

UNIVERSIDAD NACIONAL DE CÓRDOBA

Facultad de Ciencias Exactas, Físicas y Naturales

Tesis Doctoral



Protocolos para el Procesamiento Distribuido
de Funciones de Umbral
en Redes Inalámbricas de Sensores

Autor: Ing. Guillermo Gastón Riva

Director: Dr. Ing. Jorge Manuel Finochietto

Junio de 2015

Protocolos para el Procesamiento Distribuido
de Funciones de Umbral
en Redes Inalámbricas de Sensores

por

Ing. Guillermo Gastón Riva

Dr. Ing. Jorge Manuel Finochietto
Director

Ing. Eduardo Alberto González
Co-Director

Comisión Asesora

Dr. Ing. Mario Rafael Hueda (CONICET, FCEFYN-UNC)
Dr. Ing. Julián Pucheta (FCEFYN-UNC)

Comisión Evaluadora

Dra. Ing. Cecilia Galarza (CONICET, FI-UBA)
Dr. Ing. Ariel Lutenberg (CONICET, FI-UBA)
Dr. Ing. Nazareno Aguirre (CONICET, UNRC)

Esta Tesis fué enviada a la Facultad de Ciencias Exactas Físicas y Naturales de la Universidad Nacional de Córdoba para cumplimentar los requerimientos de obtención del grado académico de Doctor en Ciencias de la Ingeniería

Córdoba, Argentina

Junio de 2015

Agradecimientos

Esta Tesis no habría sido posible sin el constante apoyo y estímulo del Dr. Ing. Jorge Manuel Finochietto, bajo cuya dirección se desarrolló la misma, y a quién le estoy infinitamente agradecido.

Agradezco a las siguientes instituciones que, con su valioso aporte técnico y/o financiero, posibilitaron el desarrollo de esta Tesis durante estos seis años de formación en investigación y desarrollo: a la *Agencia Nacional de Promoción Científica y Tecnológica, del Ministerio de Ciencia, Tecnología e Innovación Productiva (MINCyT) de la Nación*, al *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)*, a la *Universidad Tecnológica Nacional (UTN)*, a la *Universidad Nacional de Córdoba (UNC)*, y en especial al *Laboratorio de Comunicaciones Digitales (LCD) de la UNC*.

Agradezco a todas las personas que de una u otra forma apoyaron, aconsejaron, y colaboraron en el desarrollo de esta Tesis: al Ing. Eduardo Gonzalez (UTN), al Dr. Ing. Oscar Anunziata (UTN), al Dr. Ing. Mario Hueda (UNC), a la Dra. Ing. Graciela Corral Briones (LCD-UNC), a la Ing. Carmen Rodriguez (UNC), al Dr. Ing. Luis Godoy (UNC), al Dr. Ing. Diego Dujovne (UDP), al Dr. Ing. Julián Pucheta (UNC), al Dr. Ing. Guillermo Leguizamón (UNSL), a la Dra. Ing. Laura Alonso Alemany (FAMAF-UNC), y la Dra. Ing. Patricia Kisbye (FAMAF-UNC).

No puedo dejar de agradecer a mi Familia, pilar fundamental de mi vida, por haber confiado en mí, por brindarme su apoyo, y acompañarme durante estos seis años de arduo trabajo.

Finalmente, agradezco a todos los que de alguna u otra forma hicieron posible la realización de esta Tesis.



UNIVERSIDAD NACIONAL DE CORDOBA
Facultad de Cs. Exactas, Físicas y Naturales

ACTA DE EXAMENES

Libro: 00001 Acta: 03062 Hoja 01/01
LLAMADO: 1 08/06/2015
CATEDRA - MESA:

DI002 TESIS DOCTORADO EN CIENCIAS DE LA INGENIERIA

NUMERO	APELLIDO Y NOMBRE	DOCUMENTO INGRESO COND.	NOTA	FIRMA
27114727	RIVA, Guillermo Gastón	DNI: 27114727 2009 T	Aprobado	

GALARZA, Cecilia -
AGUIRRE, Nazareno -
LUTENBERG, Ariel -
CORRAL BRIONES, Graciela -

Observaciones:

Córdoba, 08/06/2015.

Certifico que la/s firma/s que ha/n sido puesta/s en la presente Acta pertenece/n a: Cecilia Galarza, Nazareno Aguirre y Graciela Corral Briones

Inscriptos Ausentes Examinados Reprobados Aprobados
02/06/2015 09:15:55 (0-3) (4-10)

Libro/Acta: 0000103062 Hoja: 01/01

Prefacio

Esta Tesis es presentada como parte de los requisitos para optar al grado académico de *Doctor en Ciencias de la Ingeniería de la Universidad Nacional de Córdoba*, y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otras. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el Laboratorio de Comunicaciones Digitales (LCD), Departamento de Electrónica, Facultad de Ciencias Exactas Físicas y Naturales (FCEFYN), Universidad Nacional de Córdoba (UNC), y en el Grupo de Ingeniería Clínica (GIC), Departamento de Ingeniería Electrónica, Universidad Tecnológica Nacional (UTN), Facultad Regional Córdoba, durante el período comprendido entre el 1 de Marzo de 2009 y el 8 de Junio de 2015, bajo la dirección del Dr. Jorge M. Finochietto.

Guillermo Gastón Riva

guilriva@gmail.com

griva@scdt.frc.utn.edu.ar

guillermo.riva@unc.edu.ar

Laboratorio de Comunicaciones Digitales
Facultad de Ciencias Exactas, Físicas y Naturales
Universidad Nacional de Córdoba
Córdoba, 8 de Junio de 2015

Resumen

El desarrollo de protocolos de comunicación basados en el *paradigma de enrutamiento y procesamiento centrado en datos* ha surgido en los últimos años como una novedosa y eficiente solución para reducir el costo de comunicación presente en las redes inalámbricas de sensores. El objetivo de estos protocolos es reducir este costo e incrementar el tiempo de vida de la red mediante el uso de *técnicas colaborativas de enrutamiento y computación distribuida de los datos*. Esto se debe fundamentalmente a que los nodos tienen energía limitada, al ser alimentados con baterías, y que la comunicación inalámbrica es el principal factor de consumo energético, pudiendo considerarse despreciable el consumo de energía tanto del sensado como del procesamiento de datos. Adicionalmente, al ser redes densas y con comunicación de datos a través de múltiples saltos, los costos de comunicación se incrementan exponencialmente al aumentar el tamaño de la red. A diferencia de las redes de comunicaciones de datos tradicionales, en las redes de sensores los nodos pueden ir procesando los datos de los paquetes mientras estos son retransmitidos en la red, y en general, se tiene un flujo de datos de uno a muchos nodos, es decir, desde un nodo coordinador hacia los nodos de la red, y de muchos a uno, es decir, desde muchos nodos sensores hacia el nodo coordinador.

En los últimos años, se han propuesto una gran variedad de protocolos de enrutamiento y procesamiento centrados en datos para la reducción del volumen de datos y del costo de comunicación para reportarlos al usuario, basados fundamentalmente en *agregación y fusión de datos*. Sin embargo, la computación distribuida de ciertas funciones, como por ejemplo las *funciones de tipo umbral (máximo/mínimo, k-máximos/mínimos, rango, etc)*, en las cuales solo algunos nodos poseen o pueden aportar argumentos útiles para el cálculo de dichas funciones, es más eficiente analizarlo como un *problema de búsqueda de datos o de información en la red*, que como un *problema de agregación y fusión de datos en el reporte de los datos hacia el usuario*. Este problema es desafiante si no se conoce a priori la localización de estos nodos. Sin embargo, debido a la dinámica presente tanto en estas redes, como en los parámetros a medir, no solo se requiere de una etapa inicial de descubrimiento de información, sino también de una actualización continua del conocimiento del estado de la red.

Adicionalmente, la mayor parte de los esquemas de procesamiento desarrollados para redes de sensores considera un flujo de datos en una dirección, en el que el modo de funcionamiento de la red es configurado inicialmente, y posteriormente los nodos comienzan con el reporte de sus datos hacia el usuario a través de un nodo coordinador cada una cierta cantidad de tiempo. Sin embargo,

debido a que las redes de sensores son redes muy variables, debido a fallas en nodos, inclusión de nuevos nodos a la red, pérdida de paquetes, movilidad de nodos, etc, este tipo de esquemas requiere de una actualización permanente del conocimiento del estado de la red por parte de cada nodo, con su consecuente costo de comunicación y de energía.

En este contexto, en esta Tesis se presentan las siguientes contribuciones. Se determina el esquema de procesamiento más adecuado para la *computación de funciones de tipo umbral* en redes inalámbricas de sensores, se define una arquitectura que define el funcionamiento de cada nodo de la red, y se desarrollan dos protocolos de procesamiento y filtrado de datos y de nodos vecinos. Se muestran indicios muy fuertes, mediante el uso de diferentes casos de análisis, que el esquema más adecuado para ser aplicado en el procesamiento de funciones de tipo umbral en redes inalámbricas de sensores es el *basado en requerimiento (interés, o demanda), y respuesta*, es decir, *basado en consulta/respuesta*. Cuando un usuario necesita obtener información de los datos obtenidos por los nodos sensores que conforman la red, este realiza una solicitud de información a la misma, inyectada a la red a través del nodo coordinador de la red o nodo sink. De esta forma, se muestrea el estado de la red en cada iteración de computación. Sin embargo, *se combina la diseminación del interés en la red, con heurísticas de búsqueda basadas en gradientes de información*, que posibilitan encontrar nodos con datos relevantes para la aplicación, y filtrar nodos que no aporten datos útiles. Mediante la implementación de *mecanismos de aprendizaje por refuerzo*, se adapta la búsqueda en forma iterativa, reduciendo el área de exploración, con el objetivo de minimizar tanto el costo de comunicación, como el error de computación. Se presenta el diseño de una arquitectura que define el funcionamiento de cada nodo y que permite entender el comportamiento de los mismos. Además, se proponen y desarrollan dos protocolos de procesamiento y filtrado de nodos vecinos. El primero de ellos está basado en la *metaheurística de recocido simulado, denominado PASA, o sin estados*. El segundo, por otro lado, está basado en *niveles de feromonas, denominado PhINP, o con estados*.

Como resultado de la evaluación de los protocolos propuestos en diferentes escenarios y condiciones de funcionamiento, se puede comprobar el beneficio del uso de dichos protocolos basado en la reducción significativa del costo de comunicación, de alrededor del 60 %, llegando a triplicar el tiempo de vida de la red respecto a esquemas tradicionales como Flooding y Gossip, manteniendo una baja tasa de error relativo de computación, no mayor al 5 %. Además, se puede verificar la robustez de estos protocolos ante condiciones no deseadas en la red, y las capacidades de adaptación.

Palabras clave: Redes inalámbricas de sensores, procesamiento y filtrado distribuido en la red, computación de funciones de tipo umbral, enrutamiento de paquetes centrado en datos, reporte de datos basado en requerimientos de información o de consulta/respuesta, búsqueda de información basada en gradientes, algoritmos metaheurísticos de búsqueda, recocido simulado, inteligencia computacional basada en niveles de feromonas de colonias de hormigas, aprendizaje automático por refuerzo.

Abstract

The development of communication protocols based on the *data-centric routing and processing paradigm* has emerged in the last years as a novel and efficient solution for the reduction of the communication cost existing in the wireless sensor networks. The goal of these protocols is to reduce this cost and increase the network lifetime by using collaborative routing and distributed data computation techniques. This is mainly due sensor nodes are battery powered, and as a consequence, they have limited energy, the wireless communication is the main energy consumption factor, and the sensing and processing consumption can be considered negligible. Additionally, being dense networks and with data communications across multiple hops, communication costs increase exponentially when increasing the network size. Unlike traditional communications networks, in sensor networks nodes can process data packets while these data are broadcast on the network, and generally has a communication flow from one to many nodes, that is, from a coordinator to sensor nodes, or from many to one, that is, from many sensor nodes to the coordinator node.

In recent years, there have been proposed a variety of data-centric routing and processing protocols to reduce the data and the communication cost to report them to the user, mainly based on *aggregation and data fusion*. However, the distributed computation of certain functions, such as the *type-threshold functions (maximum / minimum, k-max/min, range, etc.)*, in which only some nodes possess or can provide useful arguments to calculate these functions, it is more efficient to analyze it as a *search problem of data or information on the network*, that as a *problem of data aggregation and fusion during data reporting to the user*. This problem is challenging if we not known a priori the location of these nodes. However, due to the dynamics present in both these networks, and the parameters to be measured, not only it requires an initial stage of information discovery, but also a continuous knowledge updating of the network state.

Additionally, most of the processing schemes developed for sensor networks consider a data flow in one direction in which the operating mode of the network is initially set, and then nodes begin to report its data to the user through a coordinator node at a certain amount of time. However, because the sensor networks are highly variable networks, due to faulty nodes, adding new nodes to the network, packet loss, mobility of nodes, etc., such a scheme requires a constant knowledge updating of the state of the network by each node, with the consequent communication cost and energy.

In this context, this thesis presents the following contributions. It is determined the processing

scheme more appropriate for the *computation of type-threshold functions* in wireless sensor networks, it is defined an architecture that defines the operation of each network node, and there are developed two processing and filtering protocols. The present work reports strong evidences, by using different analysis cases, that the most suitable scheme for type-threshold function processing in wireless sensor networks is the *based on requirement (interest, or demand) and response*, ie, *based on request / reply*. When a user needs to get information from the data obtained by the sensor nodes in the network, this makes a request for information, injected into the network through the coordinator node or sink node. Thus, the state of the network is sampled at each computing iteration. However, *the interest dissemination in the network is combined with heuristic search based on information gradients*, which allow to find nodes with data relevant to the application, and filter nodes that do not provide useful data. By implementing *reinforcement learning mechanisms*, the search is adapted iteratively, reducing the area of exploration, with the aim of minimizing both the cost of communication, such as the computing error. It is designing an architecture that defines the operation of each node and allows to understand the behavior of them. In addition, there are proposed and developed two protocols for processing and filtering of neighboring nodes. The first one is based on the *simulated annealing metaheuristic, called PASA or stateless scheme*. The second, on the other hand, is based on *pheromone levels, called PhINP or stateful scheme*.

As a result of the evaluation of the proposed protocols in different scenarios and operating conditions, we can verify the benefit of using these protocols based on the significant reduction in the communication cost close to 60 % to triple the network lifetime compared to traditional schemes like Flooding and Gossip, maintaining a low computing error rate lower than 5 %. Also, we can check the robustness of these protocols to unwanted network conditions, and the coping skills.

Keywords: Wireless sensor networks, distributed in-network processing and filtering, type-threshold function computing, data-centric routing, data reporting based on information requirements or based on query / response, information search based on gradients, metaheuristics search algorithms, simulated annealing, computational intelligence based on pheromone levels of ant colonies, reinforcement learning.

Zusammenfassung

Die Entwicklung von Kommunikationsprotokollen auf Basis des *datenzentrischen Routing und Verarbeitung Paradigmen* hat in den letzten Jahren als eine neue und effiziente Lösung für die Reduzierung der Kommunikationskosten in den drahtlosen Sensornetzwerken entstanden. Das Ziel dieser Protokolle ist es, diese Kosten zu reduzieren und die Netzwerk Lebensdauer durch Verwendung kooperative Weiterleitung und verteilten Datenberechnungstechniken zu inkrementieren. Dies ist vor allem, weil die Sensorknoten sind batteriebetrieben, und als Folge davon haben sie Energie begrenzt, wobei drahtlose Kommunikation ist der Hauptenergieverbrauchsfaktor und die Erfassungs- und Verarbeitungsverbrauch kann als vernachlässigbar angesehen werden. Zusätzlich, da sie dichte Netzwerke mit Multihop Kommunikation sind, die Kommunikationskosten erhöhen exponentiell mit zunehmender Grösse des Netzwerks. Im Gegensatz zu herkömmlichen Kommunikationsnetzen, in Sensornetzwerken die Knoten haben die Möglichkeit die Datenpaketen zu verarbeiten, während sie in an das Netzwerk weitergeleitet werden, und im allgemeinen gibt es einen Datenfluss von einem zu vielen Knoten (aus einem Koordinator zu Sensorknoten), oder von vielen Knoten zu einem (aus vielen Sensorknoten zu Koordinator).

In den letzten Jahren wurde eine Vielzahl von Routing-Protokollen und datenorientierte Verarbeitung vorgeschlagenen, die auf die Verringerung des Volumens der Daten und die Kommunikationskosten zu konzentrieren, die hauptsächlich auf *Aggregation und Datenfusion* basiert. Jedoch die verteilte Berechnung bestimmter Funktionen, wie die *Schwellwertfunktionen (Maximum / Minimum, k-max / min, Bereich, etc.)*, in dem nur einige Knoten besitzen oder kann bieten nützlich Argumente um diese Funktionen zu berechnen, ist effizienter dieses Problem als *Suchproblem von Daten oder Informationen Netzwerk über das Netzwerk* das als *Problem der Daten Aggregation und Fusion* zu analysieren. Dieses Problem ist schwierig, wenn man die Standort dieser Knoten nicht bekannt. Jedoch aufgrund der Dynamik präsentieren in diesen beiden Netzwerken and messenden Parameter, ist es nicht nur eine ersten Anfangsstufe der Informationsermittlung, sondern auch eine kontinuierliche Kenntnis Aktualisierung der Kenntnisse des Netzwerkstatus.

Zusätzlich sind die meisten der Verarbeitungsschemata für Sensornetzwerke entwickelt als Datenfluss in einer Richtung, wobei die Betriebsart der Netzwerk wird zunächst konfiguriert und dann die Knoten senden ihre Daten durch einen Koordinationsknoten zu ein Benutzer bei einer bestimmten Menge an Zeit. Da die Sensornetzwerke sind sehr variabel Netzen aufgrund von fehlerhaften Knoten,

die Hinzufügung neuer Knoten zu dem Netzwerk, Paketverlust, Knotenmobilität, usw., diese Systeme erfordert ständig aktualisierten Kenntnisse über den Zustand des Netzwerks von jedem Knoten mit der daraus folgenden Kommunikationskosten und Energie.

