICNSC)*, (Okayama), pp. 602–607, March 2009.

[164] H. Z. Abidin and N. Din, "Provisioning wsn coverage via minimax based sensor node placement scheme," in *Proceedings of the International Conference on Wireless Communications and Applications*, (Kuala Lumpur), pp. 1–5, October 2012.

[165] B. Wang, H. Xu, W. Liu, and H. Liang, "A novel node placement for long belt coverage in wireless networks," *IEEE Transactions on Computers*, vol. 62, pp. 2341–2353, Nov. 2013.

[166] K. Xu, Q. Wang, H. Hassanein, and G. Takahara, "Optimal wireless sensor networks (wsns) deployment: Minimum cost with lifetime constraint," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 454–461, August 2005.

[167] Q. Wang, G. Takahara, H. Hassanein, and K. Xu, "On relay node placement and locally optimal traffic allocation in heterogeneous wireless sensor networks," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, (Sidney), pp. 657–664, Nov. 2005.

[168] Q. Wang, K. Xu, H. Hassanein, and G. Takahara, "Minimum cost guaranteed lifetime design for heterogeneous wireless sensor networks (wsns)," in *Proceedings of the 24th. IEEE International Performance, Computing, and Communications Conference*, pp. 599–604, April 2005.

[169] Q. Wang, K. Xu, G. Takahara, and H. Hassanein, "Locally optimal relay node placement in heterogeneous wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, (St. Louis), pp. 3549–3553, Dec. 2005.

[170] E. F. Flushing and G. A. D. Caro, "A flow-based optimization model for throughput-oriented relay node placement in wireless sensor networks," in *Proceedings of the 28th. Annual ACM Symposium on Applied Computing*, (Coimbra, Portugal), pp. 632–639, May 2013.

[171] E. H. and K. F.A., "Malicious AODV: implementation and analysis of routing attacks in MANETs," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, (Liverpool), pp. 1181–1187, June 2012.

[172] R. Magán-Carrión, J. Camacho-Páez, and P. García-Teodoro, "A security response approach based on the deployment of mobile agents," in *Advances on Practical Applications of Agents and Multi-Agent Systems* (Y. Demazeau, T. Ishida, J. M. Corchado, and J. Bajo, eds.), no. 7879 in Lecture Notes in Computer Science, pp. 182–191, Salamanca, Spain: Springer Berlin Heidelberg, May 2013.

[173] R. Magán-Carrión, J. Camacho-Páez, and P. García-Teodoro, "A Security Response Approach Based on the Deployment of Mobile Agents: A Practical Vision," in *Advances on Practical Applications of Agents and Multi-Agent Systems* (Y. Demazeau, T. Ishida, J. M. Corchado, and J. Bajo, eds.), no. 7879 in Lecture

Notes in Computer Science, pp. 308–311, Springer Berlin Heidelberg, May 2013.

[174] J. Nocedal and S. Wright, *Numerical Optimization*. New York: Springer, 2006.

[175] 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.

[176] T. Saitou, M. Nukada, and Y. Uchimura, "Deployment control of mobile robots for wireless network relay based on received signal strength," in *Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, (Tokyo), pp. 237–242, Nov. 2013.

[177] R. Magán-Carrión, J. Camacho-Páez, and P. García-Teodoro, "A multiagent self-healing system against security incidents in manets," in *Proceedings of the Practical Applications of Heterogeneous Multi-Agent Systems (PAAMS)*, (Salamanca, Spain), pp. 321–332, June 2014.

[178] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, Nov. 1957.

[179] C. Sammut and G. Webb, *Encyclopedia of Machine Learning*, ch. Leave-One-Out Cross-Validation, pp. 600 – 601. Springer US, 2011.

[180] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[181] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE World Congress on Computational Intelligence - Proceedings of Evolutionary Computation*, (Anchorage), pp. 69–73, May 1998.

[182] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the 7th. International Conference on Evolutionary Programming VII*, (San Diego, California, USA), pp. 591–600, March 1998.

[183] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, pp. 153–181, Springer US: The Kluwer International Series in Engineering and Computer Science., 1996.

[184] X. Hong, M. Gerla, G. Pei, and C. C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of the 2nd. ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 53–60, 1999.

[185] L. Sánchez-Casado, G. Maciá-Fernández, P. García-Teodoro, and R. Magán-Carrión, "A model of data forwarding in MANETs for lightweight detection of malicious packet dropping," *Computer Networks*, vol. 87, pp. 44–58, July 2015.

[186] IDSIA, "Institute dalle molle for artificial intelligence." [Online; Accessed 30-March-2016] `http://www.idsia.ch`.

[187] IDSIA, "IDSIA Swarm Robotics Lab." [Online; Accessed 30-March-2016] `http://robotics.idsia.ch/`.

[188] EPFL (École Polytechnique Fédérale de Lausanne), "Marxbot project." [Online; Accessed 30-March-2016] `http://mobots.epfl.ch/marxbot.html`.

[189] J. Guzzi, A. Giusti, L. Gambardella, G. Theraulaz, and G. Di Caro, "Human-friendly robot navigation in dynamic environments," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, (Karlsruhe), pp. 423–430, May 2013.

[190] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. D. Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[191] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Taipei), pp. 4057–4062, October 2010.

[192] R. Magán-Carrión, J. Camacho, P. García-Teodoro, E. F. Flushing, and G. A. Di Caro, "Dynamical relay node placement solution in MANETs - YouTube." [Demo online; Accessed 30-March-2016] `http://youtu.be/mW1Q_MUFYs4`.

[193] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "Dos attacks in mobile ad hoc networks: A survey," in *Proceedings of the 2012 2nd International Conference on Advanced Computing & Communication Technologies*, ACCT, pp. 535–541, IEEE Computer Society, Jannuary 2012.

[194] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures," *J. Netw. Comput. Appl.*, vol. 35, pp. 867–880, May 2012.

[195] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, "Comparative study of wireless network simulators," in *7th International Conference on Networking*, ICN, pp. 517–523, IEEE Computer Society, April 2008.

[196] A. ur Rehman Khan, S. M. Bilal, and M. Othman, "A performance comparison of open source network simulators for wireless networks," in *IEEE International Conference on Control System, Computing and Engineering*, ICCSCE, pp. 34–38, IEEE Computer Society, Nov. 2012.

[197] A. Kumar, S. Kaushik, R. Sharma, and P. Raj, "Simulators for wireless networks: A comparative study," in *International Conference on Computing Sciences*, ICCS, pp. 338–342, IEEE Computer Society, Sept. 2012.

[198] H. Ehsan and F. Khan, "Malicious AODV: implementation and analysis of routing attacks in MANETs," in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, TrustCom, pp. 1181–1187, IEEE Computer Society, June 2012.

[199] T. Gamer and M. Scharf, "Realistic simulation environments for IP-based networks," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, SIMUTools, pp. 83:1–83:7, ACM, Mar. 2008.

[200] G. Dini and M. Tiloca, "ASF: an attack simulation framework for wireless sensor networks," in *IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications*, WiMob, pp. 203–210, IEEE Computer Society, Oct. 2012.

[201] S. Schmidt, R. Bye, J. Chinnow, K. Bsufka, A. Camtepe, and S. Albayrak, "Application-level Simulation for Network Security," *SIMULATION*, vol. 86, pp. 311–330, May 2010.

[202] 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," in *Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, pp. 321–326, Sep. 2014.

[203] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. Elsevier, 2001.

[204] K. E. Parsopoulos and M. N. Vrahatis, *Particle Swarm Optimization and Intelligence: Advances and Applications*. IGI Global, 2010.

[205] 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.

[206] A. Varga, "OMNeT++ Discrete Event Simulation System." [Online; Accessed 30-March-2016] `http://www.omnetpp.org/`.

Apéndices

Apéndice A

Algoritmo *ekf (element-wise k-fold)*

El pseudocódigo del método *ekf* se muestra en el Algoritmo 2. El núcleo del algoritmo realiza la recuperación de valores faltantes mediante el uso de TSR (*Trimmed Scores Regression*) [121]. A su salida se obtiene la matriz de errores de predicción $\mathbf{E}^A$ (cada valor de la matriz se corresponde con un error $e^A_{n,m}$, siendo n la n-ésima fila y m la m-ésima columna) y el error PRESS (*Prediction Error Sum of Squares*) para cada una las componentes principales $A = 1, ..., A_{max}$ consideradas.

Para entender un poco mejor el algoritmo, este se compone de tres bucles anidados. El bucle más interno itera a través de los grupos variables (para el caso de WSN (*Wireless Sensor Network*), cada variable se corresponde con un sensor y se organizan en columnas). El primer y segundo bucle iteran sobre el número de PC (*Principal*

Algoritmo 2: Pseudocódigo del algoritmo ekf *(element-wise k-fold)*.

para cada *PC* $(A = 1, ..., A_{max})$ **hacer**
  para cada *grupo de observaciones* $(\mathrm{G} = 1, ..., G_{tot})$ **hacer**
    Construir $\mathbf{X}_*$ con los datos de todos los grupos menos G
    Construir $\mathbf{X}_\#$ con los dados de G
    Modelado PCA de $\mathbf{X}_*$, obteniendo $\mathbf{P}^A_*$ y $\mathbf{T}^A_*$
    para cada *grupo de variables* $(\mathrm{H} = 1, ..., H_{tot})$ **hacer**
      Establecer $\mathbf{X}_{\#,\mathrm{H}} = 0$
      $\hat{\mathbf{X}}_\# = TSR(\mathbf{X}_*, \mathbf{X}_\#)$
      Restaurar su valor actual a $\mathbf{X}_{\#,\mathrm{H}}$
      $\mathbf{E}^A_{\mathrm{G,H}} = \mathbf{X}_{\#,\mathrm{H}} - \hat{\mathbf{X}}_{\#,\mathrm{H}}$
    fin
  fin
  Combinar las matrices $\mathbf{E}^A_{\mathrm{G,H}}$ en $\mathbf{E}^A$
  $PRESS^A = \sum_{n=1}^{N} \sum_{m=1}^{M} (e^A_{n,m})^2$
fin

Components) y los grupos de observaciones (para el caso de WSN, cada observación se corresponde con la información recogida por los sensores en un determinado instante de muestreo y se organizan por filas), respectivamente.

Apéndice B

PSO *(Particle Swarm Optimization)*

Los algoritmos de optimización bioinspirados tratan de imitar comportamientos presentes en la naturaleza. Algunos ejemplos de ello es el procedimiento empleado para la búsqueda de comida utilizado por las hormigas, la evolución de las bandadas de aves o el comportamiento de los rebaños de ovejas u otros animales. El paradigma *swarm intelligence* [203] estudia el comportamiento colectivo y las propiedades que se encuentran dentro de las estructuras sociales, así el cómo interactúan sus componentes para conseguir un objetivo común. Se introducen y describen aquí los fundamentes de uno de los más utilizados algoritmos bioinspirados: el algoritmo PSO (*Particle Swarm Optimization*). *Machine learning*, sistemas dinámicos, bioinformática, posicionamiento óptimo de RN (*Relay Node*), entre otros [204], son algunos de los campos o problemas en los que este algoritmo ha sido utilizado satisfactoriamente.

PSO surge bajo los principios del paradigma *swarm intelligence*, donde dada una determinada situación o *swarm* asociada a un determinado problema, se define este en términos de sus posibles soluciones o *particles*. De esta forma, PSO considera una determina población de partículas como posibles soluciones a un problema para, de manera iterativa, evaluar cada una de ellas de acuerdo al objetivo perseguido. Dicha meta se define mediante la definición de una o varias funciones de coste. A medida que se itera PSO va adquiriendo la experiencia recogida por cada partícula, generando nuevas soluciones orientadas a conseguir el objetivo final.

Seguidamente se concretan los anteriores conceptos, un tanto abstractos, siguiendo una descripción matemática formal. Denotemos $A \subset \mathbb{R}^n$ como el espacio de búsqueda objetivo, siendo $f : A \rightarrow Y \subseteq \mathbb{R}$ la función objetivo. La población o *swarm* se puede definir como un conjunto de N partículas (soluciones candidatas)

$$\mathbf{S}^k = \left\{x_1^k, x_2^k, \cdots, x_N^k\right\}, \tag{B.1}$$

cada una de ellas definida como

$$x_i^k = (x_{i1}^k, x_{i2}^k, \cdots, x_{in}^k)^T \in A,\ i = 1, 2, \cdots, N \tag{B.2}$$

donde el índice k representa la k-ésima iteración del algoritmo y n, como se definió anteriormente, la dimensión del espacio $\mathbb{R}^n$.

Con el fin de poder modificar o mover las partículas sobre el espacio de búsqueda, PSO excita cada una de ellas con una determinada, y así denominada, *velocidad*. Este factor se puede representar como sigue:

$$v_i^k = (v_{i1}^k, v_{i2}^k, \cdots, v_{in}^k)^T \in A,\ i = 1, 2, \cdots, N \tag{B.3}$$

La velocidad que se aplica a cada partícula se actualiza en cada iteración del algoritmo y depende, por un lado, de la mejor solución de cada partícula individual p_i^k y, por otro, de la mejor solución global obtenida hasta el momento.

La mejor solución obtenida hasta el momento para cada partícula se puede definir como:

$$p_i^k := \arg\min_k f(x_i^k) \tag{B.4}$$

De manera similar, para obtener la mejor solución de entre todas las partículas basta con elegir la mejor de ellas dentro del conjunto anterior:

$$p_g^k := \arg\min_i p_i^k \tag{B.5}$$

Finalmente, se obtiene una primera versión de PSO [135, 180] como sigue:

$$v_{ij}^{k+1} = v_{ij}^k + c_1 \cdot R_1 \cdot (p_{ij}^k - x_{ij}^k) + c_2 \cdot R_2 \cdot (p_{gj}^k - x_{ij}^k) \tag{B.6}$$

$$x_{ij}^{k+1} = x_{ij}^k + \Delta t \cdot v_{ij}^{k+1} \tag{B.7}$$

donde se muestra el proceso de actualización para la velocidad y las posiciones de las partículas (v_{ij}^{k+1} y x_{ij}^{k+1} respectivamente, con $i \in [1, N]$, $j \in [1, n]$ y $\Delta t = 1$ entre dos iteraciones consecutivas). Para la actualización de la velocidad (B.6) intervienen tres términos. El primero de ellos es la velocidad de la partícula en la iteración previa, (v_{ij}^k). Los otros dos componentes explotan el conocimiento o experiencia adquirido por cada partícula (p_{ij}^k) y por el procedimiento en general (p_{gj}^k). Ambos términos se ponderan con sendos factores: el *cognitive factor*, c_1, y el *social factor*, c_2. A través de la

modificación de ambos factores se condiciona la capacidad exploratoria del algoritmo. De esta manera, el factor cognitivo induce al algoritmo a realizar búsquedas locales, siendo el factor social el encargado de ampliar el espacio de búsqueda. R_1 y R_2 son variables aleatorias que se distribuyen uniformemente dentro del rango $[0, 1]$.

Para evitar el problema de la *swarm explosion* [204], en necesaria la limitación de los incrementos en la velocidad (*velocity clamping*). Al igual que para los coeficientes cognitivo y social del algoritmo, el valor máximo para el incremento de la velocidad (v_{max}) ha de ser cuidadosamente escogido y depende en gran medida del problema que se aborda.

Otro aspecto clave en cualquier algoritmo de optimización es su capacidad de convergencia. La convergencia en PSO se mejora introduciendo un nuevo parámetro w conocido como *inertia coefficient* [181, 182]. El coeficiente de inercia se aplica directamente sobre el valor que obtuvo la velocidad de la partícula en la iteración anterior, decreciendo este con el tiempo. Esta tendencia decreciente promueve la exploración del algoritmo en su etapa inicial, produciéndose la explotación de la solución cerca del final. A partir de estos dos importantes aspectos, se puede redefinir (B.6) como sigue:

$$v_{ij}^{k+1} = w \cdot v_{ij}^{k} + c_1 \cdot R_1 \cdot (p_{ij}^{k} - x_{ij}^{k}) + c_2 \cdot R_2 \cdot (p_{gj}^{k} - x_{ij}^{k}) \tag{B.8}$$

donde se ha de tener en cuenta que su valor se restringe tal que

$$v_{ij}^{k+1} = \begin{cases} v_{max}, & \text{if } v_{ij}^{k+1} > v_{max} \\ -v_{max}, & \text{if } v_{ij}^{k+1} < -v_{max} \end{cases} \tag{B.9}$$

Apéndice C

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 remainder of the appendix is organized as follows. Firstly, in Section C.1, we motivate the importance of strengthening security in ad hoc networks for building survivable systems. After that, in Section C.2 we clarify the objectives of the present work and highlight the main contributions in Section C.3. A discussion about survivability and security aspects in ad hoc networks is introduced in Section C.4. Through Sections C.5 and C.6 we describe two response/tolerant proposals that constitute the main contributions of this work. Finally, in Section C.7 we present the integration of heterogeneous security schemes to achieve global security solutions.

C.1. Motivation

Nowadays, and mainly motivated by the appearance of new technologies, practically any electronic device around the world has communication capabilities. Things and people are in touch through heterogeneous ways making use of several types of networks. An adequate communication and network infrastructure is required to support the diverse and increasing number of services and applications. This is specially in the current communication context, where ad hoc communications and ubiquitous computation necessities are appearing. In this overall situation, the underlying possibilities of ad hoc networks cope with most of the demanding communication necessities [2].

Ad hoc networks have especial characteristics that make them unique [3]. For instance, they do not have a fixed topology or infrastructure such that every node is placed in accordance to the network deployment objectives. Rather, the topology is flexible so that the network nodes must be able to communicate in a cooperative way by multi-hop communications. Most of times, the term "ad hoc" implies not wired communications, so that wireless connections improve the scalability and versatility of this kind of networks.

Depending on the context of use or the adoption of special features, there are several types of ad hoc networks. For instance, the nodes in MANETs (*Mobile Ad hoc NETwork*s) are able to move around the area making them suitable for deployment in scenarios like emergency rescue operations or natural disasters recovery. Other special ad hoc networks are WSNs, which are mainly intended to monitor and sensing a determined area. The gathered sensed data are afterwards used for detection and response in presence of some events. An example of use is that of firefighting scenarios in forestry areas. Also, DTNs (*Delay Tolerant Network*s), sometimes called opportunistic networks, are usually deployed in highly changing environments which principal characteristic is delay tolerance in communications. In similar scenarios, VANETs (*Vehicular Ad hoc NETwork*s) or FANETs (*Flying Ad hoc NETwork*s) are some special cases of ad hoc networks aimed at controlling roads, traffic and vehicles, or flying devices (*e.g.*, drones), respectively. In this case, the so-called WMNs (*Wireless Mesh Network*s) are mainly devised as intermediate networks to provide the Internet access to them.

Despite the big possibilities of ad hoc networks, they also present relevant drawbacks from the perspective of security. Among others, they suffer from: channel vulnerabilities due to the open transmission medium, making eavesdropping attacks feasible; node vulnerabilities, since someone with malicious intentions could access the node in order to damage or tamper it or modify the information managed, compromising the data integrity; and malicious behaviors like *dropping*, *selfish* [6] or *sinkhole* [5]. For the previous reasons, it is very important to provide them with essential security services to guarantee availability, integrity, confidentiality, privacy, authentication and non-repudiation in communications [7]. In that sense, like in any other networks and systems, three are the so-called defense lines to be implemented to strengthen security: prevention, detection and response/tolerance. Although there are many proposals in the literature addressing security issues in ad hoc networks [9], most of them are strictly related to preventive solutions based on the use of cryptographic approaches. Sometimes, nevertheless, preventive mechanisms are eluded specially when the attacker is part of the network (insider attacks). That way, detection schemes should take place in order to be aware of the malicious behaviors. By themselves, the detection procedures do not mitigate the attack, so that after detection a subsequent response/tolerance should be launched to solve the attack impact and restore the services provided by the network.

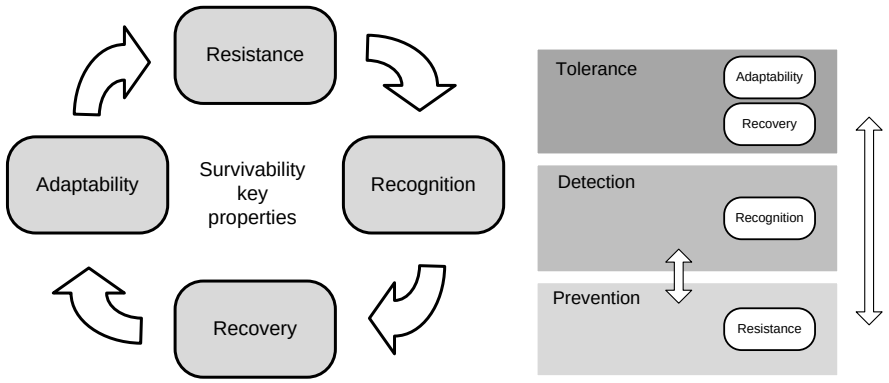


Figure C.1: Ad hoc networks survivability key properties (a), and their relationship with tipical defense lines (b).