In diesem Zusammenhang stellt dieser Arbeit folgende Beiträge. Es wird bestimmt, das Verarbeitungsschema, mehr für die *Berechnung Typ Schwellwertfunktionen* in drahtloses Sensornetzwerke Falls wird es definiert eine Architektur, die den Betrieb jeder Netzknoten definiert, und entwickelt zwei Verarbeitung und -filterung Protokolle. Die vorliegende Arbeit berichtet starke Beweise, durch die Verwendung verschiedener Analyse Fällen, dass die am besten geeignete System für den Typ-Schwellenfunktion Verarbeitung in drahtlosen Sensornetzwerken ist die *basierend auf Anforderung (Interesse oder Bedarf) und Antwort*, das heisst, *basierend auf Anfrage / Antworten*. Wenn ein Benutzer benötigt, um Informationen aus den von den Sensorknoten im Netzwerk erhaltenen Daten zu erhalten, macht dies eine Anforderung für Information, durch die Koordinatorknoten oder Senkenknoten in das Netz eingespeist. Somit wird der Zustand des Netzwerks bei jedem Rechen Iteration abgetastet. Allerdings, die Knoten mit anwendungsrelevanten Daten und Filterknoten, die nützliche Daten bieten nicht finden lassen *wird die Zinsweitergabe im Netz mit heuristische Suche basierend auf Informationen Gradienten kombiniert*. Durch die Implementierung von *Verstärkungslernmechanismen*, wird die Suche iterativ wie das Rechenfehler angepasst, wodurch der Bereich der Exploration, mit dem Ziel, sowohl die Kosten der Kommunikation, zu minimieren. Es ist die Gestaltung eine Architektur, die den Betrieb jeder Knoten definiert und erlaubt es, das Verhalten zu verstehen. Zusätzlich sind vorgeschlagen und entwickelt zwei Protokolle zur Verarbeitung und Filterung von Nachbarknoten. Die erste basiert auf der basiert auf *Metaheuristik Simulated Annealing, genannt PASA, oder Staatenlosen*. Der zweite, auf der anderen Seite wird auf *Pheromon Ebenen, genannt PhINP oder Stateful* basierend.

Als Ergebnis der Auswertung der vorgeschlagenen Protokollen in verschiedenen Szenarien und Betriebsbedingungen kann man den Vorteil der Verwendung dieser Protokolle auf der Basis der signifikanten Verringerung der Kommunikationskosten, nahezu 60 %, um die Netzwerk Lebensdauer verdreifachen Vergleich zu herkömmlichen Schemes wie Flooding und Gossip, die Aufrechterhaltung einer geringen Rechenfehlerrate nicht mehr als 5 %. Ausserdem können wir die Robustheit dieser Protokolle, um unerwünschte Netzwerkbedingungen und den Bewältigungsstrategien zu überprüfen.

Stichworte: Drathlose Sensornetzwerke, verteilte Verarbeitung und Filterung im Netzwerk, Typ-Schwellenfunktion Computing, datenzentrischen Routing, Datenberichterstattung auf Basis von Informationsanforderungen oder auf Basis von Frage / Antwort, Informationssuche basierend auf Gradienten, Metaheuristiken Suchalgorithmen, Simulated Annealing, Computational Intelligence auf Basis von Pheromon Ebenen der Ameisenkolonien, Lernen durch automatische Verstärkung

Índice general

Índice de figuras	V
Índice de tablas	VII
Glosario	IX
Siglas	XI
Nomenclatura	XIV
1 Introducción	1
1.1 Problema de Investigación	3
1.1.1 Redes Inalámbricas de Sensores	5
1.1.2 Procesamiento de Datos en WSNs	7
1.1.3 Origen y Evolución de las WSNs	9
1.1.4 Aplicaciones	10
1.1.5 Limitaciones	12
1.1.6 Técnicas de Reducción de Energía	13
1.2 Relevancia del Problema	15
1.3 Hipótesis y Metodología de Trabajo	16
1.4 Objetivos	19
1.5 Contribución	20
1.6 Publicaciones Generadas	20
1.6.1 Aportes	21
1.7 Estructura General de la Tesis	25

2 Marco Teórico: Modelos	27
2.1 Modelos de Datos	28
2.2 Modelos de Computación y Procesamiento	31
2.2.1 Definiciones de Términos	31
2.2.2 Procesamiento en redes tradicionales y en WSNs	32
2.2.3 Tipo de Funciones Analizadas	38
2.2.4 Funciones de Umbral	40
2.3 Modelos de Enrutamiento	40
2.3.1 Modelos de Reporte de Datos	45
2.4 Modelos de Energía	46
2.5 Modelos de Conectividad y Comunicación	47
2.6 Comentarios de Cierre	49
3 Procesamiento Local	51
3.1 Introducción	51
3.2 Metaheurísticas	54
3.2.1 Tipos de Metaheurísticas	54
3.3 Aprendizaje por Refuerzo	56
3.4 Elección de Estrategia de Operación	57
3.5 Esquema de Procesamiento Propuesto	58
3.6 Protocolos de Filtrado en Red	60
3.7 Estrategias de Procesamiento	61
3.8 Procedimientos	61
3.9 Arquitectura	62
3.9.1 Reglas de Decisión y Filtrado	63
3.10 Comentarios de Cierre	64
4 Protocolo de Procesamiento sin Estados	65
4.1 Introducción	65
4.2 Funcionamiento	66
4.2.1 Diseminación de Requerimientos	67
4.2.2 Reporte de Respuestas	70
4.2.3 Aprendizaje	71

4.3	Comentarios de Cierre	76
5	Protocolo de Procesamiento con Estados	77
5.1	Introducción	77
5.2	Funcionamiento	78
5.2.1	Diseminación de Requerimientos	80
5.2.2	Reporte de Respuestas	80
5.2.3	Aprendizaje	82
5.3	Comentarios de Cierre	83
6	Análisis de los Resultados	85
6.1	Métricas Analizadas	86
6.2	Parámetros de Configuración	86
6.3	Protocolos Utilizados para Comparación	87
6.4	Análisis del Protocolo sin Estados	88
6.4.1	Comparación con otros Protocolos	88
6.4.2	Cambio de Amplitud	89
6.4.3	Falla de Nodos	90
6.4.4	Paquetes Perdidos	90
6.5	Análisis del Protocolo con Estados	90
6.5.1	Comparación con otros Protocolos	91
6.5.2	Falla de Nodos	92
6.5.3	Paquetes Perdidos	92
6.5.4	Tiempo de Vida de la Red	92
6.5.5	Monitoreo con Resolución Finita	94
6.6	Análisis de la Comparación entre Protocolos	94
6.6.1	Convergencia	95
6.6.2	Análisis de Eventos	97
6.6.3	Detección de Evento Único	98
6.6.4	Robustez ante Fallas de Nodos	98
6.7	Comparación de los Protocolos	101
6.8	Comentarios de Cierre	101
7	Conclusiones	103

Índice de figuras

1.1	Nodo sensor que conforma redes inalámbricas de datos	6
1.2	Elementos que conforman una red inalámbrica de sensores	6
1.3	Redes de comunicaciones con enlaces directos, y con enlaces multisalto	8
1.4	Esquema de uso considerado para la red de sensores basado en pregunta y respuesta	9
1.5	Evolución de las redes inalámbricas de sensores	10
1.6	Aplicaciones de redes inalámbricas de sensores	11
1.7	Distribución del consumo de energía en un nodo sensor	12
1.8	Distribución de datos en una red de sensores representada mediante isocontornos	18
1.9	Estructura general de la tesis	26
2.1	Modelos utilizados	28
2.2	Modelos de fuentes de datos utilizados	30
2.3	Representación de la magnitud a sensar	31
2.4	Elementos de un paquete que pueden procesados por un nodo retransmisor	33
2.5	Reporte de datos en WSNs	36
2.6	Modelos de reporte de datos	45
2.7	Modos de comunicación	48
2.8	Esquema de múltiples saltos respecto a un salto de comunicación	49
2.9	Cantidad mínima de vecinos necesaria para asegurar la conectividad de la red	49
3.1	Métodos de optimización	55
3.2	Aprendizaje por refuerzo	57
3.3	Comparación de costos de comunicación de diferentes estrategias de pregunta/respuesta	58

3.4	Procesos involucrados en la computación de una función	59
3.5	Arquitectura de filtrado, procesamiento y toma de decisiones instanciada en cada nodo	63
4.1	Tipos de comportamientos de los nodos durante la diseminación de requerimientos . .	69
4.2	Mecanismos de reporte de datos	72
4.3	Performance del protocolo sin estados para diferentes valores de T_0	74
4.4	Filtrado iterativo con reducción del espacio de búsqueda desde A hasta E.	75
4.5	PASA: procesos de enfriamiento	75
4.6	PASA: convergencia	76
5.1	Tipos de comportamiento de los nodos durante la diseminación de requerimientos . . .	81
5.2	Distribución de los niveles de feromona en cada nodo luego de la convergencia del algoritmo	83
6.1	Entorno de simulación desarrollado en OMNeT++	87
6.2	Comparación con protocolos existentes	89
6.3	Error de computo ante cambios de amplitud de los eventos	89
6.4	Error de computo ante fallas de nodos	90
6.5	Error de computo ante pérdida de paquetes	91
6.6	Comparación con protocolos existentes	91
6.7	Error de computo ante fallas de nodos	92
6.8	Error de computo ante pérdida de paquetes	92
6.9	Tiempo de vida de la red	93
6.10	Capacidad de detección con resolución finita	94
6.11	Convergencia de protocolos sin estados y con estados	96
6.12	Análisis de eventos de protocolos sin estados y con estados	97
6.13	Detección de un evento para diferentes protocolos.	98
6.14	Robustez ante falla de nodos	99
6.15	Diseminación de requerimientos y de respuestas	100
6.16	Mecanismo de auto-curado para hacer frente a fallas en nodos	100

Índice de tablas

2.1	Características de los diferentes tipos de esquemas de procesamiento.	34
6.1	Características de los protocolos tradicionales utilizados.	88
6.2	Parámetros de configuración usados en las simulaciones.	95
6.3	Comparación de los protocolos propuestos.	101

Glosario

Ad-hoc Término latino que significa "para esto". En redes de comunicación, una red ad hoc es aquella (especialmente inalámbrica) en la que no hay un nodo central coordinador, sino que todos los dispositivos están en igualdad de condiciones. Ad hoc es el modo más sencillo para el armado de una red. Son por ejemplo redes ad hoc las que se crean de forma espontánea, sin una infraestructura específica y funcionando en un espacio y tiempo limitados.

Cognitivo Una red cognitiva es un tipo de red de datos que hace uso de tecnologías de diversos campos de investigación (autoaprendizaje, representación del conocimiento, gestión de red, etc) para resolver los problemas actuales. Una red cognitiva es aquella capaz de percibir las condiciones actuales para planificar, decidir y actuar según esas condiciones, aprendiendo de las consecuencias de sus actos mientras sigue un plan preestablecido. El sistema aprende de los resultados de sus acciones pasadas para mejorar su comportamiento futuro. Esta definición de red cognitiva no menciona de forma explícita el conocimiento de la red, solo describe el bucle cognitivo y los objetivos finales.

Contención Término que hace referencia a la forma en la que dispositivos de comunicaciones acceden al canal de comunicación cuando se encuentran en la misma banda de frecuencia y en el mismo rango de comunicación.

Informacion Conjunto de datos procesados que puede ser utilizado para obtener algún tipo de conocimiento.

Pervasivo La computación pervasiva, o inteligencia ambiental, se refiere al desarrollo de sistemas computacionales para usuarios móviles con aparatos móviles, en los cuales la integración de los sistemas en la vida diaria es la más transparente posible. Esto requiere de sistemas que sepan captar su ambiente de ejecución y adaptarse a ello, en forma dinámica. Programar dichos sistemas con lenguajes tradicionales implica un nivel de complejidad enorme para manejar todos los detalles relacionados con la naturaleza volátil del ambiente y de las conexiones.

Protocolo de Caminata aleatoria (Random Walk) Protocolo estocástico de búsqueda basado en la selección aleatoria de caminos, utilizando comunicación unicast. Es decir, cada nodo que recibe un paquete decide aleatoriamente a cual de sus nodos vecinos se lo reenvía.

Protocolo de Expansión por Anillo (Expanding Ring Search) Protocolo de búsqueda basado en expansión progresiva del espacio de búsqueda, utilizado principalmente en aplicaciones en la que se buscan eventos producidos en la red, y en las que no se conoce su localización. Inicialmente, el nodo que realiza la búsqueda de un evento de interés, lo hace entre sus nodos vecinos que se encuentran a un salto de distancia. En caso de encontrar el evento, termina la búsqueda, o en caso contrario, realiza una nueva búsqueda a dos saltos de distancia. De esta forma, se realiza progresivamente la búsqueda en forma de anillos concéntricos, considerando el nodo que realiza la búsqueda como centro de los anillos, o con una posición igual a cero.

Protocolo de Inundación Controlada (Controlled Flooding) Protocolo simple de comunicación utilizado para enviar paquetes a todos los nodos de una red de múltiples saltos. La versión controlada permite evitar el problema de explosión de comunicación en el que incurre la versión tradicional, mediante la condición de que cada nodo puede retransmitir solamente una vez el mismo mensaje.

Protocolo de Rumoreo (Gossiping) Protocolo estocástico de búsqueda inspirado en el comportamiento social de diseminación de un rumor. Existen muchas variantes de este tipo de protocolo, pero en general, cada nodo de la red transmite su dato a un nodo vecino aleatorio cada una cierta cantidad de tiempo.

Ubicuo Término latino que significa "en todas partes". En el área de la tecnología, se dice que la conexión es ubicua cuando podemos estar conectados a una red en todo momento, sin importar el lugar, o que la computación es ubicua, cuando la misma puede ser realizada en cualquier lugar. El término computación ubicua se refiere a la integración de la informática en el entorno de las personas.

Siglas y Acrónimos

AC Address Centric.

ACO Ant Colony Optimization.

ARPANET Advanced Research Projects Agency Network.

BAN Body Area Network.

BCO Bee Colony Optimization.

CADR Constrained Anisotropic Diffusion Routing.

CI Computational Intelligence.

DARPA Defense Advanced Research Projects Agency.

DC Data Centric.

DD Directed Diffusion.

DSN Distributed Sensor Network.

DTN Delay/Disruption-Tolerant Network.

ERS Expanding Ring Search.

GPU Graphical Processing Unit.

GSM Global System for Mobile Communications.

IDSQ Information-Driven Sensor Querying.

IEEE Institute of Electrical and Electronics Engineers.

IoT Internet of Things.

IP Internet Protocol.

Kb Kilobyte.

LAN Local Area Network.

LR-PAN Low-Rate Personal Area Network.

MAC Multiple Access Control.

MANET Mobile Ad-Hoc Network.

MEMS Microelectromechanical System.

MIT Massachusetts Institute of Technology.

NP-Hard Non-deterministic Polynomial-time hard.

OS Operating System.

PAN Personal Area Network.

PASA Parallel Adaptive Simulated Annealing.

PhINP Pheromone-based In-Network Processing.

PSO Particle Swarm Optimization.

QoS Quality of Service.

RL Reinforcement Learning, Aprendizaje por refuerzo.

RX Reception.

SA Simulated Annealing.

SI Swarm Intelligence.

TAG Tiny Aggregation Service.

TX Transmission.

USN Underwater Sensor Network.

VAN Vehicular Area Network.

WSN Wireless Sensor Network.

Nomenclatura

α	Coefficiente de difusión espacial de una fuente de datos.
ΔE	Diferencia entre el valor de umbral del mensaje de requerimiento y el valor sentido por el nodo.
\hat{X}'	Conjunto de valores relevantes para la aplicación sentidos y reportados en la red (argumentos).
\hat{x}'	Valor sentido por un nodo.
λ	Nivel de feromona en un nodo.
λ_{dec}	Nivel de decremento de feromona en un nodo.
λ_{inc}	Nivel de incremento de feromona en un nodo.
λ_{max}	Nivel máximo de feromona en un nodo.
λ_{min}	Nivel mínimo de feromona en un nodo.
ρ	Densidad de nodos en la red [nodos/ m^2].
$\sigma(\cdot)$	Función de permutación.
$\underline{f}(\cdot)$	Función que representa la distribución espacial de la magnitud física sensada.
\underline{x}_s	Posición sobre el eje \underline{x} del nodo s .
\underline{y}_s	Posición sobre el eje \underline{y} del nodo s .
c_q	Costo de comunicación durante el requerimiento de información.
c_r	Costo de comunicación durante la respuesta o reporte de datos.
c_T	Costo total de comunicación.
$corr$	Correlación espacial o temporal de los datos.
D	Valor de decremento de temperatura usado en el protocolo PASA.
$d(\cdot)$	Distancia euclídea entre nodos, o entre un nodo y una fuente de datos [metros].

e	Error relativo de computación.
E_P	Energía consumida en el procesamiento en un nodo [joules].
E_R	Energía remanente en un nodo [joules].
E_S	Energía consumida en el sensado en un nodo [joules].
E_T	Energía total consumida en un nodo [joules].
E_{RX}	Energía consumida en la recepción de datos en un nodo [joules].
e_{sens}	Error de sensado.
E_{TX}	Energía consumida en la transmisión de datos en un nodo [joules].
F	Cantidad de funciones que modelan la distribución de datos.
$f(\cdot)$	Función que se quiere computar.
$g(\cdot)$	Función de filtrado implementada en la red.
hop_level	Distancia mínima en saltos de comunicación de un nodo respecto al sink.
i	Número de iteración de computación.
m	Cantidad de argumentos relevantes provistos por los nodos para computar una función.
$maxDist$	Distancia máxima en cantidad de saltos entre el nodo sink y el nodo mas distante de la red.
N	Grado de la red, o número medio de vecinos por nodo.
n	Número total de nodos en la red.
q_r	Cantidad de paquetes recibidos por un nodo.
q_t	Cantidad de paquetes transmitidos por un nodo.
r	Rango de comunicación de un nodo [metros].
s	Número de nodo.
T	Temperatura del algoritmo PASA dentro de un paquete de requerimiento.
T_0	Temperatura inicial del algoritmo PASA.
Th	Valor umbral actual transportado en el mensaje.
X	Conjunto de total de valores sensados en la red.
X'	Conjunto de valores relevantes para la aplicación sensados en la red (argumentos).
x'	Valor real de la magnitud en la posición del nodo.

Introducción

Contenido

1.1	Problema de Investigación	3
1.1.1	Redes Inalámbricas de Sensores	5
1.1.2	Procesamiento de Datos en WSNs	7
1.1.3	Origen y Evolución de las WSNs	9
1.1.4	Aplicaciones	10
1.1.5	Limitaciones	12
1.1.6	Técnicas de Reducción de Energía	13
1.2	Relevancia del Problema	15
1.3	Hipótesis y Metodología de Trabajo	16
1.4	Objetivos	19
1.5	Contribución	20
1.6	Publicaciones Generadas	20
1.6.1	Aportes	21
1.7	Estructura General de la Tesis	25

EN los últimos años hemos venido asistiendo a un incremento vertiginoso en el uso de las comunicaciones inalámbricas por parte de la sociedad como consecuencia de la evolución de los sistemas de comunicación. En este sentido, se han ido sustituyendo sistemas fijos de alta potencia por redes de comunicaciones con dispositivos fijos y móviles más eficientes energéticamente. Estas últimas incluyen desde la adopción masiva de tecnologías de telefonía móvil como GSM, el uso de tecnologías como WiFi para acceso a redes de datos, hasta el reemplazo del cableado de periféricos mediante Bluetooth. Estas tecnologías permiten a las personas contar con un permanente acceso a la información, al posibilitarles la conexión a redes de datos. En definitiva, estamos experimentando una verdadera revolución, no sólo en la forma de comunicarnos, sino también en la posibilidad de crear novedosas aplicaciones que den respuesta a necesidades actuales y futuras de la sociedad.