Acting together, these defense lines will make the network and services supplied more robust in the presence of attacks. From this point of view, every network or system should be intended to be a *survivable system*. This way, a survivable system may be defined as that presenting "*the ability to fulfill its mission, in a timely manner, in the presence of attacks, failures or accidents*" [10]. As a consequence, attaining survivable ad hoc networks necessarily implies the design of techniques, systems or methodologies such that (see Figure C.1): (*i*) they give resistance (prevention), recognition (detection), recovery (response) and adaptability (tolerance) capabilities, (*ii*) they consider several additional survivability requirements, and (*iii*) they deal with different protocol layers.

The lack of proposals dealing with survivable ad hoc networks addressing the previous requirements in a global way, makes this a challenging research field. Most of the current available solutions are specifically intended to solve a reduced number of threats (just one, in most of the cases). Moreover, they are solely centered on one defense line or focused on just one network layer [12]. In regards to the survivability, ad hoc networks have implicit properties like self-management, self-organization, decentralization, scalability, etc., included as survivability requirements as shown in Figure C.2 [10]. However, some others should be explicitly devised to provide self-diagnosing, self-healing or self-adaptation at different network layers.

Mainly aimed at the provision of certain survivability aspects in ad hoc networks, the present dissertation work is intended to implement mechanisms, heuristics and methodologies principally focused on reactive, tolerant and adaptable schemes. The studies and the consequent proposals developed deal with threats in several layers in order to pursuit more global objectives (*e.g.*, communication, routing and connectivity services) instead of specific ones, thus making more general purpose solutions.

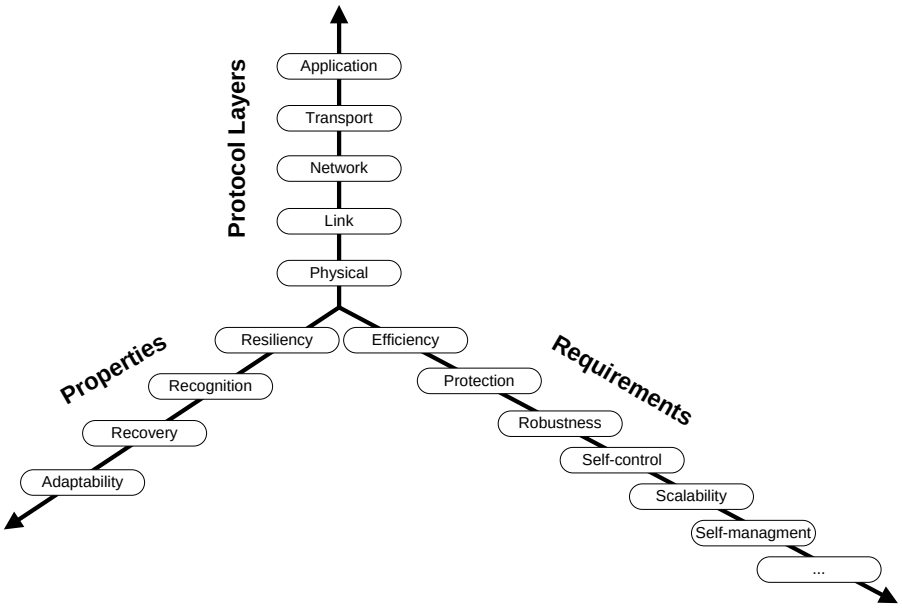


Figure C.2: Survivability dimensions.

C.2. Objectives & methodology

As previously stated, the main goal of this work is to cope with security vulnerabilities in ad hoc networks. Our main interest is to provide survivability capabilities to networks through the implementation of new reactive and tolerant security solutions adding relevant survivability requirements as self-diagnosing, self-healing or self-adaptation to mitigate the effects caused by attacks.

Guided by the previous main goal, the first step is to perform a study of the specialized literature searching for: (*i*) current and relevant threats in ad hoc networks, and (*ii*) solutions dealing with survivable systems through the proposal of response and tolerant solutions.

The next step is to focus on the design and development of useful techniques and solutions covering some research holes and security weaknesses in ad hoc networks. That is the main part of the work. Since the ad hoc network concept includes diverse kinds of networks, we center our attention on two of them, with significant different goals: WSNs and MANETs. Different response and tolerance schemes have been developed for them, which are evaluated by means of both simulated and real scenarios. The results obtained show the good performance of the proposals.

Although the solutions implemented here contribute to strengthen the network survivability against several attacks, more ambitious and global security solutions must be provided. For that, we devise and implement a framework to integrate several

security solutions related with different defense lines. With this aim, we design the necessary infrastructure to support, on the one hand, the mandatory communication between defense lines, and on the other, a procedure to implement and execute some attacks to test the operation and performance of the overall integration from a security perspective.

In the following we can see the tasks carried out along the entire work:

I. *Study of the survivability and security in ad hoc networks, intended to*

   a) Establish the dependencies and relationship between security and survivability, showing the relevance of leading secure solutions from that point of view.

   b) Study of the most relevant threats in ad hoc networks.

   c) Perform a detailed analysis of the solutions available in the specialized literature in the field of security response and tolerant defenses against these threats, establishing a novel classification of the works.

II. *Proposal and development of response and tolerant security solutions in ad hoc networks, aimed at*

   a) Proposing and developing new reactive and tolerant schemes to fight against most relevant threats in ad hoc networks, specially in WSN and MANET environments.

   b) Evaluating the correct behavior of the proposals in two ways:

      - By means of simulation, leading us to corroborate the feasibility of the approaches at first instance.
      - By means of the deployment of the solutions in real environments, corroborating their practical application which is not commonly addressed in similar papers in the literature.

III. *Integration of security solutions for survivable networks, that is*

   a) To devise and design a feasible architecture for implementing and testing attacks in ad hoc networks.

   b) To address the problem of how the different defense lines should interoperate by means of a review of the specialized literature.

   c) To devise, design and implement an integral framework capable of integrating several defense lines and their corresponding interactions.

   d) To test and evaluate the entire system by means of implementing and deploying some attacks.

C.3. Main contributions

From the above, the main contributions of this thesis are summarized as follows:

1. Study of the main security threats for ad hoc networks and solutions proposed by the research community for thwarting them, introducing a classification for response mechanisms.
2. Development and evaluation of two novel reaction solutions for fighting against security threats in ad hoc networks. The first approach consists of a multivariate missing data imputation solution for data integrity in WSNs.
3. The second proposal provides a centralized relay node placement optimization scheme aimed at maintaining, recovering or even improving the connectivity and throughput in MANETs.
4. Development of a novel framework to integrate and communicate heterogeneous security solutions at different defense lines, which relies on a flexible, scalable and versatile architecture.

C.3.1. Publications

In the following, we indicate the publications related to the main topics of the present thesis:

International Journals

1. **R. Magán-Carrión**, J. Camacho, P. García-Teodoro, E. F. Flushing and Gianni A. Di Caro. "Dynamical Relay Node placement Solution (DRNS) for MANETs," Submitted to *Ad Hoc Networks (Elsevier)*, 39 pages, 2016.
2. J. Camacho, **R. Magán-Carrión**, P. García-Teodoro and J. J. Treinen. "Network-metrics: Multivariate Big Data Analysis in the Context of the Internet," Submitted to *J. Chemometrics (Wiley)*, 45 pages, February 2016.
3. **R. Magán-Carrión**, R.A. Rodríquez-Gómez, J. Camacho and P. García-Teodoro. "Optimal Relay Placement in Multi-hop Wireless," Accepted in *Ad Hoc Networks (Elsevier)*, 34 pages, March 2016.
4. L. Sánchez-Casado, G. Maciá-Fernández, P. García-Teodoro and **R. Magán-Carrión**. "A model of data forwarding in MANETs for lightweight detection of malicious packet dropping," *Computer Networks (Elsevier)*, vol. 87, pp. 44–58, July 2015.

5. **R. Magán-Carrión**, J. Camacho and P. García-Teodoro. "Multivariate Statistical Approach for Anomaly Detection and Lost Data Recovery in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks (Hindawi)*, vol. 2015, pp. 1–20, May 2015.

6. **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.

International Conferences

7. **R. Magán-Carrión**, J. Camacho and P. García-Teodoro, E. F. Flushing and Gianni A. Di Caro. "DRNS: Dynamical Relay Node placement Solution," Accepted in *Demonstration in Advances on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, June 2016.

8. **R. Magán-Carrión**, J. Camacho, P. García-Teodoro, E. F. Flushing and Gianni A. Di Caro. "Dynamical Relay Node placement Solution in MANETs," *Demonstration in 3rd International Black Sea Conference on Communications and Networking (BlackSeaComm)*, May 2015. [Demo online; Accessed 15-December-2015]

9. **R. Magán-Carrión**, J. Camacho-Páez and P. García-Teodoro. "A Multiagent Self-healing System against Security Incidents in MANETs," *Highlights of Practical Applications of Heterogeneous Multi-Agent Systems (PAAMS)*, vol. 430, pp. 321–332, June 2014.

10. 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, Sept. 2013.

11. **R. Magán-Carrión**, J. Camacho-Páez and P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents," *Advances on Practical Applications of Agents and Multi-Agent Systems (PAAMS)* , vol. 7879, pp. 182–191, May 2013.

12. **R. Magán-Carrión**, J. Camacho-Páez and P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents: A Practical Vision," *Demonstration in Advances on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, vol. 7879, pp. 308–311, May 2013.

Book Chapters

13. 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, April 2014.

National Conferences

14. 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," *Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, pp. 321-326, Sept., 2014.

15. **R. Magán-Carrión**, J. Camacho-Páez and P. García-Teodoro. "A Security Response Approach Based on the Deployment of Mobile Agents: Limitations and Improvements," *XI Jornadas de Ingeniería Telemática (JITEL)*, pp. 445–452, October 2013.

16. 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, October 2013.

17. **R. Magán-Carrión**, J. Camacho Páez and P. García-Teodoro. "Supervivencia en redes de sensores mediante técnicas multivariantes," *12th Reunión Española sobre Criptología y Seguridad de la Información (RECSI*) pp. 315–320, Sept. 2012.

C.4. Survivability & security aspects in ad hoc networks

[Related publication: 13]

Survivability and security are closely related. In fact, as stated in Section C.1, taking into account security aspects is mandatory for achieving survivable systems. For that, any "secure" system should contemplate at least three security defense lines: prevention, detection and response/tolerance. First, to avoid threats becoming effective attacks, prevention schemes should be put into action. However, they do not guarantee the absence of attacks, as attackers can bypass them. This way, defense mechanisms to monitor the network and to detect attacks are also needed. In case of attack detection, response actions should be subsequently launched in order to solve the incidents. The previous security concerns are key aspects to provide survivable systems [10]. Nevertheless, most of the solutions provided by the research community are just circumscribed to a determined defense line. Moreover, most of solutions only focus on very specific attacks.

Due to their particular characteristics, ad hoc networks suffer from special security threats. For instance, their wireless transmission medium make them prone to *eavesdropping* attacks. Additionally, the lack of a centralized management make feasible impersonation attacks like *sybil*, and their inherent multi-hop routing philosophy contributes to the occurrence of *blackhole*, *synkhole* and *wormhole* attacks among others [16, 17, 205].

In order to mitigate the previous inherent threats an extensive work has been carried out by the research community. Nowadays, specially in ad hoc networks, most of proposals are focused on threats related to one of the most harmful groups of attacks: *packet dropping attacks* [5]. In this context, most of works deal with on prevention and detection solutions, while response or reaction related solutions are less commonly addressed. Additionally, almost all of the response solutions could be circumscribed to those intended to isolate or exclude the nodes acting as malicious ones, though such a classification is not clear. That way, publication 13 provides a detailed review of the current response solutions in the literature. Moreover, it introduces a novel classification of such solutions in three groups: *node exclusion*, *node exclusion and announcement* and *node isolation*. In Figure C.3 we can see the organization and where each work can be classified into.

In this overall context, more efforts must to be done to develop security solutions to cope with some survivability requirements. Furthermore, the solutions should pursuit global objectives to fight against several attacks.

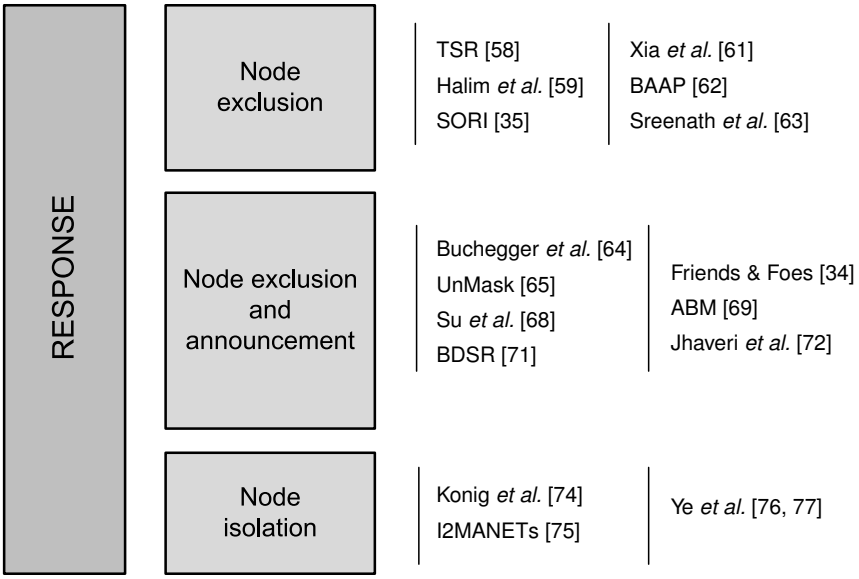


Figure C.3: Response solutions classification.

C.5. Missing data imputation in WSNs

[Related publications: 5, 6 and 17]

In the overall aforementioned context, the central part of the thesis is intended to devise, develop and test two novel response/tolerant proposals for ad hoc networks. In particular, two are the types of the specific environments considered: WSNs and MANETs. The solutions discussed are valid to mitigate a number of attacks like *tampering*, *dropping* (in its different variants) and *route poisoning* (*sinkhole*, *wormhole*, etc.). It is also worth noting that the developments are validated both in simulation and by means of real scenarios.

A WSN is a (structured or unstructured) group of hundreds or even thousands of sensor devices intended to monitor a given area by measuring one or more physical variables [85]. There are two principal applications of WSNs: monitoring and tracking. In both cases WSNs can be applied in various fields like military, medical and/or industrial [86]. Deploying monitoring mechanisms in WSNs to strengthen the services provided is encouraged. This is especially relevant in hostile environments like military actions, crisis management and disaster detection and recovery, where data loss or data modification could involve disastrous consequences. The normal operation of a WSN is vulnerable to malicious data modification attacks, such as the so-called *data tampering*, *environmental tampering* or *tampering* attack [89][90].

Along this section, we assess the application of multivariate analysis techniques for WSNs monitoring and data recovering in critical scenarios like firefighting in a forestry area. Multivariate techniques fit well when there exists a high temporal and

spatial correlation between the variables considered, which is a common feature in WSNs. The monitoring scheme is aimed at finding anomalous events. Subsequently, the diagnosis of these anomalies can show whether the anomaly is due to an actual measurement or due to data loss/modification. In the event of data loss/modification, the recovery scheme will be responsible for the estimation of the missing data.

To monitor and detect anomalies in the system behavior, MSPC (*Multivariate Statistical Process Control*) techniques based on PCA (*Principal Component Analysis*) [91][92] and PLS (*Partial Least Squares*) [93][94] multivariate models are used. To recover lost data, TSR method [95][96] using both PCA- and PLS-based models (TSR-PCA and TSR-PLS) is employed.

To the best of our knowledge, this is the first time that PCA, PLS and TSR are used in the context of WSNs. There exist several works that address missing data imputation-based solutions in the literature [100–102], but their application in WSNs is still limited. Moreover, most of them are focused on anomaly detection [108][109] and avoid the subsequent recovery procedure.

Although a multivariate-based monitoring and detection solution will be introduced in the following, the main goal of this section is to evaluate the performance of the TSR method for recovering missing data when a data tampering attack is taking place. In fact, the detection module is just intended to provide a complete multivariate-based proposal, although alternative detection schemes could be considered. Furthermore, we will also assess the impact of the data arrangement methodology [97–99] to build the appropriate multivariate model depending on the final application of the system, as well as the relevant role of the underlying routing algorithm over the imputation efficacy.

C.5.1. Multivariate analysis

This section introduces the fundamentals of the multivariate statistical analysis through the description of the main specific techniques used. They are PCA, PLS and TSR. The first one is mainly used in monitoring and anomaly detection. In this case, PLS can also be used in that context though it fits better in estimation problems due to its own nature as prediction model. Finally, the last one is used for missing data imputation. A more detailed explanation about the previous techniques as well as their application feasibility in WSNs scenarios for monitoring, anomaly detection and missing data recovery are addressed in publication 5.

PCA (*Principal Component Analysis*)

PCA transforms the original set of variables into a new and reduced set of uncorrelated variables. PCA identifies a number of linear combinations of the original variables in a data set **X**, the so-called PCs, containing most of its relevant information (variability). This is a change of variables from the original variables in the **X** space to the PCs subspace. If **X** is a data matrix with J variables associated with a given phenomenon and I observations of each variable, PCA reduces its dimension from J variables to A PCs by finding the A-dimensional latent subspace of the most variability captured.

PCA follows the next equation:

$$\mathbf{X} = \mathbf{T}_A \cdot \mathbf{P}_A^T + \mathbf{E}_A \tag{C.1}$$

where $\mathbf{P}_A$ is the $J \times A$ loading matrix, $\mathbf{T}_A$ is the $I \times A$ score matrix and $\mathbf{E}_A$ is the $I \times J$ residual matrix. The maximum variance directions are obtained from the eigenvectors of $\mathbf{X}^T \cdot \mathbf{X}$, and they are ordered as the columns of $\mathbf{P}_A$ by explained variance. The rows of $\mathbf{T}_A$ are the projections of the original I observations in the new latent sub-space. $\mathbf{E}_A$ is the matrix that contains the residual error, and it plays a crucial role in anomaly detection, as shown afterwards. The projection (score) on the PCA subspace of a new observation is obtained as follows:

$$\mathbf{t}_{new} = \mathbf{x}_{new} \cdot \mathbf{P}_A \tag{C.2}$$

where $\mathbf{x}_{new}$ is a $1 \times J$ vector representing a new object and $\mathbf{t}_{new}$ is a $1 \times A$ vector representing its projection to the latent subspace.

PLS (*Partial Least Squares*)

Another relevant problem in multivariate analysis is data regression, where two data sets are involved: **X** and **Y**, where **X** is the $I \times J$ measurement matrix used to predict **Y** ($I \times M$).

To predict **Y**, a model **B** containing the regression relationship between both data sets **X** and **Y** is first estimated. New **Y** values can then be predicted from the new **X** measures. The linear regression problem is defined by the expression

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{B} + \mathbf{F} \tag{C.3}$$

The least squares solution for Eq. (C.3) is

$$\hat{\mathbf{B}} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{Y} \tag{C.4}$$

This solution cannot be computed if matrix $\mathbf{X}^T \cdot \mathbf{X}$ is singular. It is also highly unstable when a high correlation exists among variables in $\mathbf{X}$. To overcome this limitation, the PLS method applies the latent PCA subspace idea to the regression problem. In this case, the variables in $\mathbf{X}$ are transformed into a reduced set of latent variables that maximize the covariance between $\mathbf{X}$ and $\mathbf{Y}$.

The partial linear regression problem between normalized matrices $\mathbf{X}$ and $\mathbf{Y}$ can be stated as:

$$\mathbf{X} = \mathbf{T}_A \cdot \mathbf{P}_A^T + \mathbf{E}_A \tag{C.5}$$

$$\mathbf{Y} = \mathbf{T}_A \cdot \mathbf{Q}_A^T + \mathbf{F}_A \tag{C.6}$$

where $\mathbf{T}_A$ is the $I \times A$ score matrix, $\mathbf{P}_A$ and $\mathbf{Q}_A$ are the $J \times A$ and $M \times A$ loading matrices, and $\mathbf{E}_A$ and $\mathbf{F}_A$ are the $I \times J$ and $I \times M$ residual matrices of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

The regression coefficients of the PLS model are finally established as

$$\hat{\mathbf{B}}_{PLS} = \mathbf{W} \cdot (\mathbf{P}^T \cdot \mathbf{W})^{-1} \cdot \mathbf{Q}^T \tag{C.7}$$

where $\mathbf{W}$ is a $J \times A$ matrix of weights, such that $\mathbf{T} = \mathbf{X} \cdot \mathbf{W} \cdot (\mathbf{P}^T \cdot \mathbf{W})^{-1}$. A PLS model is thus represented by matrices $\mathbf{P}$, $\mathbf{W}$ and $\mathbf{Q}$.

Finally, a new observation with the PLS model is estimated as

$$\hat{\mathbf{y}}_{new} = \mathbf{x}_{new} \cdot \hat{\mathbf{B}}_{PLS} \tag{C.8}$$

where $\mathbf{x}_{new}$ is a $1 \times J$ vector representing a new object and $\hat{\mathbf{y}}_{new}$ is a $1 \times M$ vector representing the estimation of the output variables.

TSR (*Trimmed Scores Regression*)

TSR is a regression method that presents a good trade-off between simplicity and estimation performance [96]. TSR estimates the value of the scores from the trimmed scores, *i.e.*, the scores obtained by filling the missing values with zeros. For data centered before PCA, this is equivalent to using the average value of a variable to give an initial estimation of its missing values.

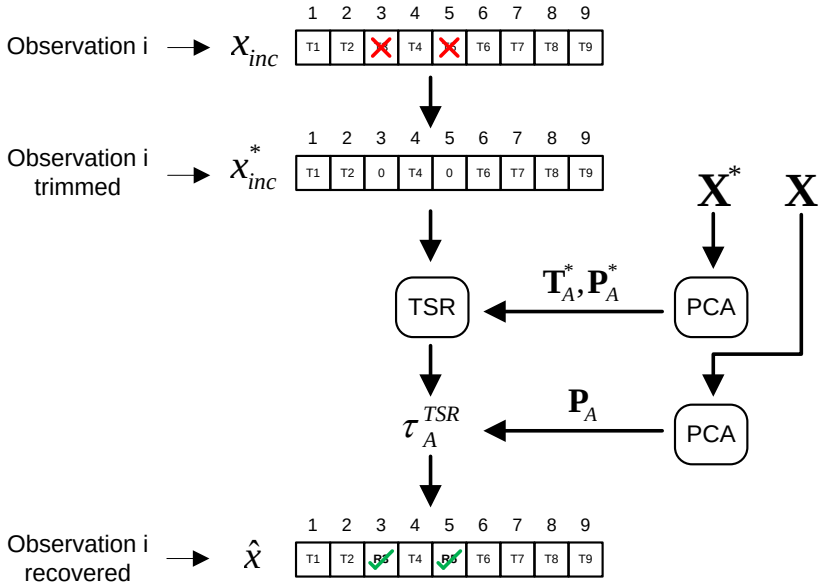


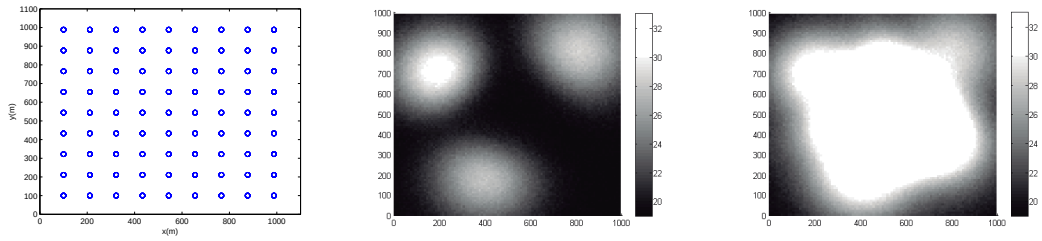
Figure C.4: Illustration of the main steps implied in the missing data recovery procedure.

The recovery procedure is activated when an altered observation is detected. A missing data situation is determined through a previously established monitoring system. Figure C.4 shows a graphical illustration of how the imputation method works. Note that this procedure is based on the use of PCA models, though considering PLS models does not change the main methodology. Although the process is self-explanatory (T3 and T5 values are missed and thus they are recovered and substituted by R3 and R5, respectively), two are the main aspects to remark here: (*i*) the imputation method only considers the available information to estimate the scores, and (*ii*) the system is able to get an estimation of the original observation by applying the complete calibration PCA model.

C.5.2. Simulation scenario

Due to the lack of WSN simulators for physical measurements (*e.g.*, the temperature gathered from each sensor in a determined area), we developed a specific simulator based on Matlab 2009b. Figure C.5(a) shows the sensor deployment around the area. Figure C.5(b) shows the temperature distribution considering three normal temperature focuses intended to emulate cooler and heater areas. Finally, Figure C.5(c) illustrates a fire situation where the fire has a central focus covering more than a half of the total area.

We assume a 1000 m×1000 m square forestry area where 81 (9×9) sensors are regularly distributed, *i.e.*, each sensor is located ~100 m away from its neighbors



(a) (b) (c)

Figure C.5: Simulation scenario: (a) sensor locations, (b) temperature map under normal conditions, (c) temperature map with a fire focus.

(Figure C.5(a)). Every sensor gathers the ambient temperature each sampling time and sends the measurements to a CU (*Central Unit*). The proposed deployment for the measurements is inspired in a real system provided by the Libelium company[1].

The simulation tool is first employed to generate a data set used to calibrate the PCA model (hereafter, CAL (*CALibration dataset*)). The data matrix $\mathbf{X}$ contains 100 observations of 81 variables (the temperatures obtained by each sensor) under normal temperature conditions, *i.e.*, without a fire situation. A situation in which a fire focus evolves over time is then simulated (hereafter, FIR (*FIRe dataset*)). These data sets are used to study the detection capabilities of our anomaly detection system.

Routing strategies

Before carrying out experimentation in the terms previously discussed, an additional issue to be considered is that of routing strategies in WSNs. In the context of security, and specially for *data tampering* attacks, the selection of the routing algorithm can affect the network performance. For example, in a multi-hop routing scheme the affected data due to a compromised sensor will vary depending on the location of that sensor and the overall information forwarded through it. This is a relevant issue. Think for instance in the context of firefighting, where tampering attacks may deceive the fire brigade, leading to potential human casualties.

To corroborate the previous hypothesis regarding the interplay between routing and the influence of tampering in a WSN, we analyze several static routing strategies in order to evaluate their impact on the detection and recovery results. They are [85]: MCFA (*Minimum Cost Forwarding Algorithm*), which establishes the routes towards the CU; and LEACH (*Low Energy Adaptive Clustering Hierarchy*), a known data aggregation algorithm mainly devised to preserve energy network resources. Depending

[1] http://www.libelium.com/wireless_sensor_networks_to_detec_forest_fires/

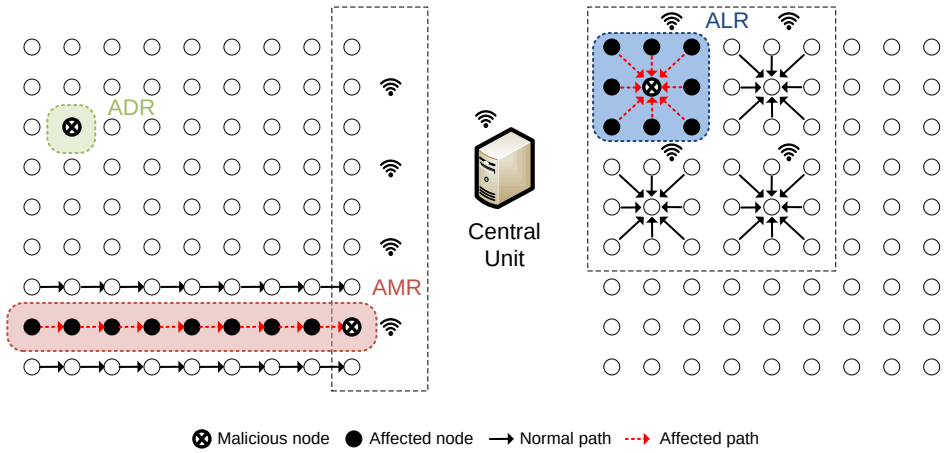


Figure C.6: Different data tampering scenarios.

on the routing algorithm selected, three are the scenarios under evaluation: ADR (*Attack on Direct Routing*), where each sensor sends the gathered information directly to the CU; AMR (*Attack on MCFA Routing*), where MCFA routing is used; and ALR (*Attack on LEACH Routing*), based on LEACH routing approach. Depending on which sensor is compromised by the data tampering attack and the chosen routing strategy, the malicious impact could be increased. Figure C.6 shows the worst cases for each scenario considered. For instance, in the ALR case the location of the malicious node (circle within a cross) is the CH (*Cluster Head*), where all sensor data (filled black circle) are in turn compromised. For each of the previous scenarios a data set is generated, hereafter called ATT (*ATTack test dataset*).

From CAL, FIR and ATT datasets we will evaluate the system performance in terms of the ability for determining occurrence of the data tampering attacks and recovering the modified or faulty data.

C.5.3. Results: Data arrangement

In what follows, we provide the experimental results obtained by our data recovery approach. For that, the problem of data arrangement arises.

This topic has been addressed in many works and applications. Among others, statistic monitoring [97], process control [98] or image processing [99] are some examples of different fields in which the organization of the data has a high relevance. In fact, most of times, data organization is closely related to the final application it is intended for [125]. This way, we will have to organize the data according to two main objectives: (*i*) monitoring and anomaly detection, and (*ii*) missing data recovery.

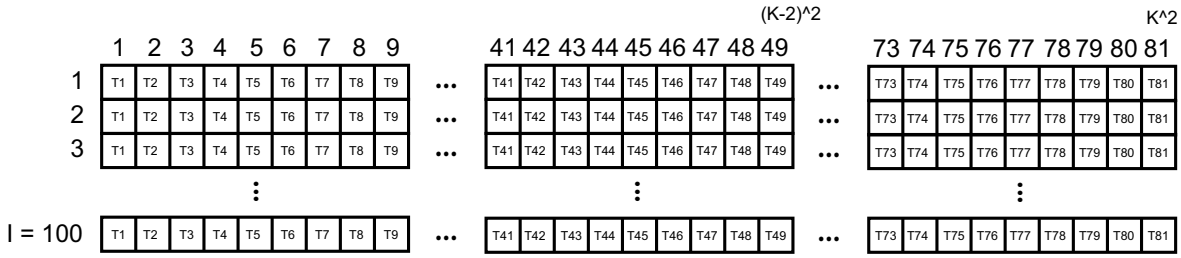


Figure C.7: Global model **X** from the calibration data, conformed by $I = 100$ observations of $J = 81$ variables each. K denotes the number of sensors places on each side of the regular deployment. T_j, with $j = 1, 2, \cdots, J$, stands for the whole set of measurements from sensor j.

For that, a *global modeling* is introduced and evaluated for both of the previously mentioned objectives. Afterwards, we will see the unsuitability of such model when it is used for missing data imputation. This way, a *local modeling* will also be introduced and analyzed.

Global modeling

We define a *global model* as a PCA model calibrated from the data gathered from the sensors. These data are arranged in a matrix form as follows: those data corresponding to each single sensor are arranged as a column, and those corresponding to each single measurement interval as a row. Thus, the matrix of data used to calibrate our PCA model, **X**, contains J variables, with J the number of available sensors, and I observations, with I the number of sampling times.

Figure C.7 depicts the data arrangement for an hypothetical area network with 81 sensors in total, and 100 time observations of each of them. In this case, the corresponding model refers to a matrix **X** of dimension 100×81.

Monitoring and anomaly detection

Despite monitoring and anomaly detection is not the focus of the present work, it is an important part to build global security solutions. Thus, we briefly explain this module, developed by using MSPC-based techniques. More details about this issue can be found in publication 5.

We illustrate the proposed approach for monitoring WSN anomalies by making use of the PLS-toolbox for Matlab [126]. By comparing the PCA model obtained from the calibration dataset (CAL) to the new observations under monitoring (*i.e.*, the test data set FIR or ATT), anomalous behaviors can be detected in the environment. This detection is performed by means of the use of monitoring graphics such as those

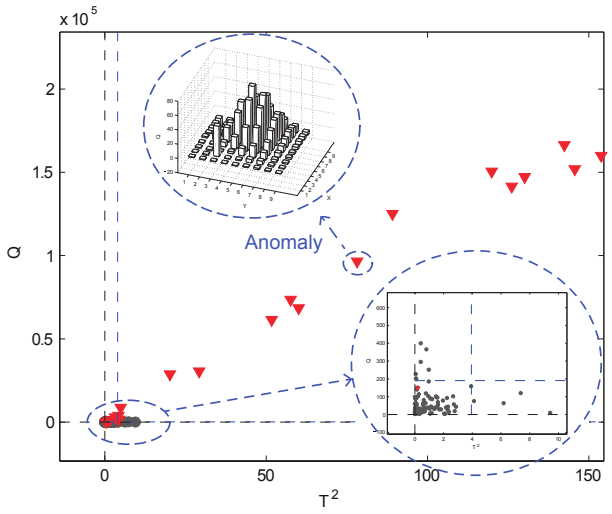


Figure C.8: Monitoring graphics and Q contribution plots for anomaly detection. Monitoring graphic representing the initial calibration data (dark circles) and control limits (dashed lines), from which anomalies are detected (inverted triangles). At the top left we can see a Q contribution plot detailing an anomalous specific observation.

presented in Figure C.8. Such figure shows the T^2 statistic as the abscissa and the Q statistic as the ordinate [116]. A point in the plot represents one sampling time of the WSN. The monitoring chart points out the sampling times corresponding to the fire evolution (inverted triangles) in the ATT data set as anomalous events, because they differ from those corresponding to normal conditions (dark circles) in the CAL data set. The monitoring system can detect the fire situation (inverted triangles) from the beginning of its evolution.

It is also important to remark at this point the fact that the monitoring system is not capable to automatically distinguish between actual anomalies, such as fire events, and false alarms caused by a potential tampering attack or a sensor malfunction. That way, we rely on the existence of a human supervisor who distinguishes between real anomalies and false alarms after an alarm is triggered. However, automatic detection proposals could be addressed instead (see publication 5). Figure C.9(a) shows the typical pattern for the Q contribution only in the presence of fire, while Figure C.9(b) shows, as we shall detail below, the pattern obtained under the previously mentioned attack scenario AMR in the same fire scenario. The tampering attacks are shown as sharp artifacts which depend on the routing scheme and which are clearly different from the smooth contribution of a true fire.

Whatever the detection method, either manual or automatic, used to determine the occurrence of false alarms due to tampering or malfunction, a missing data recovery process is afterwards executed to solve the situation and recover the affected data. This process is discussed below.

Missing data recovery

In order to perform the adequate response once an attack is detected, we will employ the TSR procedure. Aimed at assessing the viability of the proposed method, we evaluate TSR under the previously mentioned attack scenarios: ADR, AMR and ALR. In particular, Figure C.9(b)) shows the Q contribution under the AMR attack, while Figure C.9(c) depicts the result of the recovery process for AMR. Now, the Q contribution is smoother than the original case shown in Figure C.9(b). Despite the evident improvement reached, the results are far from being optimum: the Q contribution for distant sensors from the fire is lower than that for closer ones. This is motivated by the the fact that CAL was obtained under no fire conditions, while data tampering experiments are performed under fire circumstances. That way, the recovery from data tampering attacks in distant locations from the fire is more effective.

Beyond the visual-based results for AMR, Table C.1 shows numerical results of the recovery process for all the attack scenarios using the MSE (*Mean Squared Error*) parameter between the original and the restored data. To avoid the aforementioned sensor locations influence on the results, we calculate the average MSE value for all the sensors in the 9×9 network. It is worth noting that TSR works better as the number of available valid sensor values (correlated and not tampered data) is higher. For this reason, the best results (lower error) are for ADR since just one sensor is tampered. On the contrary, ALR presents the worst case. In the case of AMR, every sensor has at least two sensors above and below it with valid data, so that it presents intermediate recovery results.

In summary, the routing algorithm is a key aspect to consider in this problem. Although an aggregation algorithm is a good choice from an energetic perspective, it may be not from a security viewpoint.

Data tampering	**MSE (TSR-PCA)**
ADR	1900.8
AMR	2472
ALR	4391.6

Table C.1: MSE comparison for different tampering attacks using TSR-PCA as missing data imputation method.

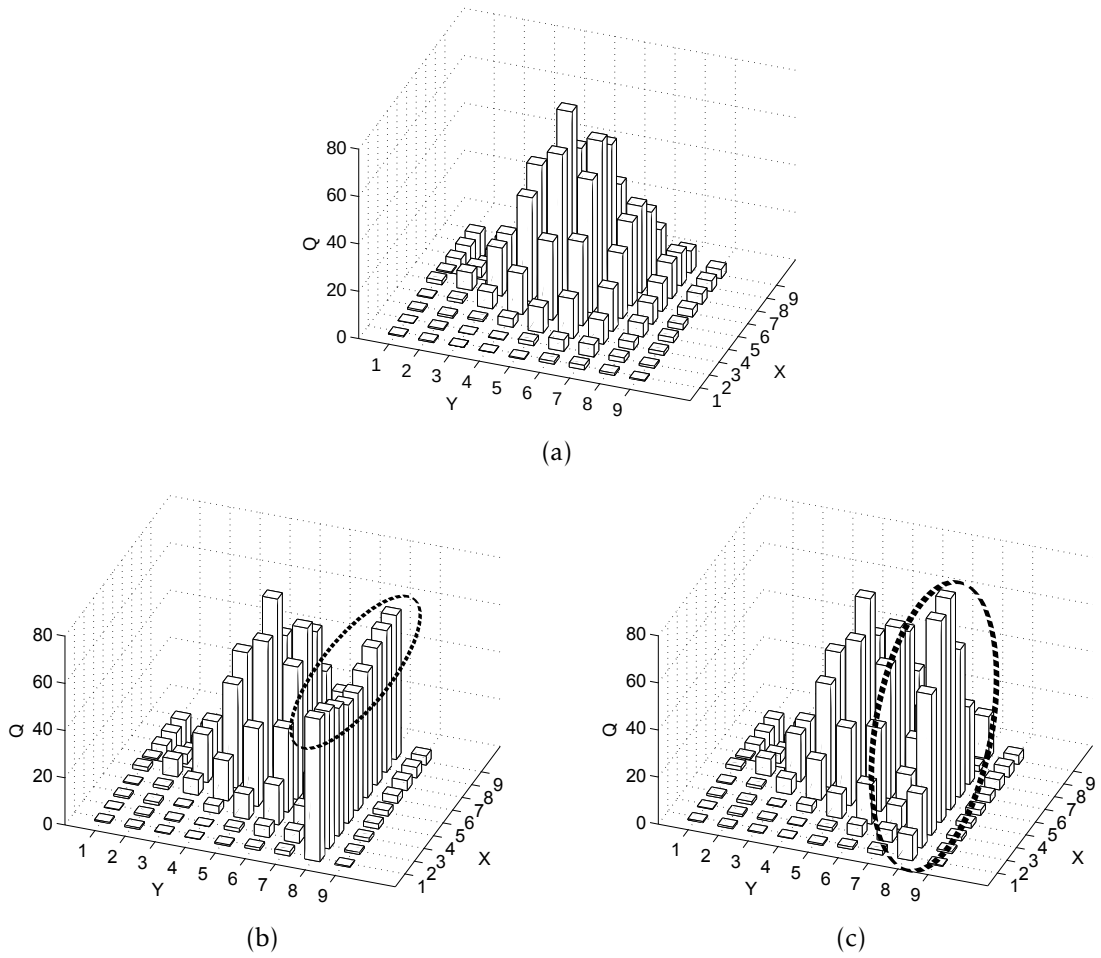


Figure C.9: AMR simulated data tampering scenario. The Q profile of the observation with fire and without attacks is shown in subfigure (a). At the bottom, the corresponding Q contribution under fire influence is shown in subfigure (b), while the reccovery result is depicted in subfigure (c). The tampered sensors are properly marked.

C.5.4. Dynamic global models & routing to improve missing data imputation

From the previous evaluation and results, we can conclude that global models are not efficient for data recovering. This is closely related to how TSR works: the higher the data correlation the better the missing data imputation performance. Global models lead TSR to impute faulty sensor data from uncorrelated values. Consequently, a recovered sensor value could be even worst than its tampered value.

In order to solve that evident limitation we have to increase the data correlation in the model. A first approach is to consider into the model the implicit WSN temporal information by building dynamic models. Nevertheless, as we will discuss later, these

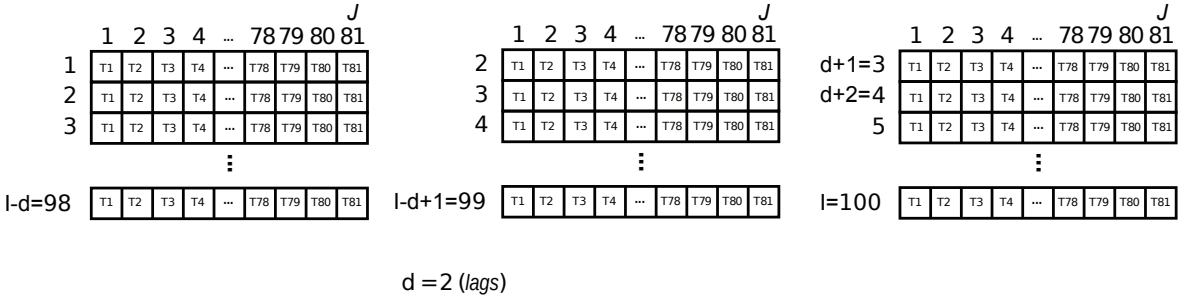


Figure C.10: Dynamic global model (X_d) with $I-d$ observations of $J\times(d+1)$ variables. J and I represent the number of original variables and observations respectively, while d denotes the number of temporal lags used. T_j means the measurements gathered by a specific sensor j, with $j = 1, 2, \cdots, J$.

models are not efficient by themselves and have to be used together with dynamic routing strategies.