Afortunadamente, los avances realizados en los últimos años, tanto en la miniaturización de la electrónica (sensores MEMS), como en el desarrollo de sistemas de procesamiento y de comunicación inalámbricos de bajo consumo, y en la implementación de sistemas de alimentación de energía más eficientes, posibilitaron el desarrollo de las *redes inalámbricas de sensores* (RIS, o WSNs -Wireless Sensor Networks-). Estas son redes conformadas por *nodos* con capacidades sensitivas, de procesamiento, de almacenamiento, y de comunicación inalámbrica de datos, que posibilitan monitorear de forma distribuida magnitudes físicas del entorno [1, 2]. Este tipo de redes han emergido como una solución para cerrar la brecha entre el mundo físico, químico y biológico, y el mundo de la información de Internet.

Los sensores que utilizan los nodos son muy variados, y dependen de la aplicación particular en la que se los utilice. Estos pueden ser de temperatura, de humedad, de luminosidad, de presión, de velocidad, de aceleración, de sustancias químicas, de imagen, etc. En la actualidad, el espectro de sensores disponibles comercialmente es muy amplio. Por otro lado, la incorporación de microcontroladores de mayores prestaciones y menor consumo energético en los nodos, posibilita que los mismos puedan realizar cierto tipo de procesamiento de los datos y aplicar alguna técnica de aprendizaje. Finalmente, el desarrollo tanto de transceptores de radio de bajo consumo, como de protocolos de comunicación eficientes energéticamente, posibilita a los nodos la formación ad-hoc de redes de comunicaciones a través de las cuales obtener y reportar datos o información. En este sentido, el término *ad-hoc* se refiere a que estas son redes desplegadas para cumplir con un fin determinado, de forma flexible, aprovechando principios de auto-organización, es decir, sin una gestión centralizada de la red.

Las mediciones obtenidas y transmitidas por los nodos sensores dentro de paquetes se denominan *datos*. Si los nodos intermedios van realizando algún tipo de procesamiento de los datos mientras los paquetes son retransmitidos, que es una de las características de las WSNs, se dice que se está transmitiendo datos procesados o información. Finalmente, el *conocimiento* se construye en base a la información obtenida, y nos permite comprender el sistema y aplicar alguna técnica de aprendizaje. A lo largo de este trabajo se hará referencia indistinta a que el nodo sink puede obtener tanto datos como información por parte de los nodos de la red. Sin embargo, los mecanismos subyacentes son muy diferentes, y esto depende del funcionamiento de la red. Es decir, si los datos transmitidos por los nodos no sufren ninguna modificación hasta que son recibidos en el nodo sink, estamos hablando de *reporte de datos*. Sin embargo, si los mismos son procesados en-ruta por los nodos intermedios, hablamos de *reporte de información*.

Las WSNs han despertado un gran interés en los últimos años debido a su gran versatilidad y a su amplio campo de aplicación, principalmente en el monitoreo de magnitudes físicas en lugares remotos e inhóspitos, o con poca accesibilidad. Las aplicaciones de detección de eventos y de seguimiento de objetos móviles por parte de una WSN están incluidas dentro del concepto de monitoreo. Algunos ejemplos de aplicaciones de las WSNs son, el monitoreo de recursos naturales (ríos, volcanes, incendios, etc), de cultivos, de animales, de pacientes y deportistas, de estructuras (puentes, edificios), el seguimiento de objetos móviles, la domótica en hogares (control de calefacción, refrigeración, iluminación, seguridad, etc.), el control de tráfico vehicular, la automatización industrial, etc. Además, este tipo de tecnología permite eliminar los elevados costos de cableado de dispositivos en red, como por ejemplo los que se encontrarán en los futuros vehículos. Sin embargo, aún no se ha cumplido con el objetivo comercial de esta tecnología, que considera que estos dispositivos sean desarrollados masivamente y comercializados con un costo muy bajo, menor a un dolar.

Afortunadamente, existen actualmente numerosas actividades de *investigación y desarrollo* en el área de las WSNs, las cuales están enfocadas en la optimización de esta tecnología para aplicaciones de la vida real. Estas actividades son llevadas a cabo no sólo por la *comunidad científica* (Universidad de California, Universidad de Berkeley, etc.), sino también por *industrias tecnológicas* (Intel, Motorola, Crossbow, Atmel, Texas Instruments, Jennic, Sun, Libelium, etc). Muchas de estas actividades contribuyeron al desarrollo de los estándares y especificaciones de comunicaciones utilizados actualmente por las WSNs, principalmente el estándar IEEE 802.15.4. Este estándar define las capas física y de enlace de dispositivos de comunicaciones de baja velocidad y bajo consumo, para redes de área personal (Personal Area Network, PAN) [17]. Sin embargo, la *adopción comercial* de esta tecnología es aún relativamente baja. Esto se debe tanto a que es una tecnología relativamente reciente y que aún necesita madurar en ciertos aspectos, como por ejemplo, el hecho de que *los estándares y especificaciones actuales dejan varios aspectos de diseño en manos de la implementación misma. Son estos espacios los que posibilitan realizar actividades de investigación y desarrollo con aportes innovadores que permiten generar un mayor valor agregado a esta tecnología.* Es por ello, que existe actualmente una gran comunidad de investigadores y desarrolladores trabajando en esta área, siendo una de las más activas dentro de los sistemas de comunicaciones y de redes de datos.

1.1. Problema de Investigación

Para que la tecnología de las WSNs se vea como una solución atractiva para la sociedad, es necesario desarrollar soluciones innovadoras y eficientes que permitan hacer frente a las limitaciones actuales de esta tecnología. Entre estas limitaciones se destacan *i) el acotado tiempo de vida de los nodos sensores debido a su limitada energía, lo que afecta a la vida de la red, ii) el desbalance en el uso de energía por parte de los nodos de la red, iii) el elevado costo de comunicación requerido por cada nodo para actualizar su conocimiento local de la red, para hacer frente a la dinámica de la misma, y iv) el aún elevado costo comercial de esta tecnología.* Como conclusión, se puede observar que la mayor parte de estos inconvenientes están relacionados con la energía, los cuales limitan el tamaño máximo de las redes que pueden ser implementadas, es decir, que afectan a su escalabilidad. Además, los tres primeros puntos están interrelacionados. A continuación se describen en detalle estas limitaciones.

En la actualidad, el principal objetivo en la investigación y desarrollo de WSNs se centra en *como hacer un uso mas eficiente y posiblemente más equitativo de los limitados recursos energéticos con los que cuentan los nodos y la red en general.* Es decir, se busca mediante diferentes estrategias, extender la autonomía, tanto funcional como energética, de los nodos, y que los datos de la magnitud sensada sean obtenidos y enviados al nodo sink de forma precisa. Sin embargo, dado que los nodos son alimentados con baterías, existe un fuerte compromiso entre ambas variables (autonomía y precisión). Ello se debe a que en el reporte de datos se requiere de un gran uso de comunicación, y en consecuencia, de energía. Lo que se busca es una doble optimización, *maximizar la precisión en el sensado y reporte de los datos, minimizando el uso de energía de los nodos.* Sin embargo, debido a que los datos obtenidos por los nodos sensores se envían a un único nodo, a través del cual accede un usuario a la red, denominado nodo sumidero, o sink, se produce un inevitable *desbalance en el uso de la energía* entre nodos cercanos y distantes al nodo sink. Ello se debe a que, al tratarse de una red con un flujo de comunicación de datos de muchos-a-uno, y a través de múltiples saltos, inevitablemente aquellos nodos mas cercanos al sink tienen una mayor tarea de retransmisión de datos que la que tienen los nodos mas distantes. Esto

repercute en un mayor consumo de energía por parte de los nodos mas próximos al sink, produciendo un desbalance tanto en el uso como en la energía remanente de los nodos que conforman la red. Como efecto no deseado, se produce la formación de huecos de comunicación en la red que terminan aislando o desconectando al nodo sink de la misma.

Por otro lado, en la mayoría de los protocolos propuestos en la literatura para la obtención de información en las WSNs, se solicita inicialmente los datos a todos los nodos de la red, o se los programa para cumplir con cierta tarea, y posteriormente se produce el reporte o flujo de datos en una dirección, es decir, desde los nodos sensores hacia el nodo sink. Sin embargo, este reporte de datos puede ser iniciado basado en intervalos de tiempo (*reporte de datos continuo*), o ante la aparición de eventos (*reporte de datos basado en eventos*). Sin embargo, bajo estos esquemas, y para hacer frente a la elevada variabilidad existente en estas redes, debido por ejemplo a fallas en nodos, paquetes perdidos, movilidad de nodos, adición de nodos, etc, es necesario que cada nodo lleve adelante, cada un cierto tiempo, una *tarea de reconocimiento de su entorno*. Esta tarea se basa en la diseminación de paquetes de exploración por parte de cada nodo, y permite que cada uno de ellos obtenga información actualizada del estado de la red, es decir, de nodos vecinos, distancia al sink y a través de que nodos vecinos se puede llegar al sink, etc. Sin embargo, este esquema es muy eficiente energéticamente, principalmente en redes de grandes dimensiones o muy densas, dado la elevada demanda de comunicación en la que se incurre.

Por otro lado, el *reporte de datos basado en un esquema de pregunta/respuesta o de requerimiento*, iniciado por un usuario para obtener los datos sensados por la red, y no por tiempo ni por la detección de un evento, presenta muchas ventajas respecto a los modelos anteriores, principalmente frente a la dinámica de estas redes. Esto se debe a que, en el proceso de consulta o de diseminación de requerimientos a los nodos de la red, se muestrea implícitamente el estado de la red. Es decir, que solo cuando un usuario requiere información de la red, se actualiza el conocimiento del estado de la misma por parte de cada nodo, lo cual es más eficiente que mantener una actualización permantente y repetitiva. En este sentido, los protocolos propuestos en esta Tesis se basan en este tipo de modelo basado en requerimientos. pero a diferencia de la mayoría de las propuestas existentes, estos utilizan *comunicación por broadcast*, los cuales serán abordados en los próximos capítulos.

Sin embargo, vale la pena aclarar que no existe un esquema de reporte de datos más adecuado o eficiente que los demás, sino que esto depende exclusivamente de la aplicación particular en la que se vaya a utilizar a la WSN. Es decir, para analizar la evolución temporal de la magnitud sensada por la red, es más adecuado el reporte de datos continuo. Sin embargo, para aplicaciones de detección de eventos esporádicos, o de seguimiento de objetos, en la que solo algunos nodos participan de la tarea, es más eficiente utilizar el reporte de datos basado en eventos. Finalmente, aplicaciones en las que se realiza computación distribuida de los datos sensados por los nodos cuando un usuario lo requiere, o en casos en los que solo algunos nodos pueden proporcionar argumentos relevantes a la aplicación, es más adecuado utilizar un esquema basado en requerimientos.

Lamentablemente, el *costo actual de los nodos para implementar WSNs es aún significativo*, debido fundamentalmente a su baja escala de producción. En este sentido, una mayor aceptación comercial de este tipo de redes, impactaría en el desarrollo de esta tecnología a mayor escala, posibilitando una reducción significativa de los costos de producción, que es uno de los objetivos aún no alcanzados por esta tecnología.

Como consecuencia de las limitaciones citadas anteriormente, hasta la fecha solo se han realizado implementaciones reales de WSNs no mucho mayores a 100 nodos. Sin embargo, mediante el uso de las potentes herramientas de simulación orientadas a eventos existentes actualmente, se pueden realizar simulaciones de redes de hasta dos órdenes de magnitud mayor, las cuales permiten evaluar la performance de los protocolos, esquemas y algoritmos propuestos en la literatura. En resumen, el problema de la limitada escalabilidad asociado al reducido tiempo de vida de este tipo de redes es un tema recurrente, y sigue siendo uno de los principales temas de investigación en el área de las WSNs.

En este sentido, y hasta la actualidad, se han propuesto diferentes *esquemas de conservación de la energía en WSNs* (tratados luego en la Sección 1.1.6), para hacer frente a las limitaciones anteriormente expuestas, entre las cuales *la reducción de datos es la principal estrategia desde el punto de vista del procesamiento de datos en red* [3]. En este sentido, en los últimos años se han desarrollado una gran cantidad de *protocolos de procesamiento de datos basados en agregación o fusión aplicados al proceso de reporte o recolección de los mismos, con el objetivo de reducir el volumen de datos, y por lo tanto la comunicación, extendiendo así el tiempo de vida de la red*. Estos protocolos implementan agregación o fusión de datos en nodos intermedios mientras los datos son retransmitidos hacia el nodo sink. Al igual que en el caso del reporte de datos, la técnica de agregación en red más apropiada depende de la aplicación específica. Sin embargo estos protocolos son ineficientes energéticamente en aplicaciones en las que pocos nodos proveen información útil al usuario, como se demostrará más adelante.

La pregunta que busca responder esta Tesis es la siguiente:

¿ Como se puede minimizar el costo de comunicación, y prolongar el tiempo de vida de redes inalámbricas de sensores, con un bajo error de computación, mediante técnicas de enrutamiento, procesamiento y filtrado distribuido centrados en datos, en aplicaciones en las que solo algunos nodos pueden aportar información útil al usuario ?

A continuación, se describe en detalle que es una WSN, sus elementos constitutivos, como son utilizadas para procesar datos, su origen y evolución, sus aplicaciones, las limitaciones actuales, y las técnicas existentes para la reducción del uso de energía.

1.1.1. Redes Inalámbricas de Sensores

Una WSN es un conjunto de dispositivos, o sistemas embebidos, de pequeñas dimensiones y bajo costo, llamados *nodos sensores*, que pueden sensar alguna magnitud de su entorno para obtener datos, almacenarlos, procesarlos, y comunicarlos inalámbricamente a otros nodos (Figura 1.1). La misma puede desplegarse sobre un área de interés para obtener información de magnitudes físicas tales como calor, luz, ruido, vibración, radiación, emisiones químicas, etc. En este sentido, los nodos incluyen sensores, microcontroladores de bajo consumo, memorias, sistemas de alimentación basados en baterías, y transceptores de radio de corto alcance, baja velocidad, y bajo consumo [2].

Los elementos principales que conforman una WSN, considerando el modelo de reporte de datos basado en requerimientos, son (Figura 1.2):

- *Nodo sensor*: dispositivo que sensa o muestrea la magnitud monitoreada en su entorno, almacena estos datos en memoria, realiza algún tipo de procesamiento local, y transmite tanto datos

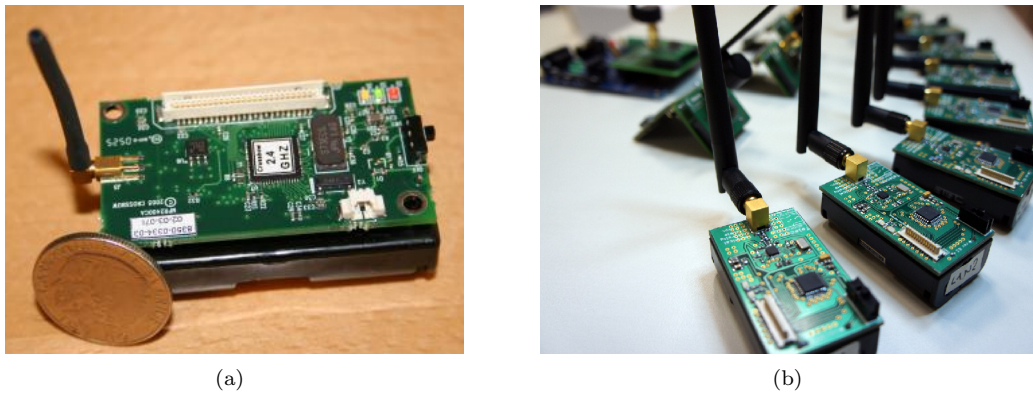


Figura 1.1: Nodo sensor (a) utilizado para conformar redes inalámbricas de sensores (b).

propios como aquellos provenientes de otros nodos.

- *Nodo base*: conocido como nodo sink, es la interfaz entre la red de nodos sensores y el usuario, y a través del cual el usuario puede introducir consultas o solicitudes, y extraer datos o información de la red. Sin embargo, por lo general, cualquier nodo sensor que sea consultado por un usuario puede tomar el rol de nodo coordinador.
- *Observador*: usuario interesado en obtener los datos o la información recolectada por la red de sensores acerca de la magnitud física. El observador, que puede ser local o remoto, puede realizar consultas a la red y recibir respuestas sobre las mismas.
- *Fenómeno físico*: entidad de interés del observador, que está siendo sensada y potencialmente procesada por la red de sensores (por ejemplo, temperatura, humedad, luz, sonido, etc). Múltiples fenómenos pueden estar bajo observación concurrentemente en la red.
- *Requerimiento*: consulta, solicitud, o interés del usuario por cierta información del fenómeno físico que está siendo monitoreado, inyectado a la red como un mensaje de consulta. Por ejemplo, cual es el valor máximo detectado por los nodos, cuantos nodos detectaron un evento determinado, etc.

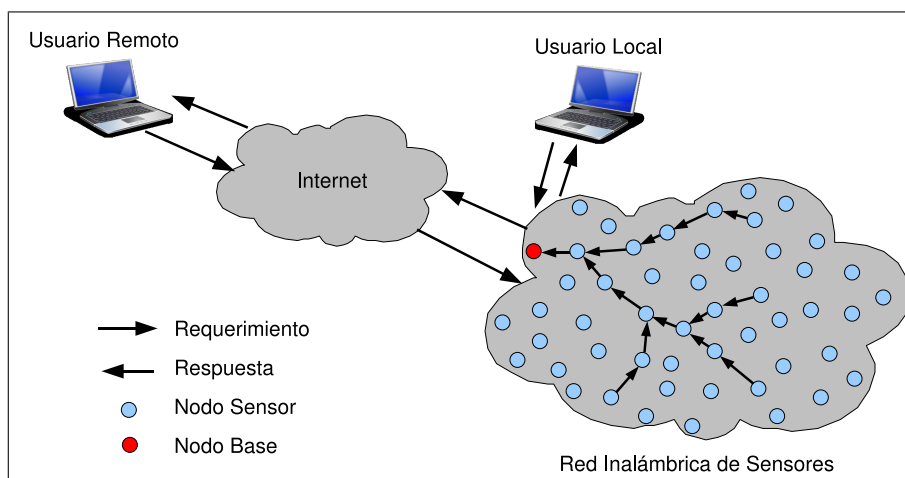


Figura 1.2: Elementos que conforman una red inalámbrica de sensores.

Lo que se busca, no es tener un sistema monolítico con elevadas prestaciones de sensado, memoria, procesamiento, y comunicación, sino más bien, un sistema altamente distribuido con prestaciones limitadas. En las WSNs no se requieren grandes capacidades de procesamiento y de comunicación en cada nodo, sino por el contrario, se busca dividir estas tareas entre los nodos, con el fin de minimizar el uso de la energía y extender el tiempo de operación de la red. Se trata de dividir la gran capacidad que podría tener una estación, en una gran cantidad de nodos con capacidades reducidas, como un sistema totalmente distribuido, pero este debe cumplir con requerimientos de eficiencia energética. Por esta razón, la eficiencia energética es un factor clave en el diseño de sistemas basados en esta tecnología. Además, el uso de sistemas de sensado distribuidos permite obtener mayores puntos de muestreo de la magnitud sensada.

En ciertos aspectos, las WSNs tienen ciertas similitudes con las redes ad-hoc móviles (mobile ad-hoc networks, MANETs), por ejemplo en el sentido en que en ambos tipos de redes se utilizan comunicaciones por múltiples saltos (multi-hops). Sin embargo, la naturaleza de las aplicaciones y los requerimientos de enrutamiento de ambas redes son muy diferentes en varios aspectos: i) el modo típico de comunicación en una WSN es de múltiples fuentes de datos a un único destino (sink), como una comunicación multicast inversa, en lugar de una comunicación entre un par de nodos, ii) debido a que los datos recolectados por múltiples sensores están basados en un fenómeno común, puede existir cierta redundancia en los datos que son comunicados en la red, iii) en la mayoría de los escenarios los nodos sensores no son móviles (a pesar de que el fenómeno sensado si puede serlo), por lo que la naturaleza de la dinámica de ambas redes son diferentes, y iv) la principal limitación en las WSNs es la energía.

Una vez que una WSN es desplegada en un área, esta debe funcionar de forma *autónoma* y *desatendida*, es decir, sin control humano. Por ejemplo, a medida que algunos nodos se van quedando sin energía, se producen desconexiones y la formación de huecos en la red. Para remediar esta situación pueden incorporarse nuevos nodos a la red, teniendo que adaptarse el funcionamiento de la misma a dichos cambios.

1.1.2. Procesamiento de Datos en WSNs

Para analizar el procesamiento de datos en WSNs se utilizan, por lo general, dos tipos de redes de comunicación, *las redes yuxtapuestas o con enlaces directos*, y *las redes con múltiples saltos*. Las *redes yuxtapuestas (collocated networks)*, son redes en las que la transmisión de paquetes de datos por parte de un nodo puede ser recibida por todos los demás nodos que conforman la red, es decir, la comunicación se realiza a través de un enlace directo, o de un salto de comunicación, como se indica en la Figura 1.3a. Este esquema tiene la ventaja de que los datos pueden ser recolectados de forma precisa en el nodo sink. Sin embargo, tiene como desventajas principales que i) el costo energético de la comunicación es elevado, debido a los grandes rangos de comunicación, ii) cada nodo recibe el paquete transmitido por cada uno de los demás nodos, y iii) se incrementa en gran medida la probabilidad de colisiones de paquetes debido al acceso masivo a un canal de comunicación común. Como el principal objetivo en el desarrollo de una WSN es minimizar el uso de la energía, y debido a que el costo de comunicación se incrementa exponencialmente con el rango de comunicación, se busca minimizar este último parámetro. Debido a esto, el esquema de comunicación utilizado en las WSNs es el de *transmisión de datos por múltiples saltos o múltiples retransmisiones* (Figura 1.3b). Además,

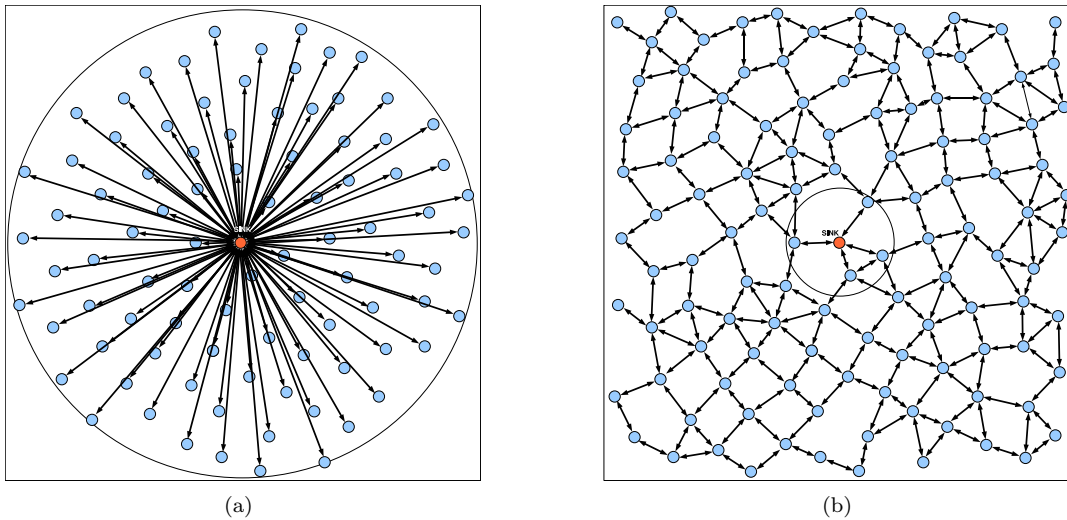


Figura 1.3: Redes de comunicaciones con enlaces directos (a), y con enlaces multisalto (b). Los círculos azules representan nodos sensores, y el círculo rojo representa al nodo sink. Las flechas indican los enlaces inalámbricos.

en las WSNs, se accede a los nodos de la red y se extrae información de los mismos, por lo general, a través de un único nodo sink, como se indica en la Figura 1.3.

Desde el punto de vista del procesamiento de los datos, este puede ser realizado *i) de forma centralizada*, tanto en un servidor de datos externo a la red, como en el nodo coordinador, *ii) de forma distribuida*, entre los nodos que conforman la red, o *iii) una combinación de los anteriores*, siendo este último enfoque el abordado en esta Tesis por considerarlo el más conveniente. Tanto el procesamiento centralizado como el distribuido requieren, por lo general, de un elevado costo de comunicación cuando las redes son de grandes dimensiones. Esto se debe al envío masivo de datos o información desde muchos nodos a un único nodo, en el caso centralizado, y al intercambio de datos entre los nodos en el caso distribuido. Sin embargo, la capacidad de los nodos sensores de poder realizar procesamiento de datos, como por ejemplo, la computación de funciones, es limitada. Ello se debe a que los nodos incluyen microcontroladores de baja capacidad y consumo, ya que existe un compromiso entre potencia de cómputo y consumo de energía. El desafío actual en esta tecnología es poder desarrollar estrategias de procesamiento distribuidas entre los nodos de la red que permitan reducir el flujo de datos reportados al usuario, lo que produce una reducción del costo de comunicación en la red, y como consecuencia, un incremento en el tiempo de vida de la red.

La motivación de esta Tesis, es considerar a las redes de sensores como sistemas que pueden ser estimulados mediante el requerimiento de información por parte de un usuario, y la red procesar los datos y responder a dicho estímulo entregando la información solicitada. Por ejemplo, desde el punto de vista de la computación de funciones, se podría llegar a considerar a una WSN como una librería matemática, a la que el usuario puede solicitar el cómputo de alguna función estadística de los datos obtenidos por los nodos, y la red responder con los resultados obtenidos (Figura 1.4). Sin embargo, como se verá en los próximos capítulos, *el protocolo más adecuado y eficiente es totalmente dependiente del tipo de función que se quiera computar, y no existe una solución general para este problema.*

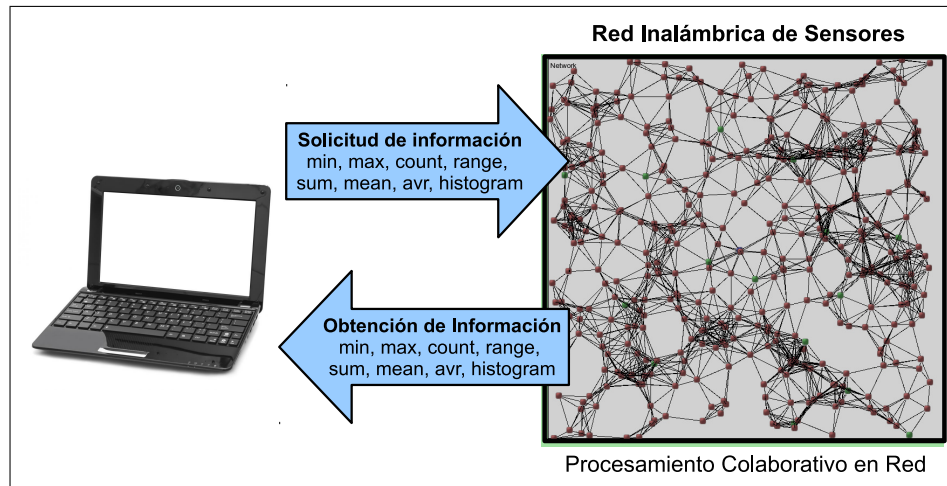


Figura 1.4: Esquema de uso considerado para la red de sensores basado en pregunta y respuesta.

1.1.3. Origen y Evolución de las WSNs

El origen de la investigación en redes de sensores se remonta al programa *Distributed Sensor Network (DSN)* de la Agencia de Proyectos de Investigación Avanzada de Defensa (DARPA), alrededor del año 1980. Por aquellos años estaba funcional la ARPANET (Advanced Research Projects Agency Network) desde hacía varios años, con cerca de 200 hosts en Universidades e Institutos de Investigación. En este programa de investigación, se consideraba que las DSNs estaban conformadas por muchos nodos sensores distribuidos espacialmente, que colaboraban entre ellos, pero que operaban en forma autónoma, enrutando información a cualquier otro nodo que la necesitara. En aquel tiempo, este fué un programa muy ambicioso, debido a que no existían las computadoras personales, y recién se estaba haciendo popular Ethernet. Un workshop realizado en 1978 (Proceedings of the Distributed Sensor Nets Workshop, 1978) fué el que permitió identificar los componentes tecnológicos necesarios para implementar una DSN. Estos incluían sensores acústicos, módulos de comunicación y de procesamiento, y software distribuido. En 1981 investigadores de la Universidad de Carnegie Mellon desarrollaron el primer sistema operativo para DSNs, denominado Accent [5]. Aplicaciones demostrativas de DSNs fueron la de un sistema de seguimiento de un helicóptero, desarrollado por el Massachusetts Institute of Technology (MIT) en 1984, y la de seguimiento de objetos basado en sistemas de decisión y sensores acústicos en 1988. Lamentablemente, estos primeros desarrollos no prosperaron debido a la inmadurez de las tecnologías utilizadas, y a los elevados costos de desarrollo.

Sin embargo, el origen de las redes inalámbricas de sensores, como las conocemos actualmente, no tiene un hito definido, sino que es el resultado de la confluencia de diferentes tecnologías (electrónica, informática, comunicaciones, energía) que han evolucionado en gran medida en los últimos 15 años. Desde el punto de vista de los sistemas de computación y de comunicación, los primeros hechos que influyeron para el desarrollo de esta tecnología han sido: el desarrollo y estandarización de redes ad hoc [6], el concepto de redes activas [7], y el desarrollo de estrategias basadas en el paradigma de procesamiento y enrutamiento centrado en datos [8], todos producidos a fines de la década de los 90, acuñándose el término WSN alrededor del año 2000. Las *redes ad hoc* son redes que se construyen en forma dinámica, carecen de una infraestructura fija, y tienen capacidad de adaptarse a los cambios. Por otro lado, las *redes activas* son redes en las que los nodos intermedios entre nodos origen y destino,

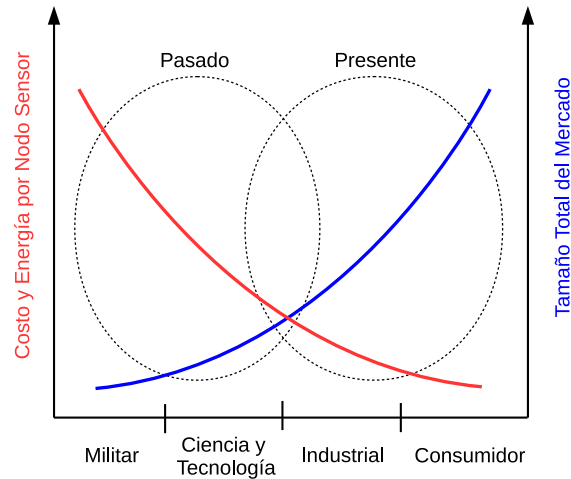


Figura 1.5: Evolución de las redes inalámbricas de sensores.

además de retransmitir los paquetes, pueden ir procesando los datos o la información contenida en los mismos. Respecto al software que se ejecuta en los nodos, los sistemas operativos para sistemas embebidos que comenzaron a utilizarse en WSNs, surgieron alrededor del año 2000, en el caso de TinyOS, y en 2004 en el caso de ContikiOS. Desde el punto de vista de los protocolos de comunicación, en el año 2003 fué publicado el estándar IEEE 802.15.4-2003, que define las capas física y de enlace para redes de área personal de baja velocidad (low-rate personal area networks, LR-PANs), conformadas por dispositivos de bajos recursos, cuya actualizaciones fueron publicadas en los años 2006 y 2011. Este es el estándar más utilizado en las WSNs actuales.

Como se puede observar, los mayores avances en WSNs se han dado en los últimos 15 años. Sin embargo, actualmente, estas redes evolucionan a pasos agigantados, siendo una de las áreas de investigación y desarrollo más activas en el ámbito de los sistemas de comunicaciones. Se puede observar en la Figura 1.5 la evolución de las WSNs respecto a la reducción de los costos y de la energía consumida por cada nodo sensor, y al incremento en el mercado de consumidores.

1.1.4. Aplicaciones

En la actualidad, existe una gran cantidad de aplicaciones en las que se utilizan WSNs (Figura 1.6), pero en general, permanecen muchas aplicaciones innovadoras aun sin explorar ni explotar. En general, las principales aplicaciones de las WSNs pueden ser clasificadas en: *detección de eventos*, *seguimiento de objetos móviles (tracking)*, y *monitoreo continuo de magnitudes físicas en un área determinada*.

En las WSNs no existe una única forma de funcionamiento, ni protocolos, mecanismos, esquemas, o algoritmos que sean óptimos para todas las aplicaciones posibles de las WSNs, sino que esto es totalmente dependiente de la aplicación particular que se considere. En este sentido, existen diferentes esquemas de adquisición de datos en una WSN, por ejemplo, i) la red podría ser configurada para proveer al nodo sink los datos obtenidos por todos los nodos cada un cierto tiempo dado (time-driven o continuous), ii) podrían generar datos hacia el sink aquellos nodos que solamente detecten un evento particular en la red (event-driven), o iii) que los nodos sensores envíen datos o información procesada basado en una consulta o requerimiento de información (query-based).

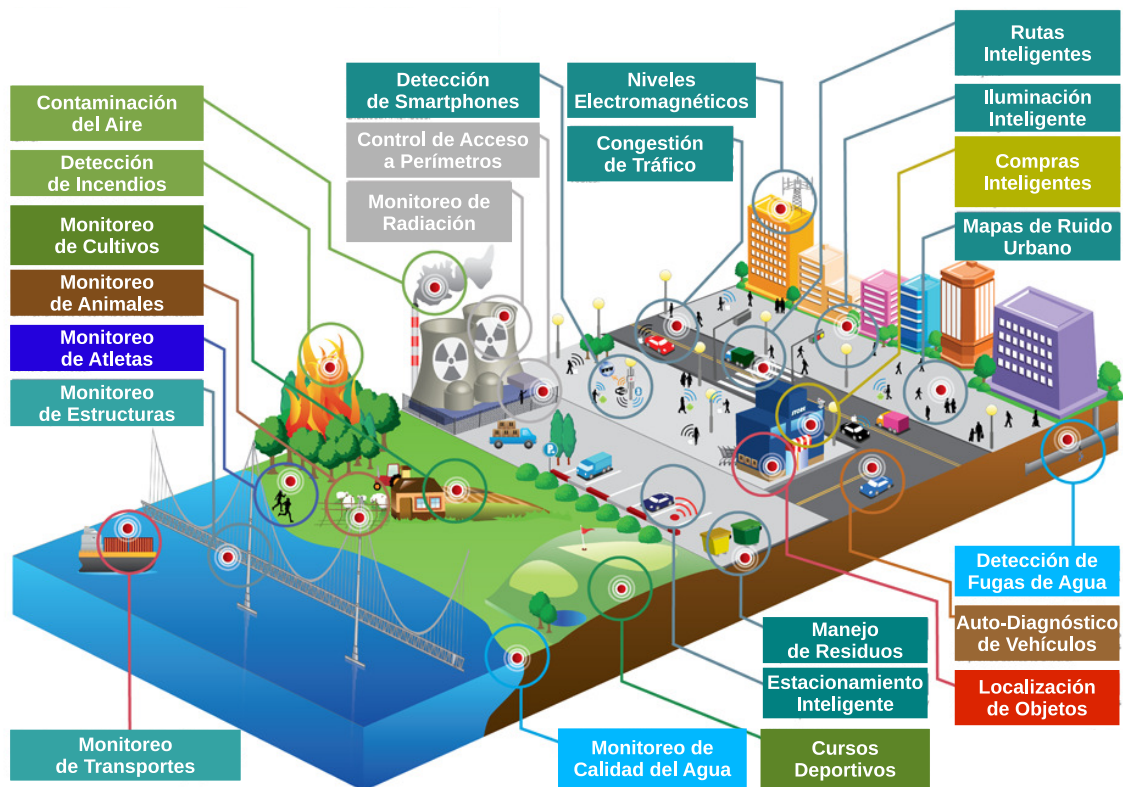


Figura 1.6: Aplicaciones de redes inalámbricas de sensores.

Sin embargo, en los últimos años se han propuesto aplicaciones que consideran a un WSN como una gran base de datos distribuida, a la cual un usuario puede requerir cierto tipo de información [9], como resultado del procesamiento de los datos dentro de la red. Por ejemplo, si en los postes de iluminación de una ciudad son colocados nodos con micrófonos, o con sensores de monóxido de carbono, un usuario puede acceder a la red en cualquier momento, y solicitar a la misma, por ejemplo, que se reporten solo los valores máximos de contaminación sonora, o de polución detectados en la red, y no todos los datos sensados. Este tipo de aplicación no está incluida ni en la detección de eventos, ni en el seguimiento de objetos, ni en el monitoreo continuo de datos.