Dynamic global modeling

To take advantage of the dynamic nature of the data gathered from WSNs, DPCA (*Dynamic PCA*) modeling [113] is used. DPCA extends the original $\mathbf{X}$ of PCA modeling by adding variable observations from past sampling times, called *lags*. From here onwards, the corresponding model will be called *dynamic global model*.

Figure C.10 depicts the data arrangement for DPCA from the original data matrix $\mathbf{X}$ used in the original PCA version. In this case, the corresponding model is fitted from matrix $\mathbf{X_d}$ which dimension depends on the temporal lags used. Generalizing, the dimension of $\mathbf{X_d}$ is $(I-d)\times J\cdot(d+1)$, with I the number of observations, J the number of the system variables, and d the time lags considered.

Dynamic routing strategies

When a sensor measurement is lost due to a tampering attack, dynamic models make the most of past (lagged) measurements for its estimation. When static routing is used, the attack affects the same sensors over time. This prevents the improvement of the estimation thanks to the introduction of dynamic information in the model. However, coupling dynamic models with variable routing, where data are relayed through different paths over time, can be effective. For this reason, we propose several dynamic routing strategies inspired on MCFA routing to assess the relevance of routes variability on the data recovery performance. All the proposals differ in the rule to decide which of the n closest sensor nodes to a given one will be selected as the next hop in the path. In our case $n = 3$, meaning that one of the three sensor

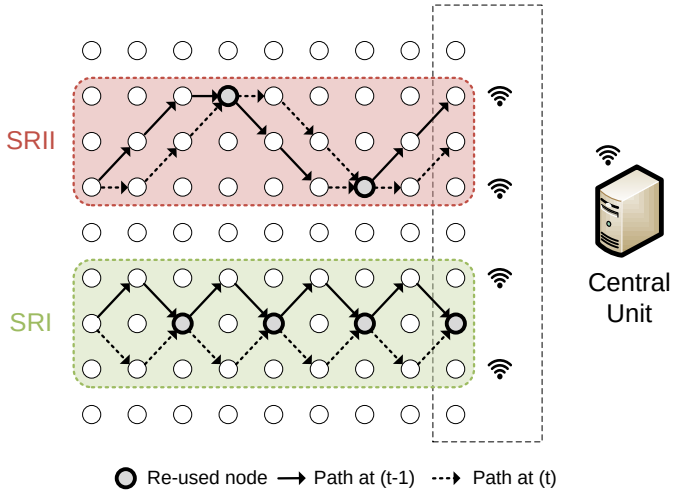


Figure C.11: SR routing variants. On the bottom, the first switching-based routing strategy, SRI. On the top, the second switching-based routing strategy, SRII. From left to right, SRII re-uses only two nodes while SRI does four in two consecutive sampling times, $t-1$ and t.

nodes just to the right of a given one could be selected as the next hop, since the CU is also to the right of the sensor deployment. They are as follows:

- RR (*Random Routing*): Random selection, so that each one of the next available n nodes has the same probability to be selected ($\frac{1}{n}$).
- DRR (*Differential Random Routing*): Random selection among the next $n-1$ nodes which were not selected in the previous sampling time.
- SR (*Switching-based Routing*): Makes a selection following a deterministic pattern intended to vary the routes in time as much as possible.

Additionally to RR and DRR strategies, we propose and evaluate three variants for SR: SRI, SRII and SRII. The first one alternatively switches between the immediate sensors on the right from top to bottom, and vice versa. That methodology is shown on the bottom of Figure C.11. On the top of the same figure, SRII is also shown. A more complex pattern is defined for each three rows of the network. This scheme reduces the number of node coincidences by half in alternative routings compared to SRI. The third variant of SR, SRIII, introduces an initial random route selection at the beginning for only the first sampling time. Then, each node switches continuously and independently among the remaining three alternative next nodes. As previously considered for static routing, we use different dynamic attack scenarios depending on the routing algorithm used. They are: ARR (*Attack on RR*), ADRR (*Attack on DRR*) and ASR (*Atacck on SR*). The latter is in turn split for each one of the SR variants.

	MSE				
Attack scenarios	*d=0*	*d=1*	*d=2*	*d=3*	*d=4*
AMR	2500	–	–	–	–
ARR	3594.9	2135.7	1476.2	1127.9	883.2
ADRR	3708.2	2207.7	1514.3	1180.4	912.1
ASRI	2128.7	1727	1364.8	1191.7	983.2
ASRII	2116.3	1735	1330.8	1143.7	933.2
ASRIII	3576.8	2118	1472	1145.3	899.3

Table C.2: MSE numerical results for each attack dynamic scenario on sampling time $t = 10$.

Missing data recovery

To evaluate the performance of the new data recovery approach combining dynamic models and variable routing, the MSE value of tampered data recovery is computed for 10 consecutive observations (from 5th to 14th) in the FIR data set, where the evolution of a fire is measured. Worst-case tampering attacks, to those nodes in the rightmost column of the WSN, are considered. Table C.2 shows the numeric MSE at sampling time $t = 10$ for each of the routing approaches considered, and from $d = 0$ to $d = 4$ ($d = 0$ means that no time lag is used). Besides, the AMR attack static scenario (see Figure C.6) is also shown as a baseline for comparison.

After the analysis of the results in Table C.2, we can observe that in all the cases the recovery performance increases with the time lag, since more dynamic information is captured by the model. Therefore, the combination of DPCA with variable routing strategies is effective in terms of recovery performance. The routing strategies which present some degree of randomization (ARR, ADRR and ASRIII) have a similar behavior when the number of lags in DPCA is changed. This also happens for purely deterministic routing (ASRI and ASRII). Deterministic routing outperforms probabilistic routing for a low number of lags, but the opposite occurs as the number of lags grows. Since a better performance is obtained for a high number of lags, we can conclude that probabilistic routing is a better solution in terms of recovery performance. This can be explained by the fact that probabilistic methods perform a high variable distribution of the sensors in the routes. In consequence, the recovery data procedure gets more valid (non-tampered) temporal data. Furthermore, the dynamic alternative outperforms the recovery performance up to 60 %, in comparison with the static one. Additionally, Figure C.12 depicts the previous MSE results considering the number of lags leading the system to achieve the best performance, $d = 4$.

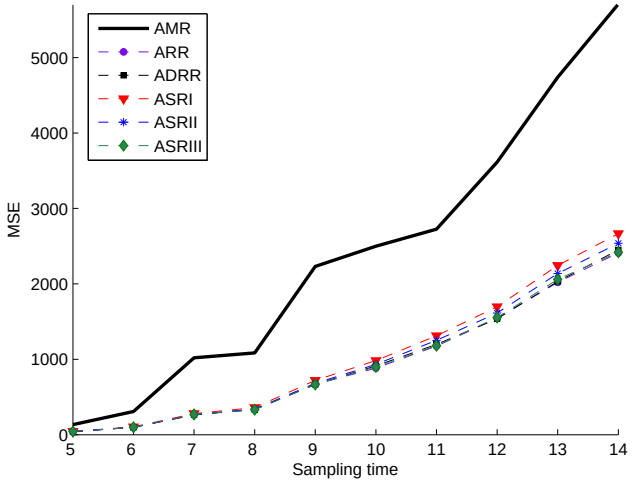


Figure C.12: MSE for each attack scenario considered and with $d = 4$ *lags*.

C.5.5. Local models to improve the missing data imputation

As previously stated, considering dynamic models together with dynamical routing strategies improves the data recovery efficiency. Despite of these achievements, in global models not all the data used for imputation are correlated. That fact is more relevant in wide and scarce WSNs. For this reason, a different data arrangement is proposed in the following.

Local modeling

We consider in this case only the sensors located in the vicinity area surrounding a compromised sensor. The PCA and PLS models calibrated using this data arrangement are referred as *local models*.

Figure C.13 depicts the arrangement process to build up a local model for PCA and PLS in the case of regular topologies (non-regular topologies are also addressed in contribution 5). The vicinity of a given sensor is defined by its closest neighbors: eight neighbors are considered. Each neighbor is represented by an arrow indicating the relative position to the affected sensor.

A measurement of each target sensor and its neighbors (*i.e.*, the locality is 9-dimensional) is acquired every sampling time, which constitutes an observation (row) in the local model. We only consider the $(K-2)\times(K-2)$ inner sensors to build the local model. Thus, the dimension of $\mathbf{X}$ is $I\cdot(K-2)^2\times 9$.

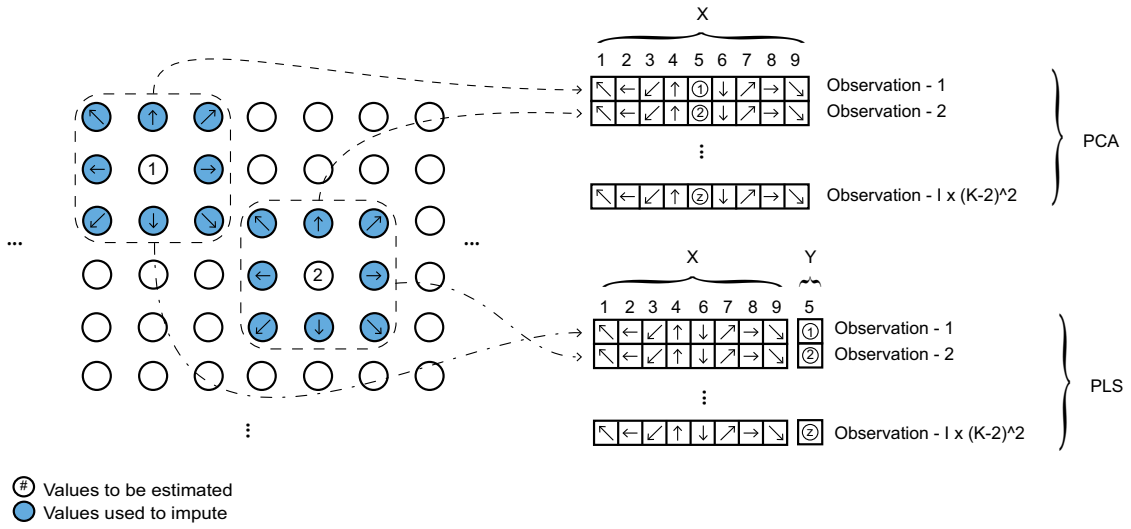


Figure C.13: Local model for a regular topology of the sensor field. The locality illustrated is established to 9 sensors, so that the total number of observations is $I \times (K-2)^2$, where I is the number of original observations and K the number of sensors per side. Thus, the number of sensors considered is $(K-2)^2$. In this structure z corresponds to the last sensor value at the I-th original observation.

In order to clarify the local model building, take the case in which we have $J = 81$ sensors, $I = 100$ observations of each of them, and $K = 9$ sensors per side corresponding to the regular sensor field. We have $(K-2)^2 = 49$ inner sensors while the dimension of the local model is 4900×9 for PCA, and 4900×8 in $\mathbf{X}$ and 4900×1 in $\mathbf{Y}$ for PLS.

Missing data recovery

To compare the recovery results obtained by using global and local models, the MSE and Q contribution plots are obtained. Figure C.14(b) shows the Q contribution after data recovery for the AMR scenario. It clearly outperforms the imputation provided by the global model in Figure C.9(b), and it resembles with high fidelity the case in which no attacks exist (Figure C.14(a)). A similar conclusion is obtained from the MSE values in Table C.3. Analogous results and conclusions can be extracted for ADR and ALR scenarios.

In summary, results in Tables C.1 and C.3 demonstrate the benefits of using local versus global modeling for data imputation. The new arrangement method can significantly reduce all MSE values. In the ADR case, a reduction of 99.86 % is

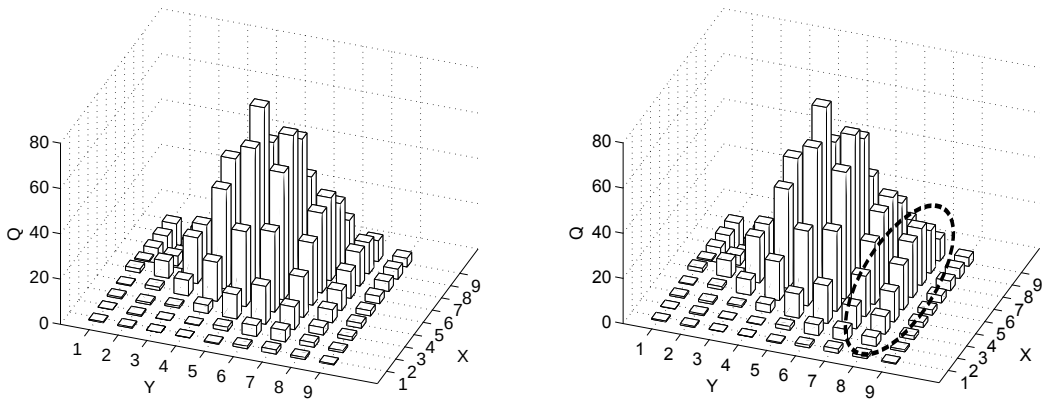


Figure C.14: TSR-PCA data tampering imputation through local modeling: (a) original profile for the fire, and (b) the recovery results for the AMR scenario. Those sensors affected by the associated attack are highlighted with a dashed line.

Attack scenarios	MSE (TSR-PCA)	MSE (TSR-PLS)
ADR	2.6506	3.4036
AMR	67.7999	71.085
ALR	149.5746	148.7155

Table C.3: MSE for local model-based TSR-PCA and TSR-PLS missing data imputation methods.

achieved, while the reduction is 97.25% in the AMR case and 96.61% in the ALR case.

Finally, we study the MSE evolution as a function of the number of tampered sensors. We randomly tamper from one to 10 sensors considering the ADR scenario. The results are shown in Figure C.15. Clearly, MSE grows with the number of tampered sensors.

From the previous discussion and results, we can conclude that our proposed missing data imputation method plus the local data arrangement lead to a high recovery performance even with adverse conditions: dynamic environmental changes (fire evolution) and a reasonably number of sensors tampered (around 12% of the total). Consequently, the proposal improves the robustness of WSNs against security threats, and so its survivability.

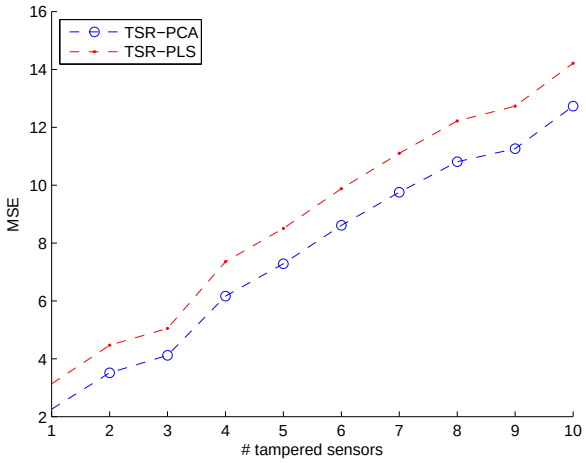


Figure C.15: MSE evolution with the number of tampered sensors considering local models, in the presence of fire and for the ADR attack scenario.

C.5.6. Real scenario: LUCE project

In this last part of the section, a real WSN scenario is used to corroborate the validity of the results previously obtained in simulation. For that, we select LUCE (*Laussanne Urban Canopy Experiment*)[2] as the real WSN deployment. LUCE is a WSN project driven at the EPFL (*École Polytechnique Fédérale de Lausanne*) campus since July 2006. The system is based on a wireless sensor network of 100 SensorScope weather stations that are deployed on the campus (about 500 m^2 area). These stations measure key environmental quantities at high spatial and temporal resolutions. Each SensorScope weather station has several sensors. Among others, there are ambient temperature, humidity and wind speed sensors. These measures are acquired and sent for analysis via GPRS (*General Packet Radio Service*) to a CU with a periodicity of 30 seconds.

Data collected from November 2006 to May 2007, available from the LUCE project web site, are used in this section. We have chosen data between January 1st and 31st, 2007, for our experiment because this corresponds to the most complete time interval, with 80,000 ambient temperature samples per sensor. The number of sensors used in our study is 61. Figure C.16 shows the location of these 61 sensors, specifying the latitude and longitude coordinates. Also we show the 8 closest sensors to a given one (sensor with ID=100) used for local modeling.

[2]LUCE deployment dataset at http://lcav.epfl.ch/page-86035-en.html

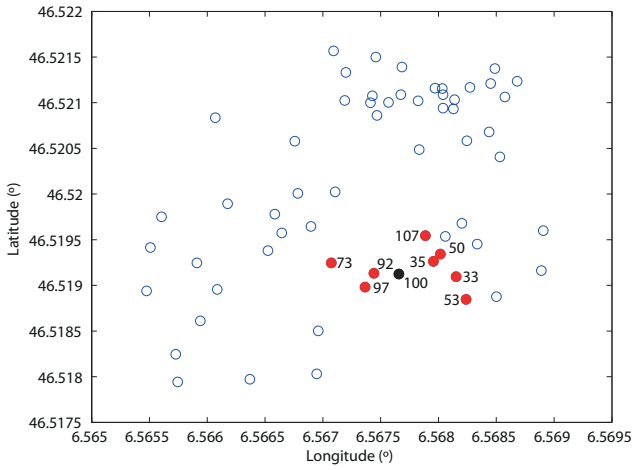


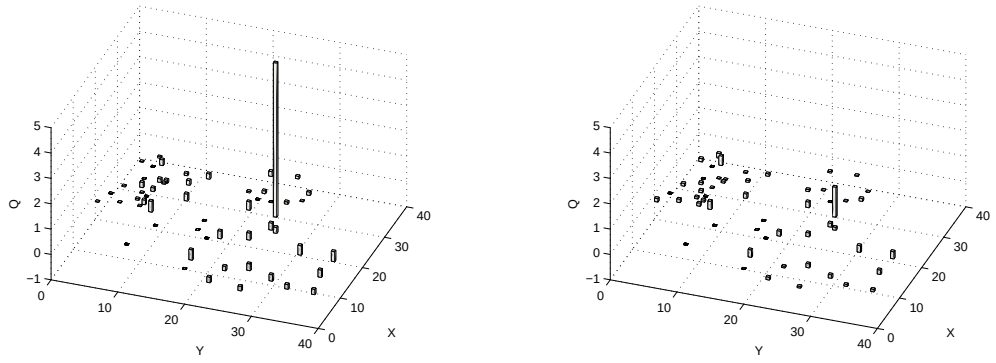
Figure C.16: Location of the 61 sensors in the LUCE deployment used in our experimentation. As an example, we show the 8 closest sensors to a given one (sensor with ID=100).

Global modeling

The same monitoring method developed for synthetic environments in simulation is deployed here for LUCE. The first twenty days of the previously mentioned data range are chosen as the calibration set to train the PCA model. The remaining days are used for testing purposes. The daily average value is subtracted from the data of the corresponding day to correct for temperature drifts along days. Note that there is no fire influence in this case. Therefore, an anomaly could be produced either by data loss or by a device malfunction. These anomalies, as in the WSN simulated case, can be deduced from the Q contribution graphics. Figure C.17(a) shows that the Q contribution presents a significant deviation for a specific sensor that was actually tampered for the experiments.

Missing data imputation

Figure C.17(b) shows the recovery results obtained when using global modeling. Numerical MSE results are also provided in the second column of Table C.4. No routing algorithm is used in the LUCE experiment, as data from sensors are directly sent via GPRS to the CU.



(a) (b)

Figure C.17: Q contribution graphics for ADR attack scenario in LUCE deployment: subfigure (a) shows the tampered sensor, while the recovery results are depicted in subfigure (b).

	Global model	Local model	
Attack scenarios	**MSE (TSR-PCA)**	**MSE (TSR-PCA)**	**MSE (TSR-PLS)**
ADR	0.1051	0.1081	0.1030

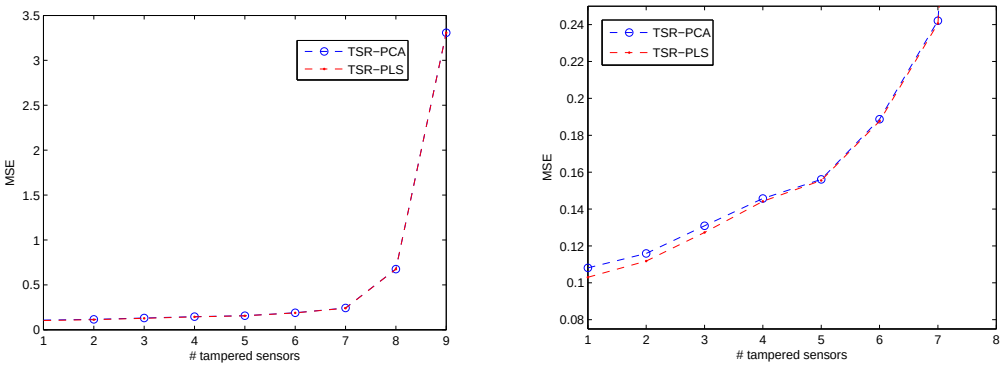
Table C.4: MSE comparison between global and local models.