Actualmente, existen muchos tipos de redes que pueden ser enmarcados dentro de las WSNs. Estos son tipos de WSNs definidos para aplicaciones particulares, como por ejemplo, las *redes inalámbricas de sensores multimedia (wireless sensor multimedia networks, WSMNs)*, las *redes de sensores (under-water sensor networks, USNs)*, las *redes de área corporal (body area networks, BANs)*, las *redes de área vehicular (vehicular area sensor networks, VANs)*, etc. Actualmente, este tipo de redes también se comienza a fusionar con el concepto de redes bajo el paradigma de *Internet de las Cosas (Internet of Things, IoT)*, conocidos también como sistemas pervasivos y obicuos, o como ambientes inteligentes. Sin embargo, cada una de estos tipos de redes tienen sus particularidades, siendo el concepto de WSNs más general.

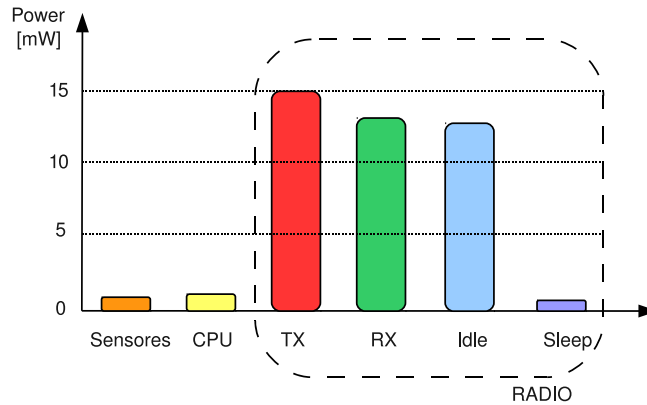


Figura 1.7: Distribución del consumo de energía en un nodo sensor.

1.1.5. Limitaciones

Más allá de los beneficios actuales y potenciales que presentan las WSNs, los nodos tienen, en general, capacidades muy limitadas, y deben operar en forma autónoma y sin mantenimiento, maximizando su tiempo de vida. Estas capacidades limitadas abarcan desde el *sensado, almacenamiento, procesamiento, comunicación y energía*. Ello se debe a que los nodos son dispositivos de bajo costo, y bajo consumo, alimentados energéticamente con baterías. Si bien existe la posibilidad de que los nodos incorporen alguna forma autónoma de carga de energía, basado por ejemplo en energía fotovoltaica, en general, la energía de un nodo es acotada y un factor limitante de las prestaciones del mismo.

En este sentido, el uso del transceptor de radio, ya sea, en modo *pasivo o Idle*, o durante la *transmisión (TX) o recepción (RX) de datos*, es el principal factor de consumo de energía de un nodo en este tipo de redes. Los procesos de sensado y procesamiento de datos (CPU) son, en general, de menor demanda energética, como se muestra en la Figura 1.7, y no son considerados en la estimación del tiempo de vida de la red en la mayoría de los casos. Por lo tanto, *el uso de la comunicación inalámbrica es el principal factor limitante de la vida de cada nodo y de la red en general, y su reducción o minimización es el principal desafío en el que se centran los trabajos de investigación y desarrollo en WSNs, y en esta Tesis en particular.*

Se ha demostrado mediante ensayos y mediciones reales, que en un nodo se tiene el mismo consumo energético tanto al ejecutar 3000 instrucciones de código con un microcontrolador de bajo consumo como al transmitir un bit a 100 metros de distancia con un transceptor de radio de bajo consumo. O de forma equivalente, transmitir 1 kB de datos es equivalente a computar 3 millones de instrucciones de código [10]. Sin embargo, el consumo utilizado en la comunicación de datos afecta tanto al emisor como al receptor, casi en la misma magnitud. En resumen, se estima que la relación entre comunicar y procesar está en el rango de 1000 a 10000 veces, según el tipo de radio y microcontrolador utilizados. Debido a esto, el definir un adecuado y eficiente manejo de los recursos limitados, como por ejemplo los energéticos, es un factor clave y un desafío en el diseño y desarrollo de WSNs [3].

Este desafío nos motiva a desarrollar protocolos, esquemas, mecanismos, y algoritmos que mediante el uso de procesamiento de datos distribuido entre los nodos de la red, se pueda reducir la comunicación de datos, y por lo tanto, se maximice el tiempo de vida de la red.

Una de las métricas más importante en WSNs, utilizada para evaluar y comparar la performance de los diferentes protocolos propuestos en la literatura, es el *tiempo de vida de la red (network lifetime)*. Esta métrica puede ser definida de varias maneras. La más comúnmente usada es la que define a la misma como: i) la duración en tiempo hasta que el primer nodo de la red se queda sin energía. Sin embargo, pueden definirse variantes de esta, como por ejemplo, ii) la duración en tiempo hasta que un porcentaje de nodos se quede sin energía, entre otras. Desde otro punto de vista, se puede definir también como iii) la duración en tiempo durante el cual un porcentaje específico de nodos están funcionales y conectados, cumpliendo la red con requerimientos de calidad de servicio. Un buen protocolo de comunicación para WSNs es aquel que logra su objetivo maximizando el tiempo de vida de la red, es decir, que permita mantener funcional a la red la mayor cantidad de tiempo posible, sin que se produzca agotamiento energético prematuro.

A continuación se describen los tipos de técnicas mas comunmente encontrados en la literatura para minimizar el uso de energía en los nodos que conforman una WSN.

1.1.6. Técnicas de Reducción de Energía

Como se describió anteriormente, es necesario aplicar alguna técnica para reducir el uso de la energía en las WSNs, con el objetivo de maximizar el tiempo de vida de la red, principalmente cuando las mismas son de gran tamaño. Si bien en la literatura se han propuesto una gran variedad de técnicas desarrolladas para cumplir con este objetivo, las principales se basan en: *i) el ciclo de trabajo, ii) la movilidad de los nodos, y iii) el manejo de los datos* [3]. A continuación se describen cada una de ellas en detalle.

- La *primer técnica* se basa en estrategias de encendido y apagado del transceptor de radio. En este sentido, el transceptor es puesto en modo de bajo consumo (Sleep) el mayor tiempo posible cuando no se requiere comunicación. Idealmente, el transceptor de radio puede ser apagado tan rápido como sea posible una vez que no haya mas paquetes de datos que transmitir/recibir, y puede ser reanudado tan rápido como sea posible cuando tenga un nuevo paquete para transmitir. De este modo, los nodos pueden alternar entre períodos activos (Tx, Rx, Idle) y de sleep, dependiendo de la actividad de la red. Esta técnica es conocida como *duty-cycling* [4], definiendo al ciclo de trabajo como la fracción de tiempo en que los nodos están activos durante su tiempo de vida. Sin embargo al utilizar este tipo de técnica en una WSN, en la que se realiza una tarea cooperativa de sensado y reporte de datos, se necesita coordinar los tiempos para dormir (sleep) y despertar (wakeup) a los nodos, y por lo tanto, se requiere de una muy eficiente sincronización entre los nodos, lo cual comienza a ser problemático cuando las redes están conformadas por una gran cantidad de nodos. Para ello se deben utilizar algoritmos de planificación de sleep/wakeup y protocolos de sincronización distribuidos en la red, cuyo tratamiento escapa a esta Tesis.
- La *segunda técnica* es utilizada en aquellas WSNs en las que algunos o todos los nodos tienen movilidad. En redes estáticas, los paquetes arriban por múltiples caminos con múltiples saltos desde los nodos sensores hasta el nodo sink. Sin embargo, algunos caminos pueden ser mas utilizados o tener un mayor tráfico que otros, produciendo un desbalance en el uso de la energía. Esto hace que algunos nodos (que se encuentran sobre estos caminos) estén expuestos a un agotamiento prematuro de energía, cuyo fenómeno es conocido como "efecto de canalización

o funneling”. Por otro lado, aquellos nodos que se encuentran más próximos al sink tienen una mayor demanda de comunicación, debido a que tienen que retransmitir más cantidad de paquetes. Si algunos nodos (incluido el sink) son móviles, el flujo de tráfico puede ser alterado si los dispositivos móviles son responsables de la recolección de datos de los nodos estáticos. Estos últimos pueden esperar el paso de dispositivos móviles y enrutar mensajes a través de ellos. Como consecuencia, los nodos estáticos pueden reducir el uso de la energía mediante la reducción de la longitud del camino de enrutamiento, la contención y el overhead de retransmisión. Además, el dispositivo móvil puede visitar la red para lograr mayor uniformidad en el consumo de energía de los nodos debido a la comunicación. Adicionalmente, los nodos sensores o el nodo sink pueden desplazarse hacia el lugar adonde hay mayor información respecto a la magnitud sensada para reducir el costo de reporte de datos. En el caso de WSNs con baja densidad de nodos, se puede utilizar un nodo móvil para la recolección de los datos, basado en el paradigma de redes tolerantes a interrupción (*delay/disruption-tolerant networks*, DTNs). En aquellos casos en que el costo energético y económico de movilizar nodos sensores es muy elevado, una posible solución es colocar los nodos sensores a objetos móviles que se estén desplazando en el área sensada, tales como vehículos, personas, animales, etc.

- La *tercer técnica* de conservación de energía es útil *cuando la cantidad de datos o de información a extraer de la red es relevante y posiblemente con una alta correlación espacial y/o temporal, o cuando el consumo del subsistema de sensado no es despreciable*. Esta técnica puede ser subdividida en *técnicas de reducción de datos* y *técnicas de adquisición de datos en forma energéticamente eficiente*. Las primeras contemplan el uso de procesamiento en red (estrategia en la que se enfoca esta Tesis), de compresión de datos, y de predicción de datos, conocidas como *in-network processing*, y son adecuadas cuando se tratan problemas en los que se tienen, ya sea, o una elevada correlación entre los datos, o bien pocas muestras de datos relevantes para la aplicación. La segunda está más enfocada en estrategias de muestreo de datos en los nodos, más adecuadas cuando el uso de energía del subsistema de sensado no es despreciable. Afortunadamente, ambas técnicas ayudan a reducir la cantidad de datos a ser entregados al nodo sink. Sin embargo, los principios detrás de estas técnicas son muy distintos. A lo largo de esta Tesis nos enfocaremos en estrategias de procesamiento de datos en red. Estas últimas técnicas permiten, mediante mecanismos simples y colaborativos de procesamiento y filtrado, reducir la cantidad de datos comunicados en la red, con su consecuente reducción de energía.

En los últimos años, la comunidad científica en el área de WSNs viene trabajando activamente en el desarrollo de protocolos para procesamiento de datos en la red (*in-network processing*) cada vez más eficientes, los cuales permiten reducir el volumen o flujo de datos, y por lo tanto, el elevado costo de comunicación presente en estas redes. Estos protocolos permiten disminuir el costo de comunicación sacrificando costo de computación, basándose en el hecho de que *la computación de datos consume significativamente menor cantidad de energía que la comunicación*. Si bien, muchas soluciones han sido presentadas, como por ejemplo, basadas en agregación, fusión, etc, la técnica de procesamiento más apropiada depende específicamente de la aplicación en que se utilice a la WSN. Todas estas técnicas se basan en la alta correlación espacial y temporal de los datos. Sin embargo, se tiene como desventaja que estas técnicas introducen una cierta demora en obtener los datos de la red, debido al procesamiento de los datos. Estas técnicas serán tratadas en la Sección 2.2, en el que se describen sus variantes, ventajas y desventajas.

1.2. Relevancia del Problema

Las redes cableadas de sensores no son una novedad, pero los recientes avances en la fabricación de microchips de transceptores de radio de bajo consumo están logrando eliminar los cables de estas redes, permitiendo crear redes inalámbricas de sensores con mayor valor agregado y potencial que sus pares cableados. No por nada, un informe anual del MIT (Massachusetts Institute of Technology) definió a esta tecnología como una de las 10 que cambiarán el mundo [11], y actualmente el número de empresas que las fabrican en un país está considerado como un indicador tecnológico. Además, la aparición de las WSNs cumple con la llamada Ley de Bell [12]: "*Aparecerá una nueva tecnología cada 10 años, y creará nuevas aplicaciones*". Los pronósticos más optimistas apuntan hacia una eventual proliferación de redes de sensores inteligentes [13], redes que recogerán enormes cantidades de información contribuyendo de forma favorable al buen funcionamiento de fábricas, al cuidado de cultivos, a tareas domésticas, a la organización del trabajo, a la atención y monitoreo de pacientes y de deportistas, y a la predicción de desastres naturales como los terremotos e incendios. En este sentido, la computación que penetra en las facetas de la vida diaria de los seres humanos (conocida como computación ubicua), y que está omnipresente (pervasivo), está a punto de convertirse en realidad [14]. Su gran poder de integración con otras tecnologías promete generar un gran impacto en varias aplicaciones comerciales [15], pero para que esto suceda es necesario que se desarrollen estándares y especificaciones que permitan hacer de estas redes una tecnología madura [16]. En este sentido existen versiones de estándares y especificaciones muy recientes, como lo son el estándar de nivel físico y enlace IEEE 802.15.4 [17], y la especificación Zigbee (2007) a nivel red [18].

En el ámbito nacional, se conocen algunos grupos de investigación y desarrollo (I+D), pertenecientes a Universidades, que están llevando adelante líneas de investigación en el área. Sin embargo no existe aún una fuerte articulación, ni líneas de trabajo comunes que puedan ser coordinadas en conjunto entre las instituciones. Se considera que el desarrollo local de esta tecnología es de gran relevancia, dado que su aplicación abarca muchas de las áreas prioritarias definidas en el *Plan Estratégico Nacional en Ciencia, Tecnología e Innovación* de la Nación Argentina [19].

Es de esperar que el uso de las WSNs comience a difundirse en los próximos años, a medida que la tecnología y su estandarización den signos de madurez. En este contexto, las mejoras en la autonomía de estas redes, las cuales se investigan en esta Tesis, son indispensables al momento de desarrollar aplicaciones. Por otro lado, la mejora de la autonomía en las WSNs está recibiendo gran interés en la comunidad científica. En un principio, este interés estuvo focalizado en el diseño de transceptores de radiofrecuencia de bajo consumo [20], para luego concentrarse en el estudio de protocolos de acceso al medio [21], y de enrutamiento de paquetes [22] que minimizan el consumo energético. En los últimos años, han comenzado a aparecer algunos trabajos donde el problema del consumo energético se plantea en niveles superiores, estudiando como deben gestionarse los recursos energéticos desde la capa de aplicación [23, 24, 25] y la disponibilidad de fuentes renovables de energía [26, 27]. Existen además trabajos recientes en los cuales se estudia el problema del procesamiento local de la información y su agregación, en relación con la autonomía del nodo [28, 29]. Es decir, se ha venido trabajando en la optimización en este tipo de redes desde las capas inferiores, y actualmente se trabaja en las capas superiores del stack de protocolos. Sin embargo, no existen aún ni un consenso ni reglas de ingeniería establecidas de como las aplicaciones deben administrar el consumo energético, lo que constituye el problema abierto para la investigación. Tampoco existe suficiente validación experimental de mecanismos de optimización de la autonomía propuestos en la literatura, por lo que se considerará de gran

importancia evaluar sus limitaciones para su implementación y proponer nuevos mecanismos.

Desde el punto de vista del *procesamiento de los datos y de la información en WSNs*, tema en el que se enfoca esta Tesis, la tendencia en los últimos años ha sido desarrollar protocolos de procesamiento en red, descentralizando esta tarea realizada normalmente en un servidor de datos externo. La mayor parte de los trabajos se centra principalmente en *como recolectar más eficientemente la información obtenida por la red*, utilizando *técnicas de agregación o de fusión de datos*, para reducir el volumen de datos, considerando la correlación de los mismos [30, 31], y en algunos casos, *mecanismos de enrutamiento basado en gradientes de información*. Sin embargo, *no se han considerado los casos en los cuales i) solo algunos nodos poseen datos relevantes para la aplicación, ni ii) en los que la correlación de los datos es baja o nula*.

En ciertas aplicaciones puede utilizarse un enfoque diferente al de agregación o fusión de datos, como por ejemplo, el *basado en búsqueda de información* [32]. Muchos trabajos han sido propuestos siguiendo este enfoque de búsqueda de información con desconocimiento de su distribución, basándose en dos tipos diferentes de aplicaciones, i) búsqueda en toda la red, o ii) búsqueda hasta detectar un evento determinado en la red [33, 34]. Los primeros están más enfocados al procesamiento de funciones estadísticas de los datos obtenidos por los nodos sensores, mientras que los segundos están pensados para detección de eventos y objetivos. En este contexto, las técnicas usadas en cada tipo de búsqueda son muy diferentes. Afortunadamente, algunos trabajos han considerado estratégicamente, el uso de gradientes de información en la red para direccionar la búsqueda, haciendo más eficiente a este proceso [35, 36, 37]. Sin embargo, el caso de búsqueda en toda la red, y la progresiva adaptación a un esquema de búsqueda reducido a zonas que prevean información útil no ha sido muy explorado. Es decir, *se pueden utilizar mecanismos de búsqueda de información para llegar con mensajes de requerimiento o de solicitud de información solo a aquellos nodos que proporcionen datos relevantes o útiles para la aplicación, limitando la participación de los demás nodos de la red*. Esto permite reducir significativamente los costos de comunicación, y maximizar el tiempo de vida de la red.

En este contexto, en los últimos años se ha comenzado a considerar el uso de algoritmos metaheurísticos de optimización para la búsqueda de información en WSNs. En esta Tesis, se han desarrollado protocolos basados tanto en procesos físicos, como en la naturaleza (bio-inspirados). En el primer caso, se ha desarrollado una versión paralelizada o multi-instanciada del algoritmo de recocido simulado, el cual es controlado por el nodo sink. En el segundo caso, se trabaja bajo el paradigma de inteligencia colectiva, en el cual se considera que cada nodo solo posee una inteligencia simple y limitada, y que este solo conoce su entorno inmediato. Bajo este paradigma, se logran excelentes resultados en este tipo de aplicaciones. Ello se debe a que *del trabajo en conjunto de nodos con capacidades limitadas pero con un objetivo común, emerge una inteligencia superior a la sumatoria de las capacidades individuales* [38].

1.3. Hipótesis y Metodología de Trabajo

La hipótesis que plantea esta Tesis es que, en la mayoría de las aplicaciones, el usuario de una WSN no está interesado en obtener los datos obtenidos por todos los nodos de la red, sino más bien, obtener el resultado del cómputo de alguna función estadística de los mismos. Sin embargo, el cómputo de funciones de umbral tiene el beneficio de que requiere de pocos argumentos, lo que permite reducir en

gran medida el costo de comunicación.