Local models for missing data imputation and non-regular locations

Using local models for data imputation in regular and equally distributed sensors environments is straightforward, as the 8 closest sensors to a given one are those surrounding the latter. Defining the number of closest sensors in non-regular scenarios is not such an easy task. Therefore, a method to estimate this number is needed (see publication 5 for a detailed explanation). As in the regular case, the 8 closest sensors to a given one are chosen to conform the local model for this non-regular scenario.

The data imputation results obtained for LUCE when using local models are visually similar to those obtained for global models in Figure C.17(b). Table C.4 shows the associated numerical MSE values. It is notable the similarity in the results obtained to those with global models. This is mainly motivated because a high spatial correlation exists in the LUCE data set such that almost all sensors are highly correlated. To corroborate the existence of this high correlation, we calculated the correlation coefficients between variables, 0,89 being the minimum value found.

Another interesting experiment is useful to assess the robustness of the imputation approach when more than one sensor is compromised. For that, we sequentially increase the number of tampered sensors from a selected one to its closest 8 sensors. For example, 1 means that only a sensor is tampered, the selected sensor; 2 means



(a) (b)

Figure C.18: MSE evolution with the number of tampered sensors. Subfigure (a) shows that trend while the results for the first 7 sensor are detailed in (b).

that we tamper the specific one and its closest neighbor, and so on until 9 sensors in total. Figure C.18 illustrates the evolution of the MSE parameter with the number of tampered sensors. The MSE value does not vary significantly when the number of affected nodes is lower than or equal to 7, which in fact constitutes a remarkable hit for the recovery solution.

C.6. Relay node placement optimization

[Related publications: 1, 3, 7, 8, 9, 11, 12 and 15]

The distribution of the nodes in a network is usually driven by some performance objective. That is the case of WSNs, where the nodes are strategically located for high coverage of a monitored area. Instead, there exist situations like military or emergency rescue operations where network nodes do not follow a predefined distribution pattern. Usually, we are not able to control these nodes in any sense, so the contribution of additional nodes is needed. For this purpose, the so-called RNs (*Relay Node*s) are used to relay the information among network nodes. The placement/movement of these RNs is controlled to fulfill some specific requirements in terms of coverage, connectivity, communication delay and/or network lifetime as a function of the energy consumption [129].

Nowadays, the RN placement problem is still challenging. Thus, most of the solutions found in the specialized literature are based on heuristics able to provide suboptimal solutions [129]. Additionally, they mainly address this problem for static environments trying to provide the minimum number of RNs for achieving k-connected networks [140–142]. Nevertheless, the previous solutions are not longer

valid in dynamically changing environments. In that case, new problems and issues arise. For instance, the optimal RN locations vary over time as the topology does, hence these locations should be updated. Besides, the optimal number of the relays is also changing but this number is initially known and fixed in practical applications.

The topology of a MANET may vary due to the intrinsic node movements or because of the appearance of faulty or misbehaving nodes, like previously said (*e.g.*, dropping or sinkhole). This way, dynamically adaptable solutions are needed due to the continuous topology changes motivated by, on the one hand, the inherent node movements and, on the other hand, malfunctions or malicious behaviors.

To tackle these issues, we present a novel RN placement solution addressing these two principal aspects: which are their optimum location at a given time?, and how should they move towards those locations? We propose a modular optimization solution to solve both issues by means of the joined maximization of the network connectivity and throughput. For that, we introduce a innovative approach based on the one proposed by Dengiz *et al.* [131], yet solving some severe drawbacks in it. Like that approach, we use a PSO based optimization algorithms and MPC (*Model Predictive Control*) inspired methodologies to control the RN movements.

Since the proposal is feasible to solve failures in the network, it may be also used as a response/tolerance security solution to mitigate and recover from the effects of malign nodes. As a matter of fact, we will test the approach against *dropper* nodes operation.

RNs placement to improve connectivity & throughput

Maintaining or recovering the connectivity lost due to the disconnections caused by node movements, failures or misbehaviors may become a crucial task depending on the network application. For instance, in military or emergency rescue operations potential communication breakages could cause human losses. Additionally, preserving a lower bound of QoS (*Quality of Service*) in terms of the available throughput is mandatory to provide some kind of network services.

In the following, we briefly describe the fundamentals of the solution proposed by Dengiz *et al.* [131] (hereafter DKS, standing for the initials of the authors), which will be used as a guide for our solution. Such proposal presents serious limitations that will be discussed and fixed by our RN placement alternative.

DKS solution

Two types of nodes are involved in DKS: UNs (*User Node*s) and RNs. UNs are final nodes demanding some given network service, while RNs try to guarantee that UNs are receiving the best network service as possible. The authors assume that in a wireless environment, *e.g.* MANETs, any two nodes are accessible or connected (that is, there is a link between them) if the Euclidean distance between them is less or equal to c, where c is the coverage range of a radio node.

Basically, the authors suggest two objectives: to maximize (*i*) the overall network connectivity, and (*ii*) the throughput. This optimization process is achieved by using a PSO algorithm [135] and several cost functions. Two important and particular entries of the PSO algorithm are: (*i*) the future motion predictions of user nodes for a specific prediction time horizon ($t + H$), and (*ii*) the best solution obtained in the previous time step. Afterwards, a comparison among several possible solutions (*particles*) is made. The different particles in the same PSO execution are specific network distributions where the UN locations at $t + H$ remain equal, and the RN positions are modified by increasing or decreasing the velocity and direction values. When the iteration of the PSO optimization process is finished, the algorithm returns the best locations for each RN that maximize the overall connectivity and throughput of the network at a given instant.

DKS involves three objective functions, jointly used during the optimization procedure. Through the first one, O_1, the global network connectivity is evaluated. That function gets a connectivity value depending on the number of interconnected pairs of nodes [131]. The limit values for O_1 are 0 and 1, for completely disconnected and connected networks, respectively. A second function, O_2, computes the minimum network throughput, which corresponds to the weakest network link. The authors approximate the *throughput* of a determined link (w_{ij}) between two nodes i and j as a function of their distance. O_2 is only evaluated when several candidate solutions in the optimization procedure (several particles under evaluation) represents completely connected networks (*i.e.*, for $O_1 = 1$). Instead, for disconnected networks (*i.e.*, $O_1 < 1$), an alternative function O_3 is considered. O_3 measures the distance from each RN to the imaginary middle points (AP, *Attraction Point*s) among network partitions, so that such distances are minimized by locating the RNs as close as possible to the APs.

In summary, the PSO algorithm follows a selective procedure to get the best location for each RN. First, the solution with the highest O_1 value ($\max\{O_1\}$) will be the best solution. If there are several solutions corresponding to completely connected situations (alternative particles with $O_1 = 1$), the one with highest O_2 value ($\max\{O_2\}$) is selected. In case of disconnected networks with equal values for O_1, the one with lowest O_3 value ($\min\{O_3\}$) is chosen.

The optimization algorithm is iteratively repeated over time, the RNs being dynamically positioned at their best locations step by step. More details about the entire process can be found in reference [131]. Publications 11 and 12 demonstrate that DKS has a direct application to strengthen the security in MANETs in the presence of attacks like *dropping*.

DKS limitations

As mentioned, most of the RN placement solutions try to solve two main aspects: (*i*) which is the best location of each available RN, and (*ii*) how they should move to those locations. These two important concerns have not been correctly addressed in DKS. This is reasoned in the following:

- The mathematical formulation and problem statement in DKS is not adequate, leading to an overall optimization that switches among three different optimization functions. This in turn leads to fairly suboptimal solutions and problems in the boundaries amongst the functions. Furthermore, the network connectivity is addressed through the use of a discrete cost function (O_1). The use of a discrete function should be, if possible, avoided in an optimization formulation, since discrete functions divide the search space into flat regions without differential information to drive the optimization.
- DKS makes use of imaginary APs towards which the RNs are moved to improve performance goals. An AP is roughly defined as the middle point between two network partitions, a naive and suboptimal definition for most challenging scenarios. Take for instance the case where two partitions in the network are far enough (just more than $2 \cdot c$, where c stands for the coverage radius) so that placing an RN in the middle point does not lead to any improvement in terms of network performance. It is also worth noting that the number of APs grows with the number of partitions, thus making necessary a previous selection procedure in a coherent way (see publications 9 and 15).
- The RN positioning in DKS for disconnected situations is based on minimizing the minimum distance among the RNs and the APs in the network. This double minimization leads to the actual movement of a single RN, the nearest one to a given AP. The rest of RNs will remain uncontrolled, which is in fact an undesired behavior (see publications 9 and 15).

Although the previous limitations are equally relevant, it is important to note at this point the special importance of the two latest points. They are further developed in what follows.

C.6.1. Movement control and positioning improvements

Preliminary aspects & concepts

As in the DKS proposal, we consider two kinds of network nodes: UNs and RNs. Let us pose some preliminary assumptions that are found elsewhere although not directly specified [131]:

- *The optimization procedure is centralized.*

  This implies that there exists a node capable, through the adequate communication and processing means, to: (*i*) retrieve the necessary network information, (*ii*) run the optimization algorithm from this information, and (*iii*) send control data (*i.e.*, the target locations) in accordance with the result of the optimization.
- *The position of the UNs in the network is not controlled.*

  Only RNs can be optimally positioned. This is a reasonable assumption that complicates the problem. Moreover, this problem definition generalizes the simpler problem in which the location of the UNs can also be optimized.
- *The number of RNs is limited.*

  This is a realistic assumption. In most situations only a limited number of RNs is available.

Additional secondary assumptions adopted throughout the work are:

- *The optimization is limited to a 2D space.*
- *UNs are mobile devices around the environment.*
- *The network is single-tiered [147], so that both UNs and RNs relay information from other UNs.*
- *The communication range for both UNs and RNs is the same: c meters.*
- *Both types of nodes have a limited speed.*

In this general context, a MANET consisting of several UNs and RNs can be specified as follows:

$$G = (N, E) \tag{C.9}$$

with

$$N = U \cup R \tag{C.10}$$

where U, R, N and E stand for the set of UNs, the set of RNs, the complete set of wireless nodes, and the wireless links (edges), respectively.

In a wireless network, the edges satisfy:

$$E := \left\{ \mathbf{e}_{ij} \mid \|\mathbf{e}_{ij}\| \leq c, \forall n_i, n_j \in N \right\} \tag{C.11}$$

with c the communication range of a single link, and n_i and n_j the i-th and j-th network nodes.

The optimal placement of the RNs can be represented by the graph composed of the complete set of wireless nodes in the network:

$$\mathbf{G}^* := \arg\max_{\mathbf{G}} \{ f(G) \mid G = (N, E) \; and \; U = U_0 \} \tag{C.12}$$

where $f(G)$ is the function to be maximized and U_0 the actual location of the UNs. Function $f(G)$ represents a procedure where both connectivity and throughput should be directly or indirectly considered.

Since the current proposal is mainly devised to solve the DKS limitations, we firstly present a formal mathematical formulation for DKS, which is indeed missing in the corresponding original work [131]. Following that explanation as a guide, we will discuss and introduce our modifications in accordance with the problematic observed for DKS.

Optimized RNs location & movement

The RNs have to be moved in accordance with the changes produced in the environment. In some sense, they should have certain awareness of the future changes. That acquired knowledge will lead them to build better and more efficient trajectories. Inspired in MPC techniques, the behavior of the MANET is predicted in DKS in a receding horizon with H steps. This way, changes in the MANET can be anticipated.

DKS considers several cost functions. Let us define R_{O1} as the set of optimal solutions, in a given PSO iteration, for the RNs placement in terms of function $O_1(G)$:

$$R_{O1} := \arg\max_{R^{(t+H)}} \left\{ O_1(G(U^{(t+H)} \cup R^t, E)) \right\} \tag{C.13}$$

where U^{t+H} are the predicted locations for the UNs in the receding horizon H, and R^t the computed locations for the RNs in the current time, $O_1(G)$ being as follows:

$$O_1(G) = \frac{2}{|U| \times (|U|-1)} \times \sum_{\forall u_i, u_j \in U, j>i} z(G,i,j) \tag{C.14}$$

where $|U|$ is the number of UNs considered and u_i and u_j the i-th and j-th UNs, respectively. $z(G,i,j) = 1$ if there is a path connecting u_i and u_j in G (that path could be direct of through a *multi-hop* based routing), while $z(G,i,j) = 0$ otherwise.

Due to the discrete nature of $O_1(G)$, it is likely that the cardinality of R_{O1} will be higher than 1, so that there is no unique optimal solution with respect to $O_1(G)$. In such a case, DKS performs the following procedure to select the optimum placement R^*_{DKS} amongst the solutions obtained in R_{O1}, where $O_2(G)$ and $O_3(R,A)$ are used:

$$R^*_{DKS} := \begin{cases} \arg\max\limits_{R^{t+H}} \left\{ O_2(G(U^{t+H} \cup R^{t+H}, E)) \mid R^{t+H} \in R_{O1} \right\}, & \text{if } O_1(R_{O1}) = 1 \\ \arg\min\limits_{R^{t+H}} \left\{ O_3(R^{t+H}, A^{t+H}) \mid R^{t+H} \in R_{O1} \right\}, & \text{if } O_1(R_{O1}) < 1 \end{cases} \tag{C.15}$$

with A^{t+H} the estimated APs in the horizon H. $O_2(G)$ follows the expression:

$$O_2(G) = \min_{u_i, u_j \in U: j>i} \{ T(G,i,j) : T(G,i,j) > 0 \} \tag{C.16}$$

where $T(G,i,j)$ represents the throughput between the i-th and j-th UNs in the network G, it being limited by the weakest link in that terms.

$O_3(R,A)$ is defined as:

$$O_3(R,A) = \min_{r_i \in R, a_j \in A} \left\{ \sqrt{(r_i^x - a_j^x)^2 + (r_i^y - a_j^y)^2} \right\} \tag{C.17}$$

where r_i is the i-th RN and r_i^x and r_i^y its 2D coordinates, and a_j is the j-th AP and a_i^x and a_i^y its 2D coordinates.

Aimed at avoiding problems caused by the simultaneous use of several cost functions and their discrete nature in the optimization procedure, we propose two different and separated optimization procedures depending on the network connectivity status: one for connected networks, and another for disconnected ones.

RN placement optimization for connected networks

A connected network means that $O_1(G^U) = 1$, where G^U denotes the network graph just considering the UNs. In such cases, the network connectivity is maximum, but the network throughput can still be maximized. For that, DKS involves $O_1(G)$ and $O_2(G)$ functions, which is inefficient. Alternatively, we devise a unique cost function to avoid the combined use of several objective functions and the discrete nature of $O_1(G)$. This function is expressed by:

$$g(G') = \sum_{\forall u_i, u_j \in U : j > i} id(G'_i, i, j) \tag{C.18}$$

where G'_i is the spanning tree starting at the i-th UN and minimizing the distance of the largest edge of each path in the network. The function $id(G'_i, i, j)$ is the inverse of the length of the longest edge in the path from i to j in G'_i network. $g(G'_i)$ can be interpreted as an estimation of the overall network throughput, where the corresponding throughput between two adjacent nodes is approximated by the inverse of the distance, *i.e.*$w_{ij} = \frac{1}{d_{ij}}$. Additionally, $g(G')$ can be seen as a smoother and continuous version of $O_1(G)$, with $G' = (N, E')$ and $E' := \left\{ \mathbf{e}'_{ij} \mid \forall n_i, n_j \in N \right\}$.

RN placement optimization for disconnected networks

Those cases where $O_1(G^U) < 1$ correspond to disconnected networks. This situation is expected to be frequent in MANET environments due to the inherent nodes' mobility. Consequently, the system will try to move the RNs to those previously computed APs for recovering (or even improving) the connectivity lost. Recall that DKS uses a function $O_3(R, A)$ that is ill-defined since just one RN is moved towards its closest AP. To solve this problem we propose the function $p(R, A^*)$ instead of the combined use and evaluation of $O_1(G)$ and $O_3(R, A)$. It follows the expression:

$$p(R, A^*) = \sum_{i=1}^{R} \sum_{j=1}^{A^*} d(r_i, a^*_j) - \sum_{i,j \in R : j > i} d(r_i, r_j) \tag{C.19}$$

where r_i is the i-th RN and a^*_j is the j-th optimized AP. The function $p(R, A^*)$ tries to move the RNs to the APs while maintaining the former ones separated. See publication 9 for a more detailed justification of the definition of $p(R, A^*)$. Unlike $O_3(R, A)$, this redefinition takes into account all RNs and APs, as well as it uses an optimized set of APs, A^*, instead of the naive original definition in DKS. As we will see later, these AP (*Attraction Point*) locations together with the alternative $p(R, A^*)$ function will lead the system to achieve a better performance.

RN placement complete view

For the sake of clarity, the complete alternative optimization procedure could be summarized as follow:

$$R^*_{new} := \begin{cases} \arg\max_{R^t} \left\{ g(G'(U^{(t+H)} \cup R^t, E)) \right\}, & \text{if } O_1(G^U) = 1 \\ \arg\min_{R^t} \left\{ p(R^t, A^{*(t+H)}) \right\}, & \text{if } O_1(G^U) < 1 \end{cases} \tag{C.20}$$

Now, the system deals with smooth, continuous and different cost functions depending on the specific casuistry observed in the network. Note that function $O_1(G^U)$ is just a selector used outside the PSO unlike in DKS where it is used within the algorithm.

C.6.2. DRNS (*Dynamical Relay Node placement Solution*)

According to the previous problem formulation, we devise a novel RN placement solution, hereafter called DRNS (*Dynamical Relay Node placement Solution*). Its functional and modular architecture in shown in Figure C.19. There we can observe several and different modules. Leftmost, the ***DKS motion prediction*** is in charge of inferring the UN positions in a receding horizon ($t + H$). The UN location prediction U^{t+H} is afterwards used to select the corresponding optimization way depending on function $O_1(G^U)$. Only for disconnected networks we first need to compute an optimized set of APs ($A^{*(t+H)}$) by means of the execution of the ***APs optimization*** module. Afterwards, the ***RN motion control*** module comes into play by running a PSO algorithm considering the corresponding unique cost function in accordance with the network status.

A more detailed explanation of the previous modules is introduced in the following.

APs optimization module

As mentioned before, the DKS solution does not locate the APs correctly. The present module solves this limitation by firstly, considering the available number of RNs, and secondly, optimizing their positions in accordance with the system performance goals. Following the previous premises, the module is in turn divided into three, each one devised to perform a specific task to get efficient AP locations. The associated modules and their relationships are shown in Figure C.20. Firstly, a homogeneous **distribution** of the APs is done. They are distributed along the edges connecting partitions in the network according to a distance-based spanning tree.

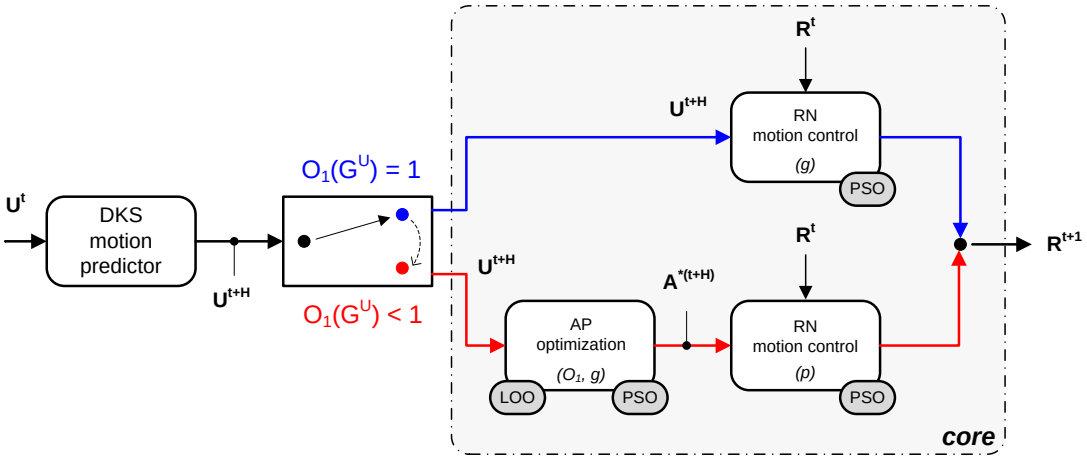


Figure C.19: Functional blocks of DRNS.

The number of APs is customized by a user-defined parameter $\lambda \in (0,1]$. Secondly, a LOO (*Leave-One-Out*) based procedure is executed to make an AP **selection** from the previous set. This module selects as many APs as RNs are available in the system. Finally, from the previous valid solution, a PSO **optimization** procedure re-locates the selected RNs such that their new positions improve the system performance in terms of connectivity and throughput.

Figure C.21 shows an example of the results of each module in the entire optimization. A remarkable fact is the relocation of the APs from their positions obtained in the selection stage (Figure C.21(c)) to those reached in the optimization one (Figure C.21(d)): the PSO extends the search space leading the system to find a better solution not contemplated in the previous stage.

It is worth noting that this module can be seen as an RN placement solution by itself, just replacing the APs by RNs and considering static scenarios, *e.g.*, infrastructure-based WiFi networks where the RNs are located to extend the network coverage.

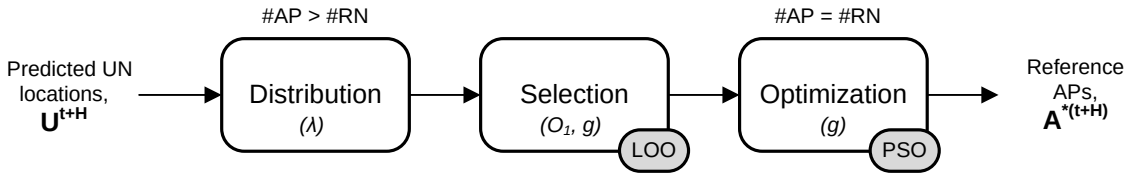


Figure C.20: Stages of the APs optimization module.

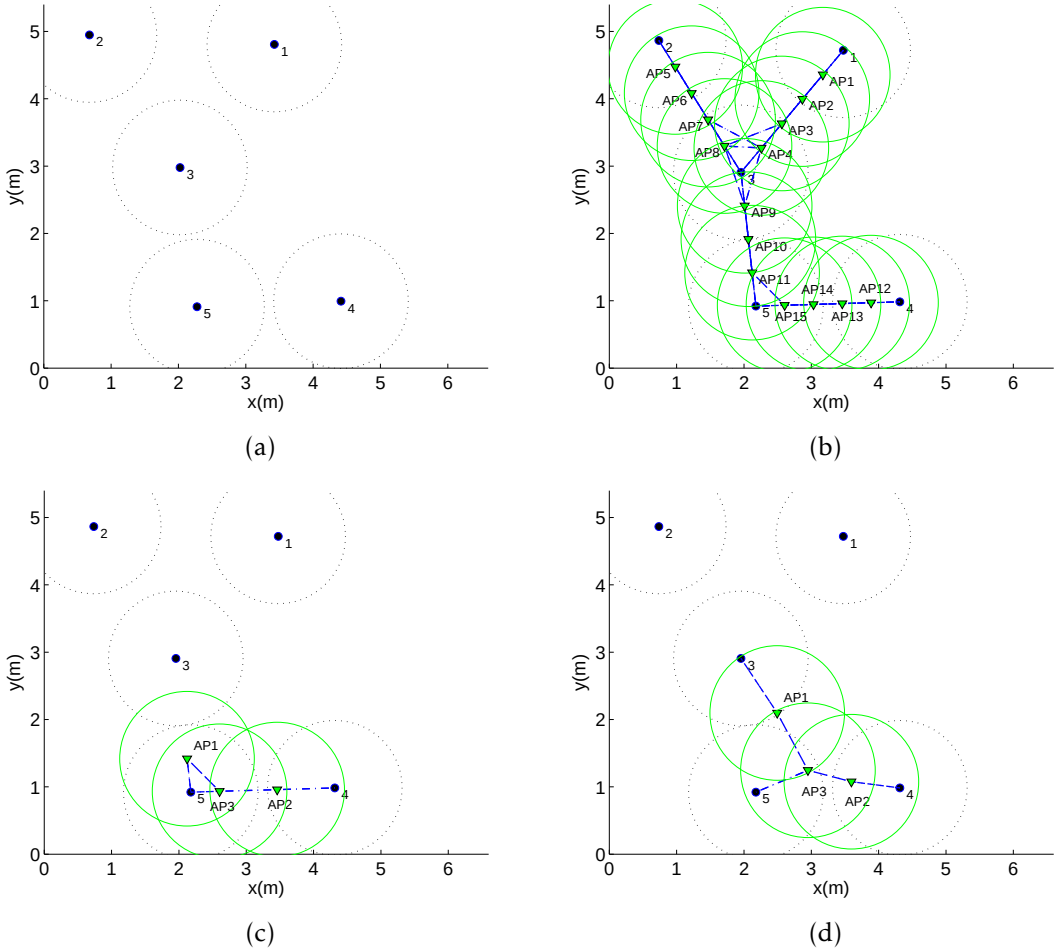


Figure C.21: APs location after each stage considering 5 UNs (filled circles) and 3 available APs (filled inverted triangles): (a) initial UNs distribution, (b) initial candidate APs location along the spanning tree minimizing the inter-partitions distance, (c) APs selection, and (d) APs optimal location after the PSO-based optimization.

RNs motion optimization module

Leading the RNs to certain target points is especially challenging in highly dynamic changing environments like MANETs. Firstly, because of the adaptation itself to the changes in the topology. But also because the RN movements are physically limited by their own maximum velocity.

To address this challenge, we first optimize the RN movements in a receding horizon through the evaluation of different cost functions according to the network status. However, the optimization candidate solutions are limited to $H \cdot max\_velocity$, where $max\_velocity$ corresponds to the maximum velocity of the RNs. Although the optimization is solved H steps ahead, only one step is implemented and the iteration

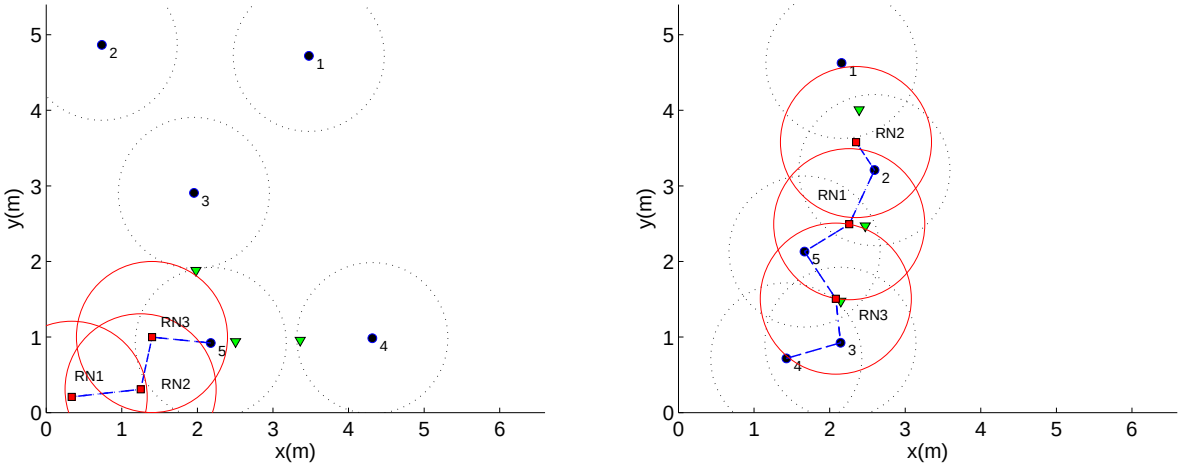


Figure C.22: Illustration of the RN movements (red squares) during the simulation time towards the APs (inverted green triangles). Subfigure (a) shows the initial status while subfigure (b) depicts the adapted locations of the RNs to the current network status some time steps after.

is repeated all over again. This working methodology is inspired in the widely used MPC theory. Figure C.22 depicts an example of the controlled RN movements for disconnected networks at the beginning of the experiment (Figure C.22(a)) and some time steps after (Figure C.22(b)). Such figure shows how the RN locations are adapted to the changes.

C.6.3. Evaluation and simulation results

In order to corroborate the viability of the placement proposal in MANET environments, we devise the simulation environment and experimentation described in the following.

Simulation scenario

The chosen experimental simulation environment is based on Matlab according to the DKS solution [131]. Network connectivity is measured through Eq. (C.14), and throughput by means of:

$$th(G') = \sum_{\forall u_i, u_j \in U : j > i} ie(G'_i, i, j) \tag{C.21}$$

where G'_i corresponds to the MST (*Minimum Spanning Tree*) starting at the i-th UN. The function $ie(G'_i, i, j)$ measures the available throughput from i to j nodes in G'_i. Moreover, as in DKS, the throughput between two adjacent nodes (w_{ij}) is approximated as a function of their distance.

Aimed at analyzing realistic MANET environments, we chose a deployment rectangular area of $6{,}6m \times 5{,}4m$ while set the node coverage range to $1m$ assuring network disconnections. Additionally, the RNs velocity should be equal or higher than the UNs velocity. Otherwise, the system could not be able to adapt to the changes in the network. That way, we set the RNs velocity to $0{,}15m/ts$ and $0{,}1m/ts$ for the UNs. Note that the velocities of the nodes are referred to one *simulation time step (ts)* instead of seconds. The effects of lower RN velocities can be seen in publication 1.

Another important aspect is the UNs mobility pattern, since their movements are not controlled. The RWP (*Random Way Point*) [183] movement pattern has been chosen to test the system behavior for sparse networks. Publication 1 shows the system evaluation when the RPGM (*Reference Point Group Mobility*) [184] mobility pattern is used for dense networks.

Results

As previously stated, the APs optimization module can be seen in fact as the RN placement solution for static scenarios. Though it is not the main purpose of DRNS, the corresponding evaluation and results can be found in publication 3.

Considering the aforementioned MANET scenario, Figure C.23 shows the cumulated mean achieved by DRNS and DKS (leftmost and rightmost graphics in the figure, respectively) in terms of connectivity and throughput (at the top and bottom of that figure, respectively). The previous results are obtained by considering 25 repetitions using RWP mobility patterns to set the UN movements around the predefined area during the simulation. The network is composed by 3 UNs varying the RNs from 0 to 3. As observed, in all the cases DRNS outperforms DKS in both connectivity and throughput for 2 RNs or more, they providing a similar performance for 1 RN. This result evidences the limitation of DKS to take advantage of using more than one RN. On the contrary, DRNS always makes use of all the RNs to get improving performance in terms of connectivity and throughput as the number of RNs is increased. Instead of DKS, DRNS is able to adapt to the continuously changing environment quicker. This is mainly motivated by the new and smart locations of the APs and how the RNs are moved to these locations over time. The network is disconnected most of the time for the selected scenario, so $p(R, A^*)$ function moves the nodes in a better way than the original DKS function, $O_3(R, A)$, does.

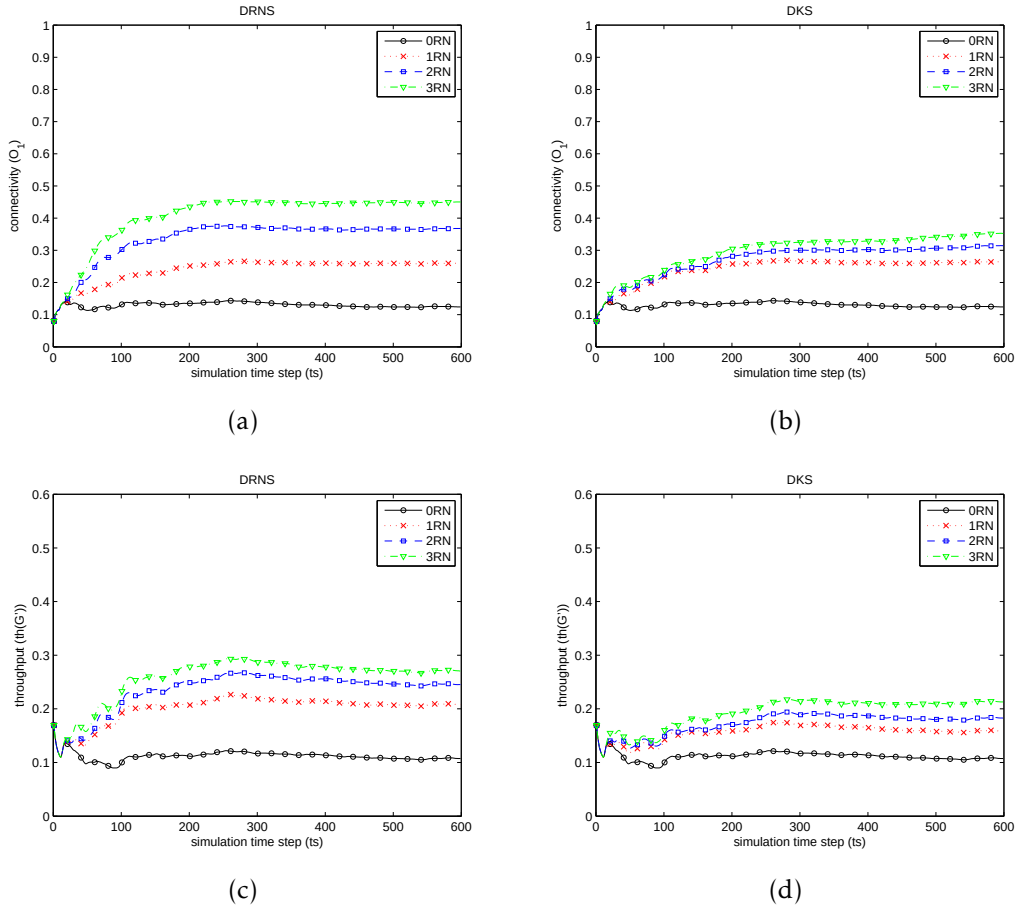
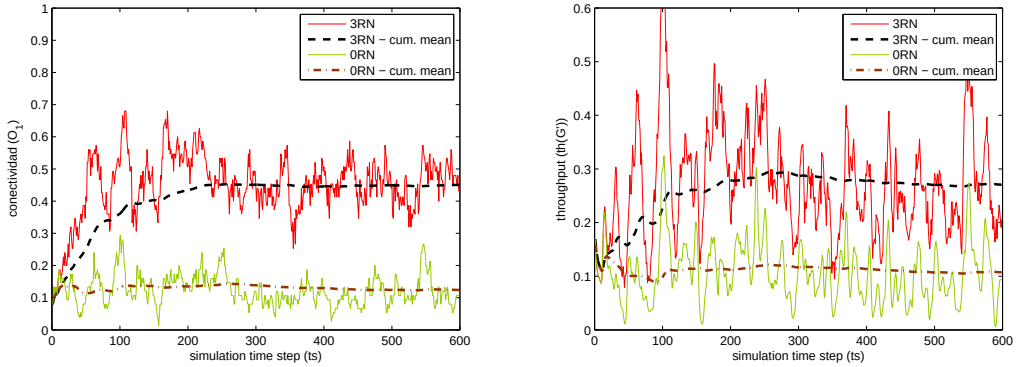


Figure C.23: DRNS system performance in comparison with the DKS solution. Connectivity and throughput cumulated mean values are represented. Three UNs are considered, they moving following a RWP pattern, while the number of RNs varies from 0 to 3. Subfigures (a) and (b) show the connectivity results for both solutions, DRNS and DKS, while sufigures (c) and (d) show the correspoding throughput values.

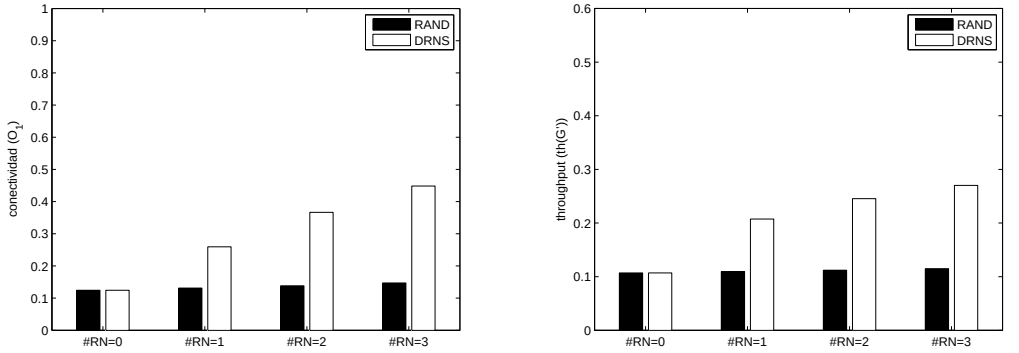
In order to show how changing the environment is, Figure C.24 depicts the connectivity and throughput evolution through their instant values as well as the cumulated mean for 3 UNs, and 0 and 3 RNs. That figure clearly shows the high dynamism of the environment addressed.

Additionally, we compare our solution with a simplistic one that moves the RNs in a random way throughout the area. This solution is called RAND from here onwards. Figure C.25 illustrates DRNS and RAND performance comparison, showing the smart localization carried out by the former in comparison with the latter. An extended work evaluating the DRNS scalability with the number of UNs and its computational complexity is successfully addressed in publication 1.



(a) (b)

Figure C.24: DRNS performance ((a) network conectivity and (b) *throughput*) considering 0 and 3 RNs, and 3 UNs, the last ones moving following a RWP mobility pattern. Cumulated mean (dashed lines) and instantaneous (continuos lines) values are shown.

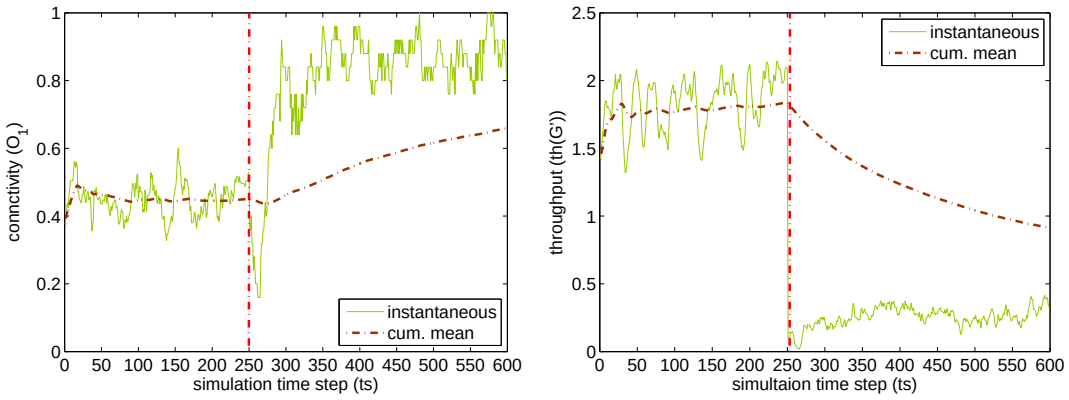


(a) (b)

Figure C.25: DRNS and RAND performance comparison when 3 UNs are used while the number of RNs varies from 0 to 3, considering RWP for the UN movements. Subfigures (a) and (b) show the results with the number of RNs for the network connectivity and throughput, respectively.

DRNS as a response/tolerance solution

As commented through the previous sections, the continuous changes in the network environment are mainly motivated by the inherent dynamism of MANETs. However, these topology modifications could have another origin. That way, faulty or malicious nodes can modify the network topology too. That way, DRNS may be also used as a response related security solution. From the point of view of network security, one of the most harmful attacks in MANETs are the so-called *dropping* attacks [5].



(a) (b)

Figure C.26: DRNS performance recovery when considering 3 malicious nodes, 5 UNs and 3 RNs. Figure (a) shows the connectivity evolution while figure (b) shows the network throughput. Vertical red dashed line denotes when the 3 malicious nodes start the attack at the same time.

For the purpose of validating DRNS as a reaction solution against dropping attacks (which can also be applied to others like *route poisoning attacks*, like *sinkhole* or *wormhole attacks*), we perform several experiments which show how our proposal is able to react to mitigate the undesirable effects provoked by such kind of malign actuations. For that, we consider 3 malicious nodes, 3 RNs and 5 UNs, the last ones moving following a RPGM mobility pattern. The malicious nodes act as normal nodes until a certain simulation time step.

We suppose that a detection mechanism is deployed as a complement of our proposal (*e.g.*, [185]). Consequently, once the attack is detected, the DRNS response mechanism is initiated to recover the network performance. Figure C.26 shows the network connectivity and throughput evolution, where we can observe the effect of the attack just before the time step 250. As expected, the performance values decrease but they are afterwards recovered and even improved in the case of the network connectivity. It is important to mention that the maximum value of the throughput is reduced according to Eq. (C.21) by the fact that we remove from the network every malicious node detected.