En este contexto, las funciones estadísticas más utilizadas en WSNs son: *máximo/mínimo*, *k-valores máximos/mínimos*, *iso-contornos*, y *rangos*. En este contexto, los protocolos de procesamiento en red pueden ser utilizados para enviar solo datos relevantes al nodo sink para fines de procesamiento. Por ejemplo, en la Figura 1.8 se describe un ejemplo de distribución de datos de la magnitud sensada en una WSN, donde los colores claros representan valores máximos, y los colores oscuros valores mínimos. El nodo sink (nodo rojo), ubicado en el centro de la red, es el nodo a través del cual el usuario puede acceder a la red. Si el usuario está interesado en obtener solo el/los valores máximos sensados por la red, para computar funciones de tipo máximo/k-máximos, es ineficiente hacer que todos los nodos reporten sus datos al nodo sink, ni que estos apliquen técnicas de reducción de datos en el proceso de reporte, como por ejemplo agregación. En este sentido, es más eficiente considerar al problema como de búsqueda de información, donde el nodo sink, mediante la colaboración de los nodos de la red, puede durante la diseminación de requerimientos a la red, realizar búsqueda de información, basada en heurísticas de búsqueda que utilizan gradientes de información. Asociado a la búsqueda de información, se utilizan protocolos de filtrado en red para reducir tanto los datos comunicados como la participación de nodos.

Uno de los principales objetivos de los protocolos propuestos en esta Tesis, es dirigir la diseminación de requerimientos de información hacia aquellos nodos que puedan proveer argumentos relevantes para la aplicación. En este sentido, para evitar la pérdida de información y ante el desconocimiento inicial de la distribución de los datos en la red, se realiza la diseminación de requerimientos a todos los nodos de la red, y luego se aplican técnicas de reducción iterativa del espacio de búsqueda hasta converger a un estado que asegure el mínimo uso de energía para la diseminación de requerimientos y con una probabilidad baja de pérdida de información relevante. De esta forma, se utiliza la mínima energía en la red para cumplir con la aplicación. En cierto modo, estos protocolos pueden ser vistos como una versión inversa de la búsqueda iterativa por expansión de anillos (expanding ring search, ERS). Es decir, como en nuestro caso no estamos interesados en detectar un evento (para lo cual se puede utilizar ERS), sino en computar funciones en la red basado en la búsqueda de argumentos útiles, se debe partir por una inundación controlada, y luego, en forma iterativa ir decrementando el espacio de búsqueda, no en forma regular basado en la distancia en hops de los nodos respecto al sink, como en el caso de ERS, sino en función de la distribución de los datos en la red. Sin embargo, se debe cumplir con criterios de robustez y adaptabilidad ante condiciones cambiantes, ya sea, tanto de la red como de la magnitud sensada.

Esta Tesis presenta el desarrollo de protocolos que implementan procesos de filtrado y procesamiento para *funciones de umbral*, tanto con umbral estático como dinámico. Estos protocolos están basados en la integración de *metaheurísticas* y de *algoritmos de aprendizaje inspirados en la naturaleza*. En particular, se analiza el diseño e implementación de los mismos a través de simulación. Estas estrategias son totalmente compatibles con los estándares y especificaciones actuales y poseen el potencial de mejorar otros aspectos de la performance de estas redes, como lo son su robustez ante fallos y adaptabilidad a nuevos escenarios de aplicación.

Además, como se desarrollan protocolos de procesamiento y filtrado de datos basados en búsqueda de información, embebidos en el proceso de consulta, se proponen estrategias de exploración no basadas en esquemas de comunicación *unicast*, como es habitual en la mayoría de los trabajos, sino en *multicast* o *broadcast*, los cuales son más eficiente para este tipo de aplicaciones de exploración. Es decir, cada

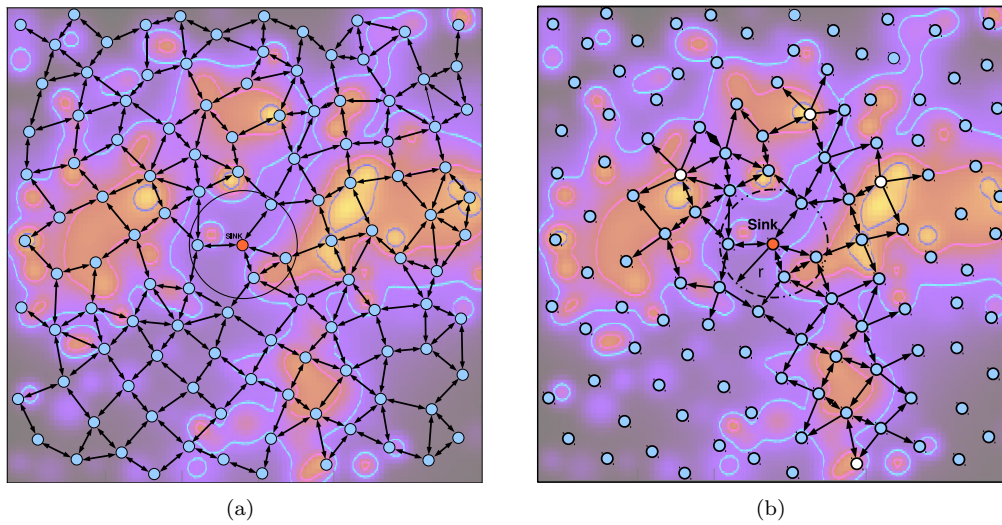


Figura 1.8: Distribución de datos en una red de sensores representada mediante isocontornos en la que todos los nodos (a), o solo algunos (b) participan del proceso de búsqueda.

nodo que recibe un requerimiento de información para diseminarlo en la red, no determina a que nodo vecino se lo retransmite (basado en información local o contenida en el paquete), sino si debe retransmitirlo a sus vecinos o no. Esto se fundamenta en la *naturaleza de broadcast de la comunicación inalámbrica*. Es decir, que cada nodo dentro del rango de comunicación de un nodo transmisor recibe los paquetes enviados. Como consecuencia se produce una paralelización de la búsqueda, o una búsqueda por múltiples caminos. En el caso de la comunicación unicast, cada nodo que recibe un paquete, determina primero si va destinado a él, y sino lo descarta. Esto es muy ineficiente desde el punto de vista del uso de la energía. Para garantizar la propagación efectiva del mensaje de requerimiento dentro de la red, se utiliza la condición de que cada nodo pueda transmitir solo una vez. Debido a que se busca que las WSNs sean de gran tamaño, lo que en el caso de redes con nodos distribuidos uniformemente se traduce en una mayor densidad de nodos, para asegurar la conectividad de la red, si por ejemplo, Por otro lado, en el proceso de respuesta o de reporte de datos, se puede utilizar cualquier esquema, tanto envío por un único camino o por múltiples caminos, cada uno con sus ventajas y desventajas, los cuales son analizados en el Capítulo 4.2.2.

El caso considerado en esta Tesis, y las hipótesis a demostrar se resumen a continuación:

Caso: computación de funciones de umbral en WSNs

Consideraciones: pocos nodos poseen información útil, correlación de datos baja o nula.

H_1 : es más eficiente realizar búsqueda de datos, que reporte de datos con agregación.

H_2 : combinar el proceso de diseminación de requerimientos con la búsqueda de información basada en gradientes permite reducir el tiempo y costo de comunicación y de procesamiento.

H_3 : utilizar reducción iterativa del espacio de búsqueda, permite adaptar la búsqueda a la distribución de los datos en la red, convergiendo a estados con el mínimo costo de búsqueda y que minimizan la pérdida de información relevante.

H_4 : el muestreo del estado de la red, integrado al proceso de diseminación de requerimientos, es más eficiente respecto al costo de comunicación, que el caso en que cada nodo realiza una actualización de su entorno de forma proactiva.

La metodología de trabajo se basa en el análisis detallado del problema, el planteo de posibles soluciones basadas en estrategias simples y escalables, el desarrollo de modelos que permitan estudiar las estrategias propuestas, el planteo a nivel de protocolos de procesamiento y de comunicación, la descripción de las fases de funcionamiento de la red, la descripción de la arquitectura implementada en cada nodo, su descripción a nivel algorítmico, la implementación en un entorno de simulación, la evaluación de los resultados obtenidos, y su comparación con los resultados obtenidos de protocolos tradicionales. Lamentablemente, para cumplimentar con el último punto anterior, no existe un conjunto de datos típico para modelar la distribución de datos (dataset), ni tampoco la disponibilidad de código fuente o la descripción detallada de la implementación de muchos protocolos propuestos en la literatura (sino solo descripciones funcionales), que permita comparar la performance de los protocolos. En base a esto se optó por utilizar protocolos tradicionales de referencia, como random walk, gossiping y flooding, los cuales son ampliamente conocidos.

1.4. Objetivos

Para cumplir con las hipótesis descritas en la sección anterior, se plantean los siguientes objetivos:

- Investigar sobre protocolos existentes para el procesamiento, filtrado, y búsqueda de información para la computación de funciones de umbral en redes inalámbricas de sensores.
- Desarrollar una arquitectura general para los nodos, que permita llevar adelante los procesos de computación y filtrado local y de nodos vecinos.
- Diseñar protocolos que permitan obtener mejoras respecto a protocolos existentes, en lo que respecta al error de cómputo, costo de comunicación, y tiempo de vida de la red.
- Validar el funcionamiento de los protocolos propuestos en la Tesis mediante análisis y simulación.
- Determinar cuales son los escenarios más adecuados para cada protocolo desarrollado.

El objetivo de esta Tesis es *proponer protocolos de procesamiento y filtrado distribuido en red*, que permitan reducir el costo de comunicación en WSNs, mediante el uso de técnicas locales de aprendizaje

y de enrutamiento de paquetes basado en gradientes de información, y heurísticas de búsqueda basadas en procesos tanto físicos como biológicos. Cabe aclarar que el costo de comunicación comienza a escalar exponencialmente a medida que aumenta ya sea, la densidad de nodos, o el tamaño de la red. Sin embargo, como se describirá en la Sección 2.5, estos parámetros son dependientes, ya que para asegurar la conectividad de la red, a medida que aumenta el tamaño de la misma, se debe incrementar la densidad de nodos.

Además, se busca incrementar las competencias locales en el campo de las WSNs, generando conocimiento innovador en un área poco desarrollada en el ámbito nacional y de gran relevancia actual. En este sentido, los resultados obtenidos durante el desarrollo de esta Tesis fueron presentados en conferencias y revistas tanto nacionales como internacionales.

1.5. Contribución

En esta Tesis se presentan tres contribuciones respecto a los protocolos de procesamiento y filtrado existentes para las WSNs. En primer lugar, se propone una arquitectura de procesamiento y filtrado, local en cada nodo, para el computo de funciones de umbral, descrito en el Capítulo 3. Luego, se propone un protocolo de filtrado de nodos vecinos sin estados, controlado en forma centralizada y optimizado con aprendizaje por refuerzo, denominado PASA, descrito en el Capítulo 4. Finalmente, en el Capítulo 5 se propone un protocolo de filtrado de nodos vecinos con estados, el cual utiliza una estrategia descentralizada, denominado PhINP.

Estas contribuciones se resumen a continuación:

- *Arquitectura de procesamiento y filtrado*, local en cada nodo, para la computación de funciones de umbral.
- *Protocolo de filtrado de nodos vecinos sin estados*, basado en recocido simulado, controlado en forma centralizada, y optimizado con aprendizaje por refuerzo.
- *Protocolo de filtrado de nodos vecinos con estados*, basado en el concepto usado en sistemas bio-inspirados de caminos marcados con niveles de feromona, y totalmente descentralizado.

1.6. Publicaciones Generadas

El aporte principal de esta Tesis se basa en el *desarrollo de protocolos de procesamiento y filtrado distribuido para el computo de funciones de umbral basados en inteligencia computacional*. Se desarrollaron dos protocolos basados en esquemas de pregunta/respuesta, con algoritmos metaheurísticos de búsqueda basados en gradientes de información y aprendizaje automático por refuerzo. A continuación, se enumeran las siete publicaciones originales que componen esta Tesis, en orden cronológico: dos publicaciones en revistas indexadas, tres de publicaciones en conferencias internacionales de primer nivel, y dos publicaciones en conferencias nacionales.

I. G. G. Riva, and J. M. Finochietto, "*Probabilistic Query Routing Scheme for Wireless Sensor*

- Networks*," Argentine Symposium on Artificial Intelligence (ASAI), JAIIO 2011, Cordoba, Argentina, August 2011 [39].
- II. J. P. Leal Licudis, J. C. Abdala, G. G. Riva, and J. M. Finochietto, "*Flexible Prototyping for Ad Hoc Wireless Sensor Network Protocols*," Argentine Symposium on Technology (AST), JAIIO 2011, Cordoba, Argentina, August 2011 [40].
 - III. G.G. Riva, and J. M. Finochietto, "*A Parallel and Adaptive Query Routing Scheme for Wireless Sensor Networks*," IEEE International Conference on Communications (ICC 2012), Ottawa, Canada, June 2012 [41].
 - IV. G. G. Riva, and J. M. Finochietto, "*Pheromone-based In-Network Processing for Wireless Sensor Network Monitoring Systems*," IEEE International Conference on Communications (ICC 2012), Ottawa, Canada, June 2012 [42].
 - V. G. G. Riva, and J. M. Finochietto, "*Pheromone-based In-Network Processing for Wireless Sensor Network Monitoring Systems*," Journal of Network Protocols and Algorithms, Macrothink Institute, vol. 4, no. 4, December 2012 [43].
 - VI. G. G. Riva, J. M. Finochietto, and G. Leguizamon, "*Bio-Inspired In-Network Filtering for Wireless Sensor Monitoring Systems*," IEEE Congress on Evolutionary Computation (CEC 2013), Cancun, Mexico, June 2013 [44].
 - VII. G. G. Riva, and J. M. Finochietto, "*In-Network Filtering Schemes for Type-Threshold Function Computation in Wireless Sensor Networks*", International Journal of Distributed Sensor Networks (IJDSN), Hindawi Publisher, Special Issue: Bio-Inspired Mechanisms in Wireless Sensor Networks, August 2014. (Impact Factor: 0.923) [45].

Las publicaciones I-III están relacionadas con los conceptos que se tratan en los Capítulos 3 y 4. Por otro lado, las publicaciones IV-VI están relacionadas con los conceptos que se tratan en los Capítulos 3 y 5. Finalmente, la publicación VII, la cual incluye todos los trabajos, integra los conceptos tratados en los Capítulos 3, 4, y 5.

Para la presentación y validación de la relevancia de las publicaciones generadas, y debido a que el trabajo desarrollado en la Tesis abarca diferentes áreas, se consideró a dos comunidades diferentes, la del *área de las comunicaciones* (publicaciones II-V y VII), y la del *área de la computación* (publicaciones I y VI).

Como fase de evaluación experimental, se realizaron pruebas de concepto en redes reales de los protocolos de computación, filtrado, y enrutamiento desarrollados en esta Tesis, como parte de proyectos integradores de la Carrera de Ingeniería en Computación de la Facultad de Ciencias Exactas, Físicas y Naturales (FCEFyN), de la Universidad Nacional de Córdoba (UNC).

1.6.1. Aportes

Los aportes principales de cada una de estas publicaciones generadas a lo largo del desarrollo de esta Tesis son descriptos a continuación:

Publicación I

G. G. Riva, and J. M. Finochietto, "Probabilistic Query Routing Scheme for Wireless Sensor Networks," Argentine Symposium on Artificial Intelligence (ASAI), JAIIO 2011, Cordoba, Argentina, August 2011.

El uso de las WSNs para descubrimiento y monitoreo de información de magnitudes físicas continuas ha emergido como una novedosa y eficiente solución. Para ello, se enrutan mensajes de requerimiento de información a los nodos de la red, para obtener datos de los mismos y reportarlos hacia el nodo sink. Como en el caso de varias aplicaciones solo se requiere un limitado subconjunto de los datos disponibles en la red (por ejemplo, encontrar el valor máximo/mínimo, o los n-valores máximos/mínimos sensados en la red), este requerimiento de información puede ser idealmente enrutado o dirigido a aquellos nodos que tengan datos relevantes. Esto posibilita reducir el costo de comunicación entre nodos, que es el principal factor de consumo de energía en estas redes.

En este trabajo se considera la aplicación de inteligencia computacional en los nodos que permita basado en un esquema probabilístico de diseminación de mensajes de requerimiento a los nodos de la red, utilizado para computar funciones de umbral en forma distribuida, mediante el cual se puede reducir el costo de comunicación de la red. Este esquema de diseminación realiza una búsqueda probabilístico de información basada en gradientes de información. Para ello se desarrolló un algoritmo denominado *Parallel Adaptive Simulated Annealing (PASA)*, basado en la metaheurística de recocido simulado, multi-instanciado en cada nodo de la red cuando se recibe un requerimiento. El control o configuración de la búsqueda es realizada por el nodo sink en forma centralizada, basado en aprendizaje por refuerzo. Luego de varias iteraciones de búsqueda, se converge a un estado de mínima energía necesaria para llegar con un requerimiento a aquellos nodos con datos relevantes para la aplicación. Esta forma de consulta dirigida y respuesta, permite muestrear y adaptarse al estado actual de la red en cada iteración de computación, dada la gran variabilidad temporal de este tipo de redes. Debido a que en los nodos sensores no mantienen ningún estado local sobre estados previos, se denomina a este, protocolo *sin estados*.

Además, se describen los beneficios de utilizar PASA respecto a esquemas tradicionales como Flooding y Random Walk, respecto al error y al costo de comunicación. PASA permite obtener una reducción del costo de comunicación del 60 %, con un error relativo máximo del 5 % respecto a Flooding.

Publicación II

J. P. Leal Licudis, J. C. Abdala, G. G. Riva, and J. M. Finochietto, "Flexible Prototyping for Ad Hoc Wireless Sensor Network Protocols," Argentine Symposium on Technology (AST), JAIIO 2011, Cordoba, Argentina, August 2011.

El desarrollo tanto de sofisticados protocolos eficientes energéticamente como del incremento en la complejidad de las aplicaciones para las WSNs, impone el uso de arquitecturas de programación abiertas y flexibles que brinden acceso y permitan la interacción con el stack de comunicación de manera simple. Los desarrollos en el área de las WSNs no se enfocan solamente al desarrollo de mecanismos eficientes a nivel de aplicación, sino también en la interacción con el stack de comunicaciones. Debido a que el principal consumo de energía en un nodo sensor está en el uso de la radio, el principal desafío es el de reducir los costos de comunicación por medio de procesamiento en red.

En este trabajo se analizan los sistemas operativos (SO) existentes para las WSNs, se analizan sus

ventajas y desventajas, y se propone a Contiki como el más conveniente. Contiki es un sistema operativo de código abierto, que posibilita, mediante su arquitectura, el prototipado y la implementación flexible y rápida de protocolos, y algoritmos para las WSNs. Además, se describe el stack de comunicaciones propio de Contiki, utilizado en las implementaciones, denominado Rime. Rime es un stack muy liviano, el cual reduce la complejidad de implementación. Debido a que Contiki también soporta el stack uIPv6, se realizó una comparación respecto al consumo de energía de una red utilizando en cada nodo el stack de comunicaciones uIPv6, respecto a utilizar el stack Rime.