C.6.4. Real scenario: IDSIA robotic laboratory

Most of existing RNs location approaches are tested by means of simulation, as this saves deployment efforts, time and costs in comparison with real environments. Nevertheless, not all inherent aspects in real environments can be accurately simulated. Although simulation results can determine the efficacy or efficiency of a

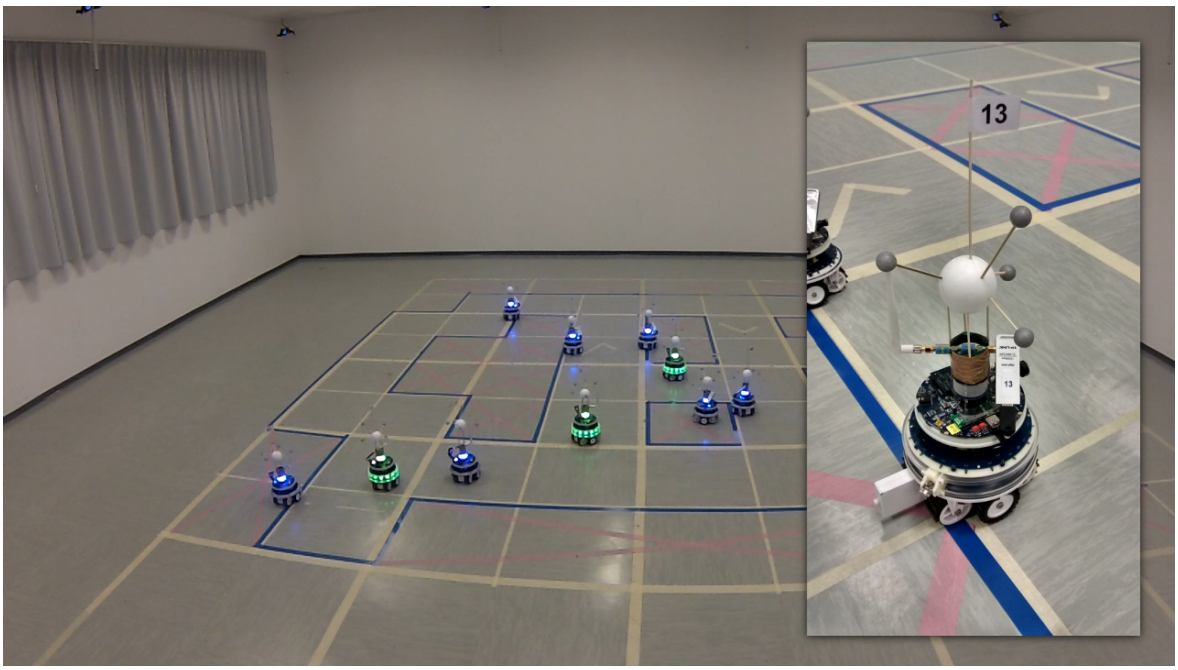


Figure C.27: Snapshot of the real robot environtment used in the experimentation. We can see 7 UNs (blue light) and 3 RNs (green light) which locations are optimized.

proposed solution at a first instance, its deployment in real scenarios can lead the system to very different results. Many aspects need to be considered before deploying the system in a real environment, specially when considering indoor MANET scenarios. For example, the coverage radii, the node movements, node control and communications, physical aspects like node energy consumptions and obstacles avoidance, among others.

In this work, we have used a robotic scenario to validate our proposal, so that the results obtained can be compared with those in simulation. Such a robotic scenario has been deployed at IDSIA (*Institute Dalle Molle for Artificial Intelligence*) [186], and corresponds to an indoor laboratory with an available area of $6{,}6m \times 5{,}4m$ that maintains a safety zone from the surrounding walls. Figure C.27 shows a picture of the environment where the mobile robots can be seen on the arena with different color lights on board, while one of them is shown in detail on the right (those robots are colloquially called *foot-bots* [188]).

A number of concerns arise for this real framework. For instance, the robots movements throughput the area, avoiding potential obstacles and others robots. Moreover, each one must know the own 2D location and has to be able to receive the new target coordinates. For that, we devise and implement and architecture based on the use of an existing tracking system including several software modules allowing

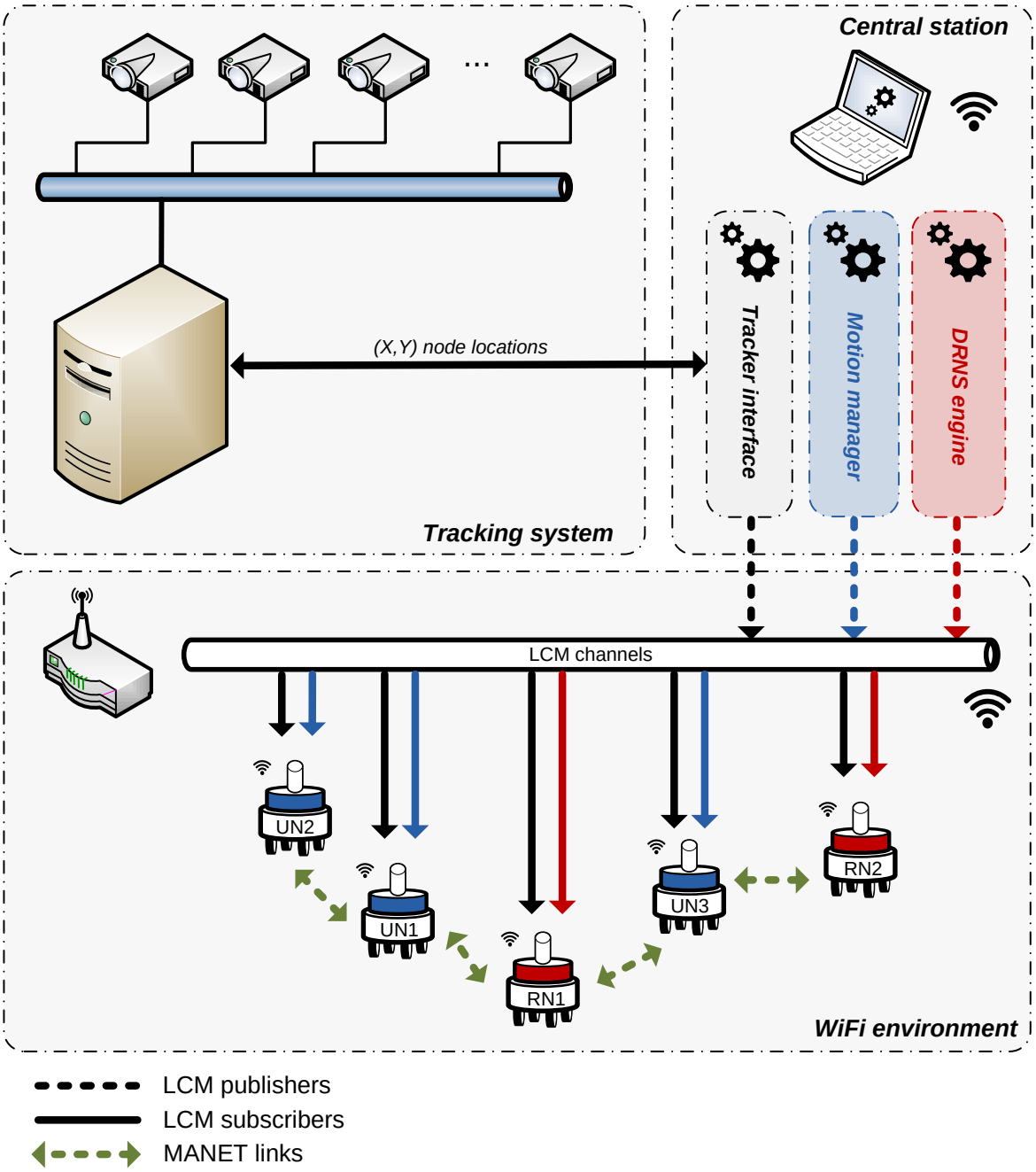


Figure C.28: Overall functional architecture of the real scenario.

to control the RN and UN movements. The developed architecture is shown in Figure C.28, for which more details can be found in publication 1.

In the following we show the results obtained for that environment. With the aim of comparing the results obtained in simulation we replicate the same experiments in the real environment. Unlike the simulation case, here we obtain the system

performance values through just 5 repetitions, the rest of the configuration remaining equal except for the RNs velocity. This is set to $0{,}2m/s$ to soften the effects of the acceleration and direction changes not considered in simulation.

Results

Figure C.29 shows the connectivity and throughput evolution provided by DRNS in the real robot-based scenario described in the previous section. We can observe a similar behavior than that in simulation: the system performance increases with the number of RNs. However, two main differences may be noticed in regards to simulation results (Figure C.23): (*i*) the system provides a slower adaptation to changes in network topology, and (*ii*) the general performance values are lower than those obtained in simulation.

Clearly, the real scenario includes some unexpected elements not considered in the simulation case. In particular, these differences are in part motivated by the fact that robots, including the RNs, have to avoid obstacles such as other nodes and walls. Another aspect that affects the results obtained is the velocity of the robots. In simulation, this parameter is roughly approximated by fixed displacements in each time interval, while in a real environment robots need to turn and accelerate while changing their direction of movement. For this reason, we had to increment the RNs velocity mostly softening these effects not considered in simulation. In summary, the inherent characteristics in real scenarios give way to slightly lower performance values than those in simulation. Such restrictions are specially relevant in dense scenarios. This behavior is depicted in Figure C.30, which shows the system performance obtained by DRNS and RAND with a RPGM mobility patterns. In that case, the robots will intercept others more likely. This causes that the improvement achieved with 3 RNs is almost the same than with 2 RNs. The higher the number of nodes the bigger the effect.

Beyond particular effects as those previously discussed associated to physical environments, we must remark the validity of DRNS approach also to be deployed in real networks. As a proof not only of the performance improvement achieved but also of its actual usability, a video with some of the real experiments carried out can be seen in publication 8 as supplementary material.

C.7. Integration of security solutions

[Related publications: 4, 10, 14 and 16]

As stated along the document, the main objective of the present work is to develop novel response/tolerant solutions for ad hoc networks as part of a more ambitious

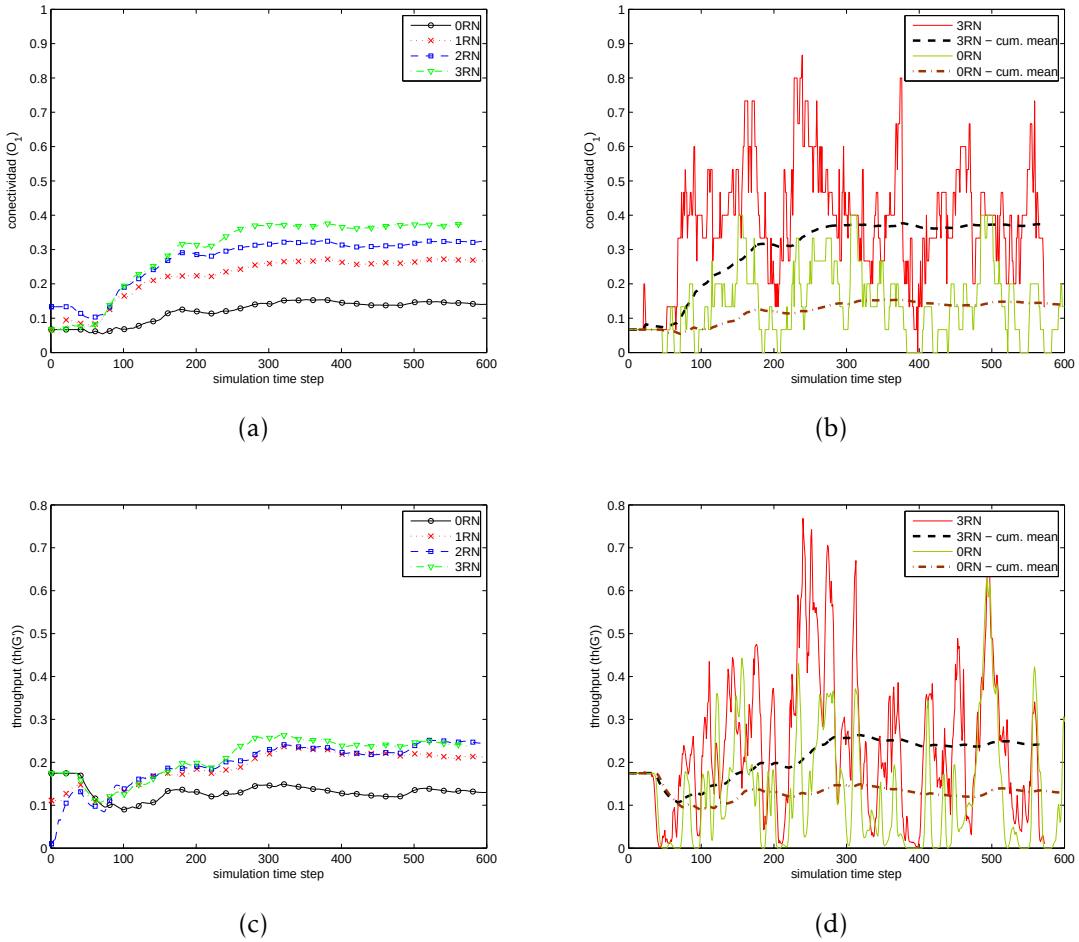
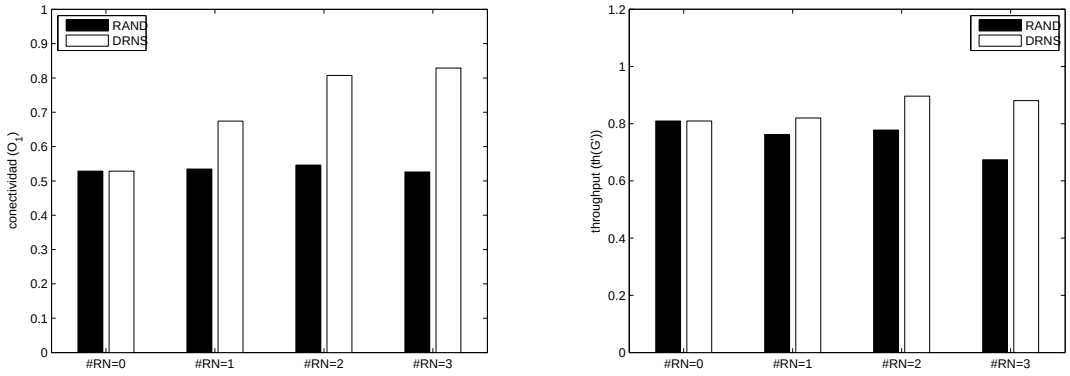


Figure C.29: DRNS results obtained for the real environment. The number of RNs varies from 0 to 3, while that of UNs is fixed to 3. Subfigures (a) and (c) depict the cumulative mean evolution for connectivity and throughput as a function of the number of RNs. Subfigures (b) and (d) show the instantaneous (continuous lines) and the cumulative mean (dashed lines) values by using 0 and 3 RNs.

purpose: the *network survivability*. That way, achieving survivable systems imply to consider several properties and requirements apart from the recovery related one addressed until the moment in this work. For this reason, we present here a modular framework mainly devised to integrate several defense lines (in particular the response schemes developed) to provide more overall robust and resilient systems from the point of view of security.



(a) (b)

Figure C.30: Performance comparison between DRNS and RAND with RPGM mobility pattern. Subfigures (a) and (b) show the connectivity and throughput values, respectively, versus the number of RNs, the number of UNs being fixed to 3.

Our integration solution is supported on the NETA (*NETwork Attacks*)[3] framework (see publication 10), which is in turn developed in the OMNeT++ (*Objective Modular Network Test-bed in C++*) simulator [206]. Devised for testing and developing attacks in an easy way, NETA is expected to be a useful tool for the research community.

In particular, until our very developments, NETA provided tools just to deploy attacks and detection mechanisms against them. We have extended such capabilities in two directions. On the one hand, to be able to put into action response techniques too. On the other hand, to integrate in a modular and robust way each of the different modules in order to provide and test global security solutions. This way, we are now capable in NETA to define and run simulations where: (*i*) a given network is deployed and executed, (*ii*) an attack is carried out along the simulation, (*iii*) some detection technique is configured in the network so that in case a malicious behavior is detected, (*iv*) a notification is generated, and (*v*) some response scheme is afterwards launched in order to recover the system.

In what follows we first introduce the fundamentals of NETA. After that, we briefly describe the response mechanism implemented in NETA. In particular, such a technique is based on an early version of the current DRNS proposal (see publication 9) as a proof of concept. Finally, we present a versatile framework developed within NETA, which allows the integration of heterogeneous defense solutions as part of a global security approach.

[3]NETA framework is developed by the *Network Engineering & Security Group* and it is available at http://nesg.ugr.es/neta.

C.7.1. NETA: A simulation framework for NETwork Attacks

Network security threats are continuously and quickly evolving [22, 72, 194], thus making the task of building defense mechanisms challenging. Security researchers are constantly offering defense solutions against new discovered threats. For that, they commonly need to develop their own implementation of the attack, generating a diversity of implementations with none or little adoption by the community. That way, it is desirable to have a common framework to develop and test network attacks and their corresponding defenses.

Motivated by this, we have developed NETA (*NETwork Attacks*). It is an OMNeT++ based network attacks framework, intended to provide a base reference framework to unify attacks, solutions and results. NETA is extensible and offers a high degree of versatility for the development of new and heterogeneous network attacks.

The main idea is to develop models in OMNeT++ implemented as new nodes to launch attacks: the *attacker nodes*. For that, the attacks are managed by the so-called *attack controllers*. These controllers manage one or more modules of a NETA attack node by sending *control messages*. These messages are sent by the attack controllers to specific modules that implement a modified behavior for the attack (for example, a packet dropping attack implies the modification of the NETA IPv4 module to get a malicious behavior). They are called *hacked modules* hereafter. For implementing this modified behavior, these hacked modules are inherited or replicated from INET (*INET Framework*)[4] modules and conveniently modified to obey the orders from the attack controllers.

The creation of an attacker node can be summarized as: (*i*) add to the associated `.ned` file the controllers related to the attacks to be executed, (*ii*) create the associated control messages and, (*iii*) substitute the modules needed by these attack controllers for the corresponding hacked modules. Figure C.31 shows the differences between a normal and an attacker node.

In a brief manner, the processes carried out by an attack controller inside an attacker node 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 attacker node by sending, at the time they will be initiated, activation messages which can contain configuration information.

[4]INET is an open-source model suite for wired, wireless and mobile networks for OMNeT++ and is available for download at https://inet.omnetpp.org/

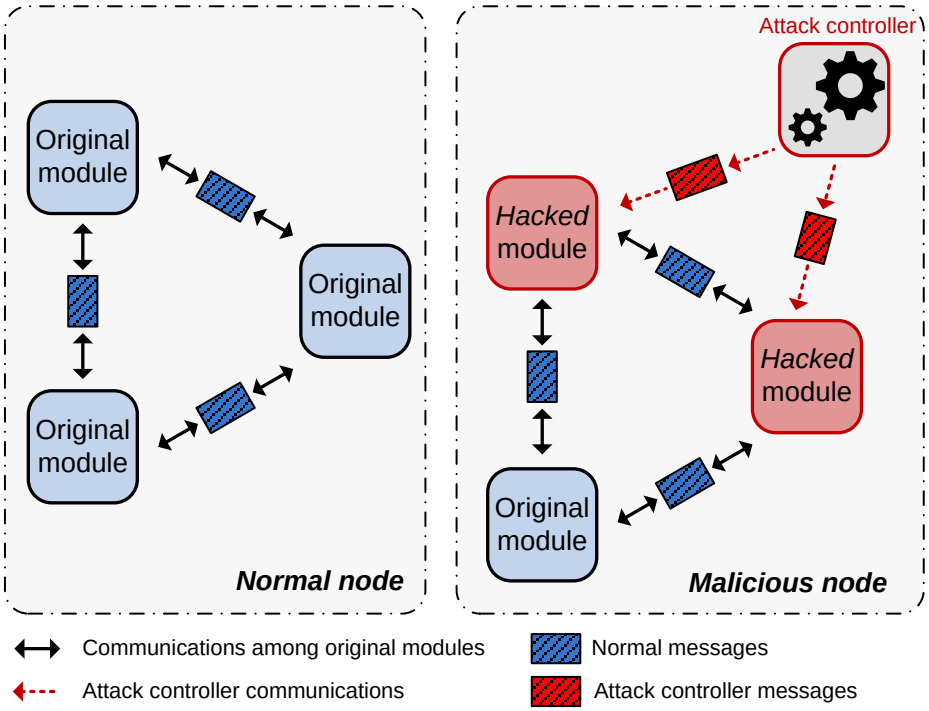


Figure C.31: Comparison between benign (normal) and attacker nodes in NETA.

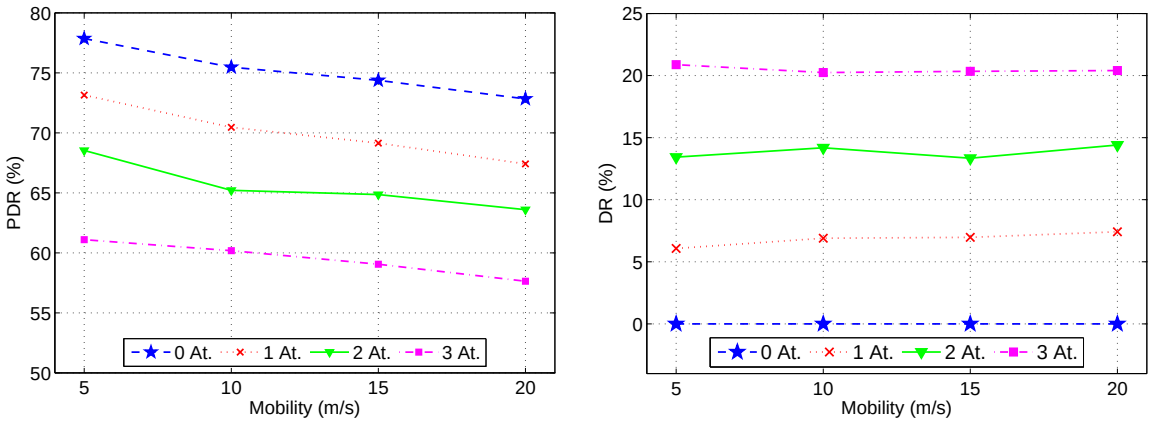
3. To deactivate the hacked modules in the attacker node by sending a deactivation message to finish the attack.