Para validar el funcionamiento de esta arquitectura para prototipado rápido, se realizaron primero simulaciones y luego implementaciones en nodos reales. Se implementó una primera versión del algoritmo PASA, descrito en Publicación I, para reducir el espacio de diseminación de requerimientos a una red de sensores basado en un algoritmo metaheurístico multi-instanciado. Este se basa en el algoritmo de Recocido Simulado, pero a diferencia de este, se lo adapta para ser implementado en forma paralela en una WSN.

Publicación III

G.G. Riva, and J. M. Finochietto, "A Parallel and Adaptive Query Routing Scheme for Wireless Sensor Networks," IEEE International Conference on Communications (ICC 2012), Ottawa, Canada, June 2012.

Este trabajo está basado en la Publicación I. Se incorpora un análisis adicional sobre la conveniencia de utilizar un esquema de pregunta/respuesta respecto a uno de una sola pregunta y proactivo. Se muestran los beneficios respecto a error y costo de comunicación respecto a utilizar protocolos tradicionales como Flooding y Random Walk. Además, se muestra la conveniencia de utilizar esquemas de tipo pregunta/respuesta, ante los de tipo proactivos basados en un solo requerimiento inicial, en escenarios donde los datos, obtenidos de la magnitud sensada, no son estáticos. Además, de los beneficios de minimización tanto del error de computación como del costo de requerimiento de utilizar PASA respecto al uso de protocolos tradicionales de diseminación y enrutamiento, como Flooding y Random Walk.

Publicación IV

G. G. Riva, and J. M. Finochietto, "Pheromone-based In-Network Processing for Wireless Sensor Network Monitoring Systems," IEEE International Conference on Communications (ICC 2012), Ottawa, Canada, June 2012.

El monitoreo de magnitudes físicas continuas temporal y espacialmente ha emergido como una novedosa y eficiente solución. El desarrollo de algoritmos eficientes energéticamente, tanto para la diseminación de requerimientos hacia la red, como para el envío de datos hacia el nodo sink, en aplicaciones donde solo algunos nodos pueden aportar datos relevantes, es un problema desafiante, especialmente si no se cuenta con información acerca de la localización de esos nodos. Una solución a este problema es dejar a los nodos cooperar y decidir en forma conjunta que datos son relevantes y su localización, imitando al comportamiento de colonias de hormigas que refuerzan caminos con feromona hacia fuentes de alimento en forma cooperativa.

En este trabajo se propone un protocolo probabilístico de procesamiento en red de funciones de umbral basado en un esquema de pregunta/respuesta, denominado *Pheromone-based In-Network Processing, PhINP*. La diferencia con el protocolo PASA, propuesto en las Publicaciones I-III, es que en este caso no se requiere de un control o configuración de búsqueda centralizada en el nodo sink.

En este caso, cada nodo mantiene un estado local de estados previos, denominado *nivel de feromona*, que es utilizado para mantener un estado de memoria del comportamiento previo de la red. En este sentido, se denomina a este como *protocolo con estados*.

Si bien la utilización de algoritmos basados en colonias de hormigas (ant colony optimization, ACO) ha sido propuesta para su utilización en las WSNs, la forma de implementación y aplicación propuesta en este trabajo es totalmente nueva.

Publicación V

G. G. Riva, and J. M. Finochietto, "*Pheromone-based In-Network Processing for Wireless Sensor Network Monitoring Systems*," Journal of Network Protocols and Algorithms, Macrothink Institute, vol. 4, no. 4, December 2012.

Este trabajo es una extensión de la Publicación IV, en el que se presentan nuevos análisis respecto al funcionamiento del protocolo PhINP. Por ejemplo, como varía el error de computación, y la velocidad de convergencia para diferentes configuraciones de los niveles de feromona. Además, se analiza la capacidad de readaptación de PhINP respecto a los protocolos Flooding y Gossip, cuando varían la amplitud de los eventos producidos en la red.

Publicación VI

G. G. Riva, J. M. Finochietto, and G. Leguizamon, "*Bio-Inspired In-Network Filtering for Wireless Sensor Monitoring Systems*," IEEE Congress on Evolutionary Computation (CEC 2013), Cancun, Mexico, June 2013.

En esta Publicación, que es una versión mejorada y extendida de las publicaciones IV y V, se formula matemáticamente el computo de funciones umbral en las WSNs. En este sentido, el computo se realiza en dos etapas secuenciales. Se hace una pre-computación en la red, y posteriormente, cuando el nodo sink recibe esta información, realiza el computo final de forma centralizada.

Una de las mejoras incluidas en esta Publicación, respecto a las Publicaciones anteriores, es la inclusión de un esquema de envío de datos de los nodos seleccionados hacia el nodo sink por un único camino (one-path routing), en lugar de un esquema de envío por múltiples caminos como en las versiones anteriores. Otra diferencia respecto a las Publicaciones anteriores, es el modelado de la superficie que representa la magnitud física monitoreada. En los trabajos previos se consideraba a esta superficie como formada por múltiples eventos, los cuales eran modelados como funciones exponenciales decrecientes. En esta publicación, se modelan los eventos por funciones que siguen una ley de difusión con la distancia. Finalmente, se analiza y estima cuantitativamente el tiempo de vida de la red basado en la cantidad de iteraciones de computación que pueden ser llevadas a cabo en la red.

Publicación VII

G. G. Riva, and J. M. Finochietto, "*In-Network Filtering Schemes for Type-Threshold Function Computation in Wireless Sensor Networks*," International Journal of Distributed Sensor Networks (IJDSN), Hindawi Publisher, Special Issue: Bio-Inspired Mechanisms in Wireless Sensor Networks, August 2014.

En esta Publicación se presenta la arquitectura implementada en cada nodo para el procesamiento de funciones de umbral, el auto-filtrado, y el filtrado de nodos vecinos sin estados (protocolo PASA) y sin estados (protocolo PhINP). Se analizan en detalle los protocolos PASA y PhINP propuestos, y se realizan comparaciones de desempeño de ambos, con el fin de detectar fortalezas y debilidades,

y los casos de aplicación mas adecuados para cada uno de ellos. Se analiza el funcionamiento de ambos protocolos, respecto al error y al costo de comunicación, para diferentes cantidades de eventos presentes en la red, determinando la peor condición de filtrado.

Adicionalmente, se proponen mecanismos de auto-curado, basados en estrategias de escucha activa y aprendizaje por refuerzo, para evitar la pérdida de paquetes de respuesta, en los que se envían los argumentos o datos relevantes para computar la función solicitada por el usuario. Ello se debe a que en las WSNs no se utilizan transmisiones seguras basadas en confirmación de recepción. Con este mecanismo propuesto, un nodo que envía un paquete de respuesta, al nodo vecino desde el cual recibió el requerimiento, y detecta que este no lo ha retransmitido con destino hacia el sink, luego de un cierto tiempo lo retransmite a otro nodo vecino desde el cual recibió el mismo requerimiento la segunda vez.

1.7. Estructura General de la Tesis

Esta Tesis se estructura en 7 capítulos. En el Capítulo 1 se presentan la pregunta que busca responder esta Tesis, el problema de investigación, los antecedentes, la hipótesis y metodología de trabajo, los objetivos de la Tesis, la contribución, y las publicaciones generadas. En el Capítulo 2 se presenta el marco teórico, describiendo los modelos utilizados en la literatura, y los usados en esta Tesis. En el Capítulo 3 se describen los procedimientos utilizados para el procesamiento y filtrado de los datos, y la arquitectura propuesta. En el Capítulo 4 se presenta un protocolo de procesamiento y filtrado en red sin estados en los nodos que conforman la red, el cual es configurado en forma centralizada por el nodo sink. Por otro lado, en el Capítulo 5 se presenta un protocolo alternativo de procesamiento y filtrado en red pero considerando que cada nodo mantiene un estado local, y se utiliza un esquema de toma de decisiones totalmente descentralizado basado en mecanismos bio-inspirados. El análisis de los resultados obtenidos y de la performance de ambos protocolos propuestos respecto a protocolos tradicionales, y la definición de las áreas de aplicación más adecuadas para cada uno de ellos es abordado en el Capítulo 6. Finalmente, en el Capítulo 7 se presenta las conclusiones generales de esta Tesis. Esta estructura es descripta en la Figura 1.9.

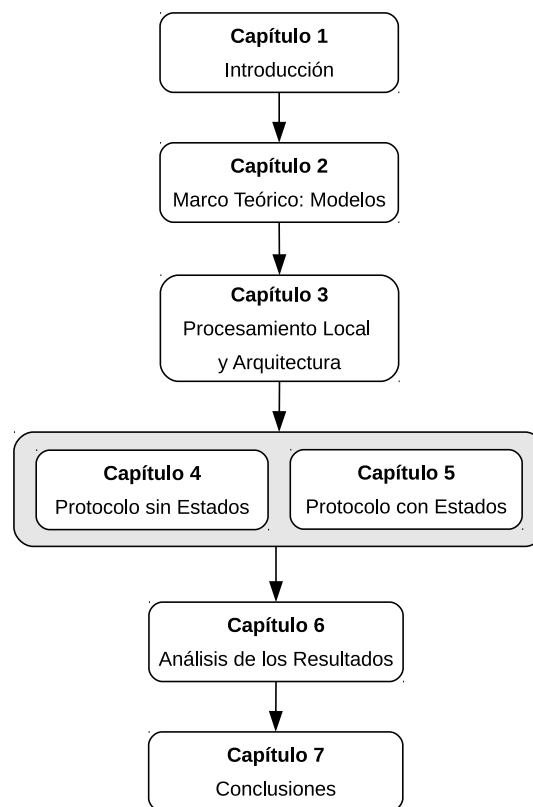


Figura 1.9: Estructura general de la Tesis.

Marco Teórico: Modelos

Contenido

2.1	Modelos de Datos	28
2.2	Modelos de Computación y Procesamiento	31
2.2.1	Definiciones de Términos	31
2.2.2	Procesamiento en redes tradicionales y en WSNs	32
2.2.3	Tipo de Funciones Analizadas	38
2.2.4	Funciones de Umbral	40
2.3	Modelos de Enrutamiento	40
2.3.1	Modelos de Reporte de Datos	45
2.4	Modelos de Energía	46
2.5	Modelos de Conectividad y Comunicación	47
2.6	Comentarios de Cierre	49

EN este capítulo se describen los modelos utilizados en la literatura para representar tanto el fenómeno o magnitud física de interés, de la cual los nodos obtienen los datos, el funcionamiento de cada nodo, como la interacción entre los nodos que conforman la red. Estos abarcan desde modelos de comunicación, de enrutamiento de paquetes, de generación de datos, y de cómo son modeladas las magnitudes físicas a monitorear. A fin de una mejor organización, se abordarán en este capítulo los modelos desde las capas superiores hasta las inferiores, según se indica en la Figura 2.1. Es decir, se tratará inicialmente el modelado del fenómeno físico de interés, pasando por las capas que modelan el comportamiento del nodo, como son las capas de aplicación, de red, de energía, y finalmente el modelado de la comunicación entre los nodos. El modelado de las dos capas inferiores del nodo (de enlace y física) no son incluidas, debido a que las mismas se consideran ideales y no se utilizan modelos específicos. Es decir, no se modela el control de acceso al medio por parte de cada nodo, ni el modelado de la radio utilizada. A continuación, se describe cada uno de los modelos propuestos en la literatura y los utilizados en esta Tesis.

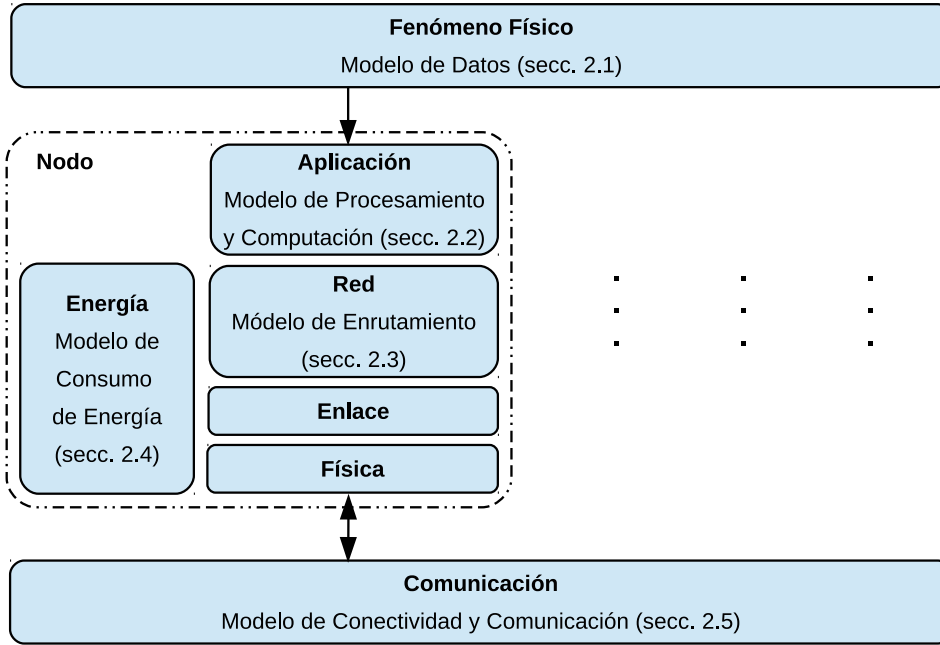


Figura 2.1: Modelos utilizados.

2.1. Modelos de Datos

Existen diferentes alternativas para representar a los datos (o lecturas) obtenidos por los nodos sensores sobre un fenómeno físico de interés: i) un *caso realista, basado en modelos de distribución de datos*, en la que cada nodo obtiene su lectura muestreando a la función que representa al fenómeno físico, en función de su posición dentro de la red e instante de tiempo, ii) un *caso realista, basado en conjuntos de datos conocidos (datasets)*, donde la posición de los nodos es un parámetro predefinido, y iii) un caso extremo para la búsqueda con gradientes, *basado en datos artificiales generados pseudo-aleatoriamente en cada nodo*, en el que no hay relación entre la posición del nodo y el valor sensado.

En la primer alternativa, se consideran funciones de tipo superficie $f(\underline{x}, \underline{y})$, las cuales representan la distribución espacial de la magnitud física de interés sobre los coordenadas \underline{x} e \underline{y} , en un instante de tiempo t . Particularmente, cada nodo sensor s obtiene una lectura estimada $\hat{x}'_s = f(\underline{x}_s, \underline{y}_s)$ de dicha magnitud con una cierta exactitud. Esta lectura está basada en la posición geográfica del nodo $(\underline{x}_s, \underline{y}_s)$ en el instante de tiempo de muestreo t_i o iteración i , siendo el valor real de la magnitud en ese punto x' . Sin embargo, si la *exactitud* $\rightarrow \infty$, y la *precision* $\rightarrow \infty$, implica que $\hat{x}'_s \rightarrow x'_s$. En este trabajo se considera esta última condición, es decir, error de sensado nulo ($e_{sens} = 0$). Esta condición permite analizar el funcionamiento de los protocolos propuestos independientemente del error cometido por los sensores. Sin embargo, considerar el error de sensado durante el funcionamiento de la red es la condición más real.

Por otro lado, en la segunda alternativa, se utilizan conjuntos de datos o datasets (en forma de tablas), obtenidos previamente de mediciones reales. Sin embargo, si bien esta alternativa posibilita la comparación de performance de diferentes protocolos, ya que se tiene un escenario bien definido y determinístico, es poco flexible desde el punto de vista de la cantidad de casos considerados. Es decir, la posición de los nodos es predefinida, lo que no permite considerar múltiples despliegues aleatorios de

los nodos sensores en el área sensada. Además, los datasets existentes pertenecen a redes reducidas de no más de 100 nodos, en las que se obtienen, en general, datos de humedad, temperatura y luz [46, 47].

Finalmente, en la tercer alternativa, se considera que cada nodo sensor s obtiene su valor sentido de forma sintética o artificial, mediante generación pseudo-aleatoria, es decir $\hat{x}'_s = rand()$. De esta forma, los datos obtenidos por los nodos no están relacionados o correlacionados, y se dice que la correlación $corr \rightarrow 0$. En este caso, como así también el aquel caso en el cual la resolución de sentido sea muy alta, la agregación de datos basado en la correlación de los datos no es eficiente.

Uno de los temas recurrentes, tanto en el modelado de datos como en el desarrollo de procesamiento de datos dentro de una WSN, es la *correlación de datos* ($corr$). Esta se define como la similitud existente entre muestras del fenómeno físico. En general, se definen dos tipos de correlación: la *espacial*, basada en la similitud entre los datos o muestras obtenidas por nodos vecinos, y la *temporal*, basada en la similitud entre muestras sucesivas obtenidas por el mismo nodo. En este contexto, dada la alta correlación de datos existente en una WSN, debido principalmente a que los nodos sensores tienen una resolución de sentido finita y cuantificada, y a que el parámetro a medir puede tener una distribución espacial y temporal poco variable, se han propuesto en los últimos años una gran cantidad de protocolos para reducir la comunicación de datos en la red. En este sentido, estrategias basadas en agregación y fusión de datos han sido las de mayor relevancia, las cuales se describirán en la Sección 2.2. Sin embargo, en el desarrollo de estos protocolos, no se aborda el caso en el que no exista correlación entre los datos (es decir, $corr = 0$).

Si bien en esta Tesis se han evaluado las tres alternativas anteriores para representar a los datos obtenidos por los nodos sensores, se ha optado por utilizar la primera (basada en modelos de distribución de datos), debido a su flexibilidad para definir y evaluar diferentes escenarios, permitiendo configurar además diferentes tipos de comportamiento dinámico. Bajo esta forma de representación, la magnitud física puede ser modelada i) por la sumatoria de fuentes puntuales de datos o de información con diferentes posiciones y amplitudes, o ii) por una función continua. La primera considera por ejemplo, fenómenos físicos en los que existen fuentes de datos, como por ejemplo fuentes de calor, de luz, de ruido, etc, representando a la magnitud como la sumatoria de las contribuciones de todas las fuentes existentes. La segunda está más relacionada con parámetros que tienen una distribución espacial mas gradual, por ejemplo, parámetros ambientales, como la humedad y temperatura ambiente, presión atmosférica, etc. Sin embargo, el segundo tipo puede ser considerado como una variante del primero, en el que la difusión de las fuentes de datos tiende a infinito.