To improve the flexibility of the framework by allowing the execution of more than one attack simultaneously, just one hacked module per modified module will exist.

Implemented attacks: IP dropping attack

Through the developed NETA architecture, we implemented several attacks as a proof of concept. They are the *IP dropping*, *delay* and *sinkhole* attacks, which are well-known MANETs related attacks. In particular, in the case of dropping behavior, the malicious node will drop incoming packets with a fixed probability instead of forwarding them. This probability can be configured by the `droppingAttackProbability` parameter defined between 0 and 1. By default, it is set to 0 which makes de attacker node to behave normally (no dropping at all).

All of the attacks are evaluated in a realistic MANET scenario by means of the execution of several experiments. In the case of the IP dropping attack, the following performance metrics are defined:



(a) (b)

Figure C.32: IP dropping attack performance evaluation. PDR and DR evolution are shown in subfigures (a) and (b) respectively, as a function of the mobility of the nodes and the number of attackers.

- PDR (*Packet Delivery Ratio*): Total number of delivered data packets divided by total number of transmitted packets.
- DR (*Dropping Ratio*): Total number of data packets lost due to the execution of the attack divided by total number of transmitted data packets.

As we can see in Figure C.32, if the number of attackers is increased, PDR is deteriorated and DR rises up. Additionally, PDR decreases with the mobility[5], whereas DR remains nearly constant. This is due to the fact that the mobility increases packets lost due to collisions and channel errors, while it will remain constant due to the dropping attack. Further details about the attacks and their evaluation can be found in publication 10.

The obtained results validate NETA as a novel framework for developing heterogeneous network attacks. Moreover, it allows the integration of different and heterogeneous defense solutions like the response scheme developed here. In particular, we describe in the next subsection the implementation of an early version of DRNS as a response scheme in NETA.

[5]The UNs movements are following a RWP mobility pattern.

C.7.2. Integration of response/tolerant schemes

As in the real environment deployment, the DRNS solution presented in the next is centralized, which means that a central node exists together with the UNs and RNs. This central node is equipped with a `control` module that:

1. Is able to get the locations of all UN and RN.
2. Runs the DRNS optimization engine.
3. Sends the optimized target points to the RNs.

NETA offers a direct communication channel between the nodes deployed in the network, instead of that implemented in OMNeT++ by default. Because of that, the control module knows which and where all the nodes are. In fact, it acts over the RN `mobility` module, which determines how the node is moved. In our case, we implemented a specific mobility module named `mobilityRelay` that inherits from `LineSegmentMobilityBase` to provide linear movements towards a specific target point.

Thanks to this, the control module runs the DRNS engine and sends the (X,Y) optimized coordinates to each RN to drive them properly. Figure C.33 depicts the communications and messages exchanged between the central node and each different network node. The messages exchanged between the central node and the UNs (the blue ones) just transport the current UN locations. However, the RNs needs to know which are their next target points to go there, as well as the central node associated locations of the RNs. Such an information is sent within the corresponding messages (the red ones).

DRNS relies on several aspects determining its performance like, among others: (*i*) the DRNS engine execution time interval, which should be adjusted depending on the network dynamics; (*ii*) the own DRNS execution elapsed time; (*iii*) the coverage range, since small values imply more partitions in the networks; and (*iv*) the maximum velocity of the RNs. All of them are relevant and should be accordingly modified depending on the chosen simulation scenario.

Some experiments are carried out to validate the successful integration and operation of the response module. For that, we deploy a MANET scenario composed of 10 UNs while the number of RNs varies from 0 to 3. We also consider an area of 1000 m × 1000 m with a coverage range of 250 m for every node. The UNs are moving following a RWP mobility pattern configured such that the pause time is 15 s, and with a maximum velocity of 10 m/s. The RNs velocity is limited to 30 m/s. IEEE (*Institute of Electrical and Electronics Engineers*) 802.11 and AODV (*Ad hoc On-demand Distance Vector*) are selected as MAC (*Medium Access Control*) and routing

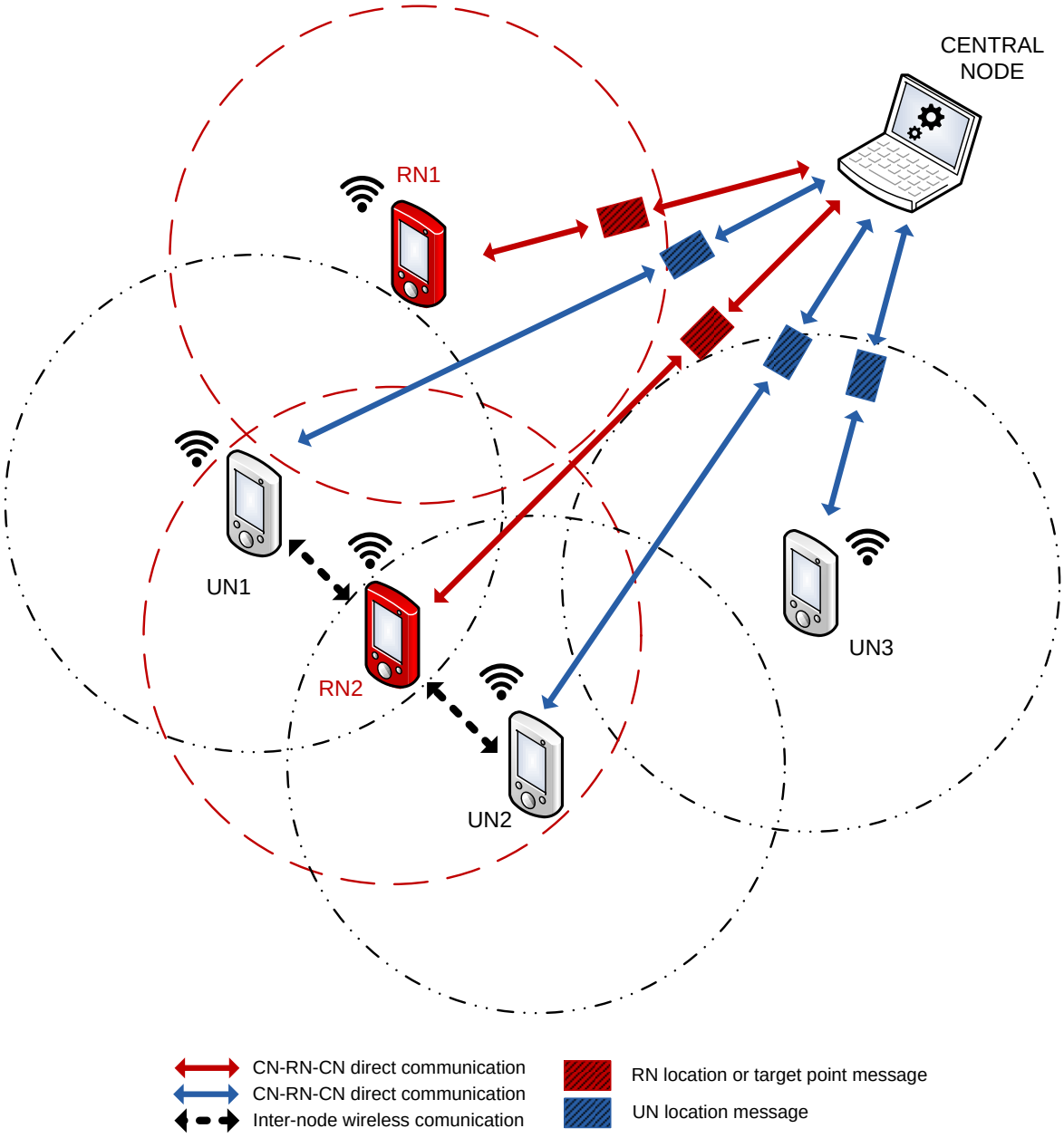


Figure C.33: Functional overview of the response/tolerant scheme implemented in NETA.

protocols, respectively. Ten CBR (*Constant Bit Rate*) traffic flows are configured for transferring information among UNs. For that, a UDP (*User Datagram Protocol*) burst (`UDPBasicBurst` application, in the context of the OMNeT++ simulator) is configured such that four packets of 512 bytes each are sent per second. Each experiment has a duration of 100 s and the results obtained correspond to the average PDR value through 50 repetitions.

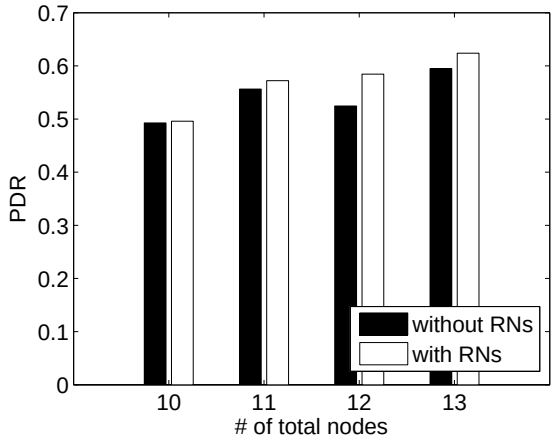


Figure C.34: PDR values obtained with and without the presence of RNs.

To fairly compare the network performance impact with and without RNs, we compute similar experiments with the same number of total nodes. The first one considers the presence of up to 3 of them, and the second one does not consider any of them. In Figure C.34 we can observe the average PDR values obtained in each case. Clearly, the network performance increases with the number of total nodes. We can also observe the improvement achieved when the RNs are deployed since they are driven by the DRNS optimization engine.

C.7.3. Towards global security solutions with NETA

As aforementioned, the present thesis work contributes with novel response/tolerant solutions. However, by just considering those kind of schemes are not enough for achieving survivable systems. For that reason, we present here an integration framework aimed at allowing a coherent integration of several defense solutions as a global security solution. Figure C.35 shows the overall architecture of the integration solution, with special emphasis in the necessary interaction mechanisms between the modules corresponding to the different defense lines.

The `NA_ModuleAdapter` adapter separates the modules it connects to in both senses, physical and functional. It allows to substitute any of the modules considered (detection, notification or response) easily. Just one message is exchanged among the modules and the `NA_ModuleAdapter`: `NA_AdapterMsg`. Hence, the communication between two modules connected through the `NA_ModuleAdapter` can be summarized as follows. Firstly, the source module generates a `NA_AdapterMsg` message encapsulating the corresponding information. This message is afterwards sent to the `NA_ModuleAdapter`, which is in charge of forwarding it to the destination module. Once the latter receives the message, the useful information from the

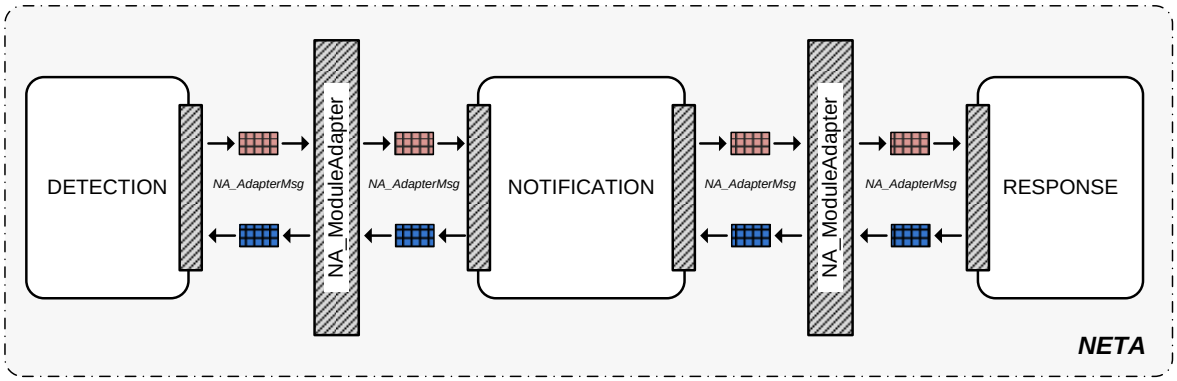


Figure C.35: Overall integration architecture and communication elements between the modules. The dashed modules and the depicted messages belong to the integration framework while the others could be any security scheme developed by the user.

`NA_AdapterMsg` can be extracted. How that message is sent among modules is part of the functionalities provided by the NETA framework, as described in Section C.7.1.

In order to validate the complete framework proposed, we deploy and integrate within NETA some specific solutions for detection, notification and response for the dropping attack. In regards to the detection solution, we followed the work in publication 4. That work detects packet dropping malicious behaviors according to a model established for forwarding process in MANETs. Once the attack is detected, the notification procedure described in publication 14 is launched. Finally, the module in charge of reacting against the attack detected introduced in Section C.7.2, is activated.

Although the notification module is able to manage different kinds of messages, for the sake of clarity, just one message is used and sent here to the response module: an *alert message* indicating the presence of one o more malicious nodes. Once the response module receives such an alarm, the malicious nodes will be no longer available and the RN movements will be controlled by the DRNS engine in order to mitigate the effects on the network performance.

Simulation scenario and preliminary results

Once defined the defense modules and the communication among them, we run several experiments to evaluate the PDR evolution when dropping attacks are taking place.

As in Section C.7.2, we also consider the same experimentation setup, just adding several malicious nodes. In particular, before the attack, the RNs remain static at

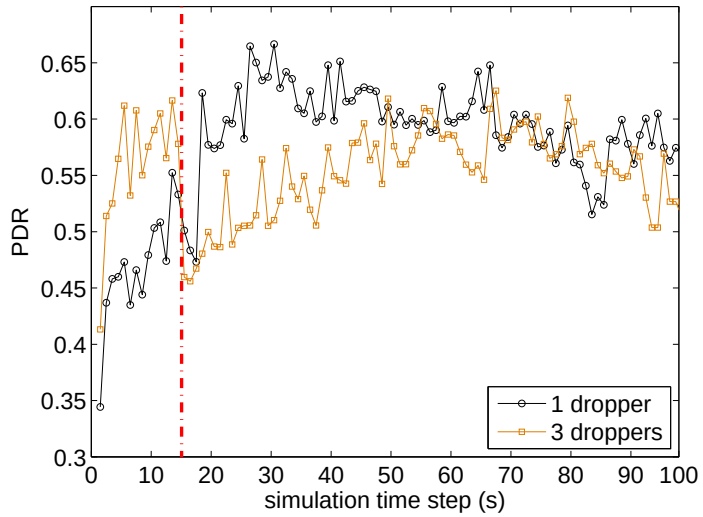


Figure C.36: PDR recovery when varying the number of dropper nodes with 3 RNs and 10 UNs, the last ones moving according to a RWP mobility pattern.

their initial positions. Once the alert message is received at the response module, the RNs will be driven by it.

The malicious nodes act as forwarding nodes since they are not the destination nor the source of the transmitted packets. Once the attack is activated, the nodes will not cooperate in the normal forwarding procedure anymore, dropping every packet received. The PDR evolution is shown in Figure C.36, where 10 UNs and 3 RNs are considered and the number of dropper nodes varies from 1 to 3. That figure shows that the attack takes place at the 15th second of the simulation. A vertical red dashed line highlights this event. At this moment the performance is reduced due to the action of the malicious nodes. Additionally, the bigger the number of dropper nodes the higher the effect on the network performance. Even though, the system is able to recover or even to improve the PDR values before the attack both for 1 and 3 droppers. Finally, is also remarkable that the response module provides a quicker recovery for low number of malicious nodes, since the damage caused in the network increases with the number of malicious nodes, as evident.

As a final consequence of the experimentation performed, we can conclude the goodness and validity of our overall integration proposal. However, further work should be done specially to measure how versatile and scalable it is.

Apéndice D

Conclusions and Future Work

Once described the contributions of the work in Appendix C, 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.

D.1. Conclusions

Ad hoc networks versatility relies on their special inherent characteristics. However, such desirable capabilities can become weaknesses from the point of view of security. More resilient, robust and finally survivable networks are needed indeed to combat current threats. Before providing solutions to solve the inherent security issues in ad hoc network, we have to perform the appropriate study of the current proposals in the literature. In this sense we have introduced:

- A study of the main and relevant threats in the context of ad hoc networks at present.
- A review of the current proposals against the principal threats in ad hoc networks, specially regarding response/tolerant solutions.
- A novel organization of the previous response/tolerant solutions, which are classified in three main groups: *node exclusion*, *node exclusion and announcement* and *node isolation*.

Once a review of the specialized literature is done, we will focus on the principal contributions of this work. They can be arranged (o grouped) in three topics: missing data imputation in WSNs, RNs location in MANETs, and security solutions integration. Regarding the missing data imputation solutions in WSNs, several achievements can be highlighted:

- We have introduced a multivariate-based missing data imputation scheme to fight against data integrity threats in critical scenarios like firefighting. Additionally, we also devise a monitoring and detection system that involves MSPC-based techniques to discern between environmental anomalies and malicious *data tampering* actuations.
- We have shown the influence of the data organization depending on the final application. That way, we develop a novel data arrangement, called *local models*, which increments the missing data recovery performance in comparison with the application of *global models*.
- We have demonstrated the relevant role of the underlying routing algorithm on the data recovery performance. To corroborate such an influence we propose and evaluate several routing strategies which, depending on the sensors compromised, could lead the system to different results.
- We developed a specific simulator to evaluate the solution. With it, we reproduce a firefighting scenario where the temperature variations can be gathered by the corresponding sensors.
- We performed extensive experiments to evaluate the missing data imputation method performance, not only in simulation but also in the LUCE real deployment.

In the case of the RNs location solutions in MANETs environments, some additional contributions can be highlighted:

- We developed and implemented a relay node placement approach to maximize the connectivity and throughput in such a kind of highly changing environments.
- We proposed a thorough mathematical formulation of the problem in order to address two main issues: (*i*) which are the best locations for the relay nodes, and (*ii*) how they should be moved towards them.
- From the previous mathematical formulation, we have developed and implemented a modular and versatile placement solution called DRNS (*Dynamical Relay Node placement Solution*). This approach is based on a previous work in the specialized literature for which several detected flaws are fixed. DRNS

is mainly based on the use of PSO-based optimization algorithms and MPC inspired methodologies to obtain the optimal positions and control the relay node movements.

- We addressed the problematic associated with this kind of methodologies by considering a fixed number of relay nodes, which complicates the problem but makes it more realistic.
- We stated that the module in charge of computing the AP locations for disconnected networks is in itself a suitable placement approach for static scenarios.
- We concluded the goodness of our proposal in terms of the performance goals. Even more, we also conclude the proper adaptability of the solution for changing environments.
- We tested our system as a valid response/tolerant system in the presence of malicious nodes. Specially for fighting against *dropping attacks*.
- We carried out a study about the computation complexity and execution time of the proposal. After that, we concluded the additional computation complexity introduced in comparison to other proposals, though a relevant system performance is also added.
- We successfully deployed the approach in a real MANET where the nodes are robots. For that, we devised and implemented a specific architecture to deal with several intrinsic aspects in such a kind of scenarios. Thus, we can conclude the practical feasibility of the system.

Finally, regarding security solutions integration, we can highlight the following contributions:

- We have developed and implemented a useful framework relying on a modular and versatile architecture.
- As a proof of concept, we have integrated several defense security schemes. Besides, a *dropping attack* is simulated to corroborate the successful integration and operation of the overall security approach. That way, we are able to detect and react against this kind of attacks.

D.2. Future work

The research of this thesis opens some interesting future work ideas. Among others, they are:

- To study new data arrangements to improve the recovery performance in WSNs by means of incrementing the correlation among the information in the imputation process. One potential approach would be to develop a local model sensor-wise, so that the recovery is particularized to the sensor.
- To evaluate the influence on the recovery performance in WSNs when using local modeling together with dynamic modeling and dynamic routing strategies.
- To tackle autonomous and automatic detection systems to discern the cause of the anomaly as part of a global security solution to combat data integrity attacks in WSNs. For that, discrimination analysis-based techniques can be used.
- To build a decentralized and distributed version of DRNS such that the system gets self-managed, scalable and resilient against faults.
- To design a new module to make DRNS adaptable to the environment changes. Such module should be able to dynamically recalculate the PSO algorithm parameters and meta-parameters depending on the network dynamics.
- To build more elaborated objective functions within DRNS to address relevant and realistic aspects in MANETs (*e.g.*, PDR, communications interferences, crowded node areas, node energy, network *goodput*, etc.).
- To systematically analyze how to measure the capability or capabilities of survivability in a objective and quantitative way, considering some special metrics in these kind of networks together. For instance, links capacities and lifetime, connectivity and throughput, node energy, etc.
- To use the previous information to provide global security schemes with capability of survivability. Such a systems should be able to measure this capability in a quantitative way. After that, they could optimize such a value through the use of multivariate techniques or those derived from the survival analysis.
- To extend solutions to more complex scenarios not limited to 2D spaces. For instance, the application in FANET environments, where a third dimension appears.
- To evaluate and benchmark alternative optimization algorithms in comparison with that used in DRNS.
- To develop and implement new attacks and defense solutions in the context of NETA and the integration framework developed here.