En el caso de la representación de la magnitud física modelada por fuentes de datos puntuales distribuidas en la red, se pueden utilizar diferentes tipos de funciones para representar a dichas fuentes. Para tal fin, se utilizaron inicialmente *funciones exponenciales* del tipo $f(e^{-x*x} * e^{-y*y})$, como la mostrada en la Figura 2.2a. Posteriormente, se utilizaron *funciones que siguen leyes de difusión con la distancia*, que representan mejor al comportamiento de las fuentes reales (Figura 2.2b). Estas últimas son del tipo $f(1/d(\cdot)^\alpha)$, donde $d(\cdot)$ representa la distancia euclídea $\sqrt{(x_f - x_s)^2 + (y_f - y_s)^2}$ entre la posición de la fuente de datos (x_f, y_f) y la posición del nodo sensor (x_s, y_s) , y α es el coeficiente de difusión de la magnitud física. Por ejemplo, se consideran valores de $\alpha \approx 1$ para el caso de calor, y $\alpha \approx 2$ para el de la luz. En este sentido, la mayoría de los fenómenos físicos (luz, calor, ruido, radiación, etc) siguen una ley de este tipo [35].

Desde el punto de vista de la *capacidad de detección de fuentes de información por parte de los*

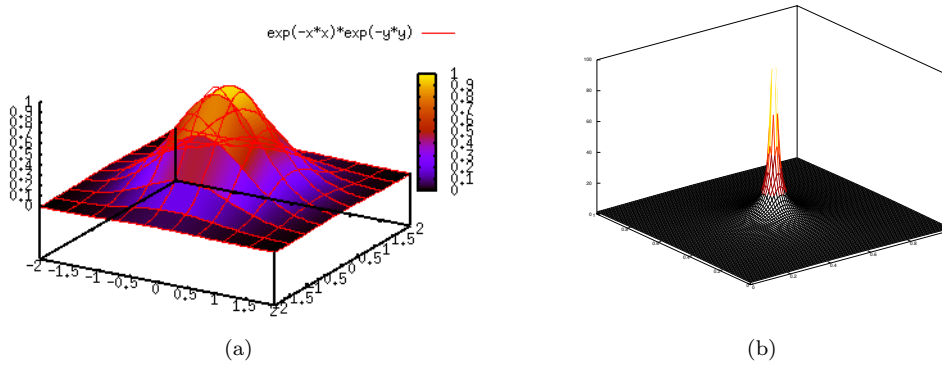


Figura 2.2: Modelos de fuentes de datos utilizados, basados en funciones (a) exponenciales, o (b) que siguen leyes de difusión con $\alpha = 1$.

nodos de la red, si se considera que los nodos tienen una resolución de sensado infinita, es decir, que no hay error de cuantificación, cada uno de ellos puede detectar fuentes existentes en la red. Sin embargo, la capacidad para detectar estas fuentes por parte de nodos con una resolución finita, es función de varios factores. Los principales son, la densidad de la red ρ , el rango de comunicación de los nodos r , la precisión y sensibilidad de los sensores, y la ley de difusión $1/d(\cdot)^\alpha$ de la magnitud física considerada. Sin embargo, en esta Tesis consideramos que los nodos tienen una resolución de sensado infinita, ya que ese es el caso más general, y al comenzar a introducir resoluciones finitas, con su consecuente error de cuantificación, se comienza a perder capacidad de detección y comienza a producirse la correlación de datos.

En el caso de que existan varias fuentes de información en un área sensada, el valor \hat{x}'_s obtenido por el nodo sensor s se define como la contribución o sumatoria de todas ellas, como:

$$\hat{x}'_s \propto \sum_{k=1}^F \frac{1}{d(s, k)^\alpha}. \quad (2.1)$$

donde F es el número de fuentes presentes en la red, $d(\cdot)$ es la distancia euclídea entre el nodo sensor s y la fuente k , y α es el coeficiente de difusión. Ejemplos de superficies conformadas con los dos tipos de funciones analizadas son descriptos en la Figura 2.3.

En base a las superficies que modelan la distribución de los datos en la red (Figura 2.3), se definieron dos tipos de análisis, basados en i) una cantidad constante de fuentes de información ($F = cte$), o en ii) un porcentaje de fuentes proporcional a la cantidad de nodos presente en la red ($F \propto n$). En el primer caso, se define una superficie formada por una cantidad constante F de fuentes de información (específicamente, $F = 160$) con diferentes amplitudes (Figura 2.3a). En caso de que se incremente la cantidad de nodos o tamaño de la red, y dado que se debe mantener una densidad constante de nodos, tanto el área abarcada por la red, como la superficie que modela la magnitud sensada son expandidos respecto a los ejes x , y . En el segundo caso se considera que esta superficie esta formada por un porcentaje de fuentes de información proporcional a la cantidad de nodos de la red, y para simplificar el análisis, se considera que las mismas tienen o emiten la misma magnitud (Figura 2.3b). Estas fuentes pueden ser objetos tales como lámparas de luz, aires acondicionados, calefactores, fuentes de ruido, etc.

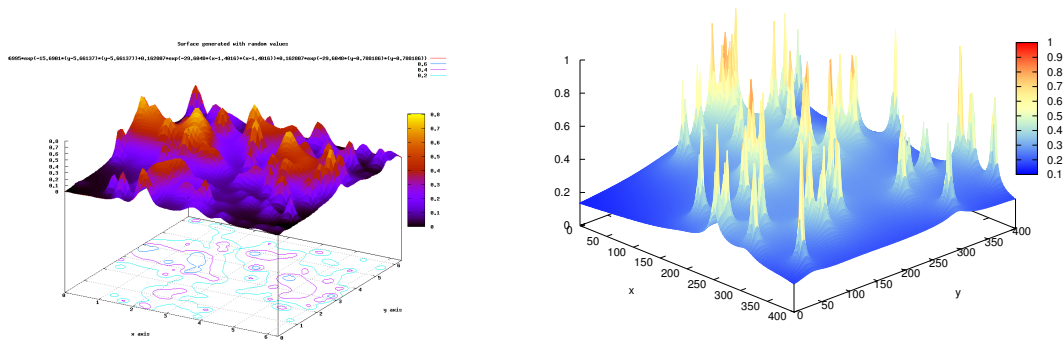


Figura 2.3: Representación de la magnitud a sensor basada en: funciones exponenciales con diferentes amplitudes (izquierda), y en ley de difusión con $\alpha = 1$ y amplitudes constantes (derecha).

Afortunadamente, el entorno de simulación desarrollado en esta Tesis, permite definir de forma sencilla mediante archivos de configuración, como se genera la magnitud sensada de la cual cada nodo obtiene su lectura en base a la Ecuación 2.1. En este sentido, se puede configurar el tipo, cantidad, posición y amplitud de las fuentes. Además, se puede definir el comportamiento dinámico, es decir, como varían estas fuentes con el tiempo, o agregar/eliminar ciertas fuentes en un instante de tiempo dado, entre otros.

2.2. Modelos de Computación y Procesamiento

En esta sección se describen los *modelos de computación y procesamiento de datos e información desarrollados para redes de comunicaciones inalámbricas*, los diferentes tipos, sus principales características, y las particularidades existentes en el caso de las WSNs. Inicialmente, vamos a repasar el significado de diferentes términos y conceptos utilizados a lo largo de esta Tesis, y que muchas veces son considerados equivalentemente.

2.2.1. Definiciones de Términos

2.2.1.1. Dato, información, y conocimiento

Existe una sutil diferencia entre los términos *dato e información*. Los datos son hechos de los cuales se deriva la información, siendo los datos individuales raramente útiles. Para que los datos se transformen en información, estos deben estar dentro de un contexto, ser procesados, organizados, estructurados, o presentados en un contexto dado que los hagan útiles. En este sentido, las lecturas obtenidas por los nodos sensores son denominados datos, y el resultado del procesamiento de estos datos constituye información, considerando como procesamiento a cualquier operación realizada con los mismos que produzca un resultado. El conocimiento surge como resultado de la información y la experiencia obtenida de la ejecución de un proceso a lo largo del tiempo.

2.2.1.2. Computación y procesamiento

Normalmente, se considera a los términos *computación* y *procesamiento* como sinónimos. La *computación* está relacionada con el procesamiento de la información, pero existen diferencias sustanciales, las cuales muchas veces son omitidas. El término *procesamiento de información* es muy amplio, e incluye una gran cantidad de operaciones posibles, por ejemplo, el procesamiento de las cabeceras de los paquetes, el procesamiento de los datos transportados en los paquetes como la agregación, la fusión, el filtrado de datos, entre otros. Por otro lado, la *computación*, palabra que deriva de cómputo o cálculo, es definida como el proceso algebraico que permite generar resultados correctos mediante el uso de un procedimiento efectivo [48].

2.2.2. Procesamiento en redes tradicionales y en WSNs

Las tareas de procesamiento son funciones que están cada vez más presentes en las redes de comunicaciones, y son de extrema importancia, dadas las actuales demandas de optimizar los recursos tanto energéticos como de comunicación. Sin embargo, existen diferencias sustanciales entre las redes de comunicación tradicionales y las WSNs, desde diferentes puntos de vista, como por ejemplo, *qué tipo de elemento se va a procesar*, y *adónde y cómo es realizado el mismo*. A continuación, se analiza cada uno de estos puntos.

2.2.2.1. Qué se va a procesar

En las *redes de comunicación tradicionales*, cada nodo normalmente genera paquetes de datos dirigidos hacia otro nodo de destino (por lo general distante), lo que se conoce como comunicación punto a punto. En este contexto, los nodos intermedios tienen la función de dirigir o enrutar los paquetes hacia el nodo de destino, bajo un cierto criterio o métrica (por ejemplo, camino más corto, de menor tráfico, etc). Para este fin, los nodos intermedios pueden leer y modificar parte de los encabezados de los paquetes, como por ejemplo, la cantidad de saltos que realizó el paquete (hop count), a qué nodo vecino se le retransmite el paquete, etc, como se indica en la Figura 2.4a. Sin embargo, estos nodos no tienen permitido ni leer ni modificar el contenido del paquete (payload), ya que esto es una violación al modelo OSI (Open Systems Interconnection). Un paquete transmitido por un dispositivo de comunicación está conformado por un encabezado (H, header), y por el dato (DATO). Luego de la intervención de un nodo intermedio, el contenido del encabezado del paquete retransmitido es modificado, denotado por el cambio de H_1 a H_2 , mientras que el dato contenido en el mismo se mantiene inalterado.

En contraste, en el caso de las WSNs, durante el proceso de reporte de datos, los nodos no solo envían paquetes de datos hacia el mismo nodo de destino (nodo sink), sino que también los nodos intermedios pueden procesar tanto los encabezados de los paquetes, como en el caso de las redes tradicionales, como así también realizar algún tipo de procesamiento o computación local de los datos o información transportada en el payload de los mismos. Este tipo de procesamiento, que tiene lugar durante la retransmisión de los paquetes en la red, se conoce como *on-the-fly processing*. Afortunadamente, en este tipo de redes, a excepción de las anteriores, esto no implica una violación al modelo OSI, sino más bien una optimización de capa cruzada (cross-layer optimization) que es

utilizada para hacer más eficiente el funcionamiento de estas redes. En la Figura 2.4b, se puede observar que tanto el header como el dato transportado en el paquete pueden ser modificados por un nodo retransmisor.

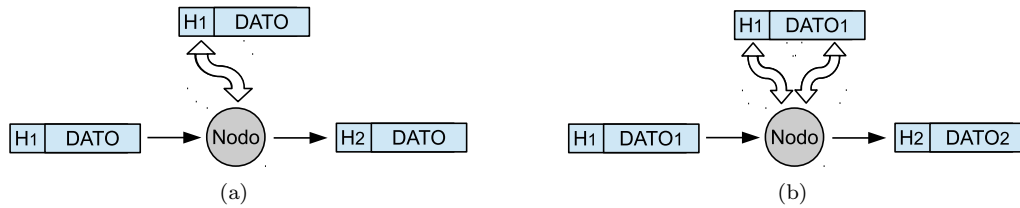


Figura 2.4: Elementos de un paquete que pueden ser procesados por un nodo retransmisor: (a) en redes de comunicaciones tradicionales, y (b) en WSNs.

El objetivo de este tipo de optimización es reducir la cantidad de paquetes comunicados, mediante la implementación de parte del procesamiento dentro de la red, durante el reporte de los datos hacia el nodo sink. Además, la aplicación que se ejecuta en el nodo puede tener ingerencia en la información manejada en cada una de las capas inferiores, sin tener que utilizar un esquema secuencial, en el que cada capa inferior le provee servicios a la capa superior inmediata. En este sentido, el procesamiento de los datos o de la información en los nodos que conforman un sistema de sensado distribuido posibilita la implementación de técnicas de reducción del volumen de paquetes de datos transferido hacia el nodo sink, disminuyendo tanto el costo de comunicación como la carga de procesamiento en el sink.

A partir de ahora, nos enfocaremos en el procesamiento de los datos transportados en la carga útil (payload) de los paquetes, ya que esta es la propiedad que distingue a las WSNs de las redes tradicionales, y en la que se basan los protocolos propuestos en esta Tesis para optimizar el uso de los recursos limitados con los que cuentan los nodos sensores.

2.2.2.2. Adonde y como se va a procesar

Respecto al lugar donde se realiza el procesamiento o computación de los datos o de la información en las WSNs, se distinguen dos casos bien diferenciados, el *centralizado*, y el *distribuido*. El primero considera que todos los paquetes son enviados al nodo sink, para el procesamiento de los datos contenidos en los mismos, ya que el mismo posee mayor capacidad de procesamiento que los demás nodos. Por otro lado, el segundo caso considera que los datos son procesados por los nodos dentro de la red de forma colaborativa y coordinada mientras son reportados hacia el nodo sink. A continuación se analiza cada uno de estos casos.

En el *esquema centralizado*, se concentra toda la capacidad de procesamiento en el nodo sink, dejando a los nodos de la red solamente la tarea reporte y retransmisión de paquetes de datos. Esto simplifica en gran medida el hardware de los nodos sensores, sin embargo, el costo de comunicación para reportar los paquetes de datos comienza a incrementarse a medida que las redes son más grandes, debido a su naturaleza multi-hop.

En cambio, en el *esquema distribuido*, la mayor tarea de procesamiento es realizada dentro de la red por los nodos sensores, entregando al nodo sink argumentos o información pre-procesada, con la cual este último puede obtener fácilmente el resultado buscado. Los mecanismos de procesamiento más utilizados bajo este tipo de esquema son la agregación y la fusión de datos. Afortunadamente, este tipo

	Tipo de procesamiento	
Característica	Centralizado	Distribuido
Latencia de procesamiento	baja	media, alta
Sincronización entre nodos	simple, no relevante	compleja, relevante
Demanda de hardware de nodos sensores	baja	media
Aplicación en redes	chicas, no densas	medias o grandes, densas

Tabla 2.1: Características de los diferentes tipos de esquemas de procesamiento.

de esquema permite reducir en gran medida el costo de comunicación respecto al caso centralizado. Sin embargo y como desventaja, se penaliza en gran medida el tiempo, dado que se producen demoras sustanciales en la obtención del resultado, debido a las capacidades limitadas de computación de los nodos. Cabe aclarar, que en los nodos se utilizan normalmente microcontroladores de 8 o 16 bits, de bajo consumo y bajas prestaciones computacionales, ya que existe un fuerte compromiso entre ambas. Además, en este esquema se incrementan los requerimientos de hardware de los nodos, y se requiere de estrategias de sincronización entre los nodos, para realizar esta tarea (de reporte y procesamiento) de manera sincronizada.

Como conclusión, se puede observar que no existe un esquema óptimo de procesamiento para todos los tipos de aplicaciones posibles. Cada esquema presenta ventajas y desventajas que hacen a cada uno de ellos más eficiente para cada tipo de aplicación. Por ejemplo, el esquema centralizado es más adecuado o conveniente en el caso de WSNs chicas, en aplicaciones críticas en tiempo, y en redes en las que los nodos sensores no tienen capacidad de procesamiento, y viceversa. Un resumen de estas características es descrito en la Tabla 2.1.

A lo largo de este trabajo, se va a representar a la red de nodos, como una *topología jerárquica*, en la que se tiene al nodo sink como raíz o padre general de los demás nodos. Cada nodo puede tener tanto un padre como uno o más hijos, y a su vez, puede ser padre e hijo de otros nodos, según su posición en la red. En este sentido, la *posición de cada nodo* puede ser definida, por ejemplo, en función de su distancia en saltos respecto al nodo sink. Sin embargo, esta distancia puede ser determinada por cada nodo, durante la primera fase de diseminación de requerimientos, mediante la determinación del paquete recibido con la menor cantidad de saltos realizados. Por ejemplo, si se reciben tres copias del mismo requerimiento, con una cantidad de saltos (hops) de 2, 3, y 4, el nodo puede determinar claramente que él se encuentra a 3 ($hops + 1$) saltos de distancia del sink, o lo que su *hop level* = 3. El tráfico de paquetes de datos es dirigido desde los nodos hijos hacia los padres, hasta que los datos alcanzan al nodo sink, y viceversa para el tráfico de consultas o de diseminación de requerimientos hacia los nodos de la red. En este contexto, un nodo s_1 cuya distancia al nodo sink es *hop level*₁, tendrá un nodo padre s_2 localizado a una distancia *hop level*₂ < *hop level*₁, y uno o más nodos hijos localizados a un *hop level* > *hop level*₁. Sin embargo, en el caso de reporte de datos por múltiples caminos (por broadcast o multicast), un nodo particular puede tener varios nodos padres.

En el caso particular de no aplicar ninguna técnica de procesamiento y reducción de paquetes de datos (o de datos) en la red, durante el proceso de reporte de los mismos hacia el sink, y de realizar el procesamiento de forma centralizada en el nodo sink, se incurre en un elevado costo de comunicación, debido fundamentalmente al gran tráfico resultante (Figura 2.5a). En este contexto, cada nodo debe reportar no solo su dato, sino el de todos sus nodos hijos. El costo de comunicación es igual a la sumatoria del producto entre la cantidad de nodos en cada nivel y su correspondiente distancia al

sink.

Normalmente, la mayoría de los protocolos de procesamiento distribuido, desarrollados para reducir el costo de comunicación, se basan en la agregación o fusión de datos durante el proceso de reporte de los mismos [49, 50, 9, 30, 31]. Es decir, el reporte de datos comienza desde los nodos más distantes respecto al nodo sink, de manera sincronizada. Cada nodo padre, recibe los datos obtenidos por sus nodos hijos, realiza el procesamiento parcial de los mismos, y genera un paquete con el resultado de la computación para enviar a su nodo padre. Este mecanismo se repite con cada nodo de la red, y posibilita que el nodo sink reciba los datos pre-procesados (Figura 2.5b). Sin embargo, el procesamiento puede ser con o sin reducción del tamaño de los paquetes. En este caso, se logra una reducción significativa en el tráfico de paquetes reportados al nodo sink. En este contexto, el costo de comunicación es proporcional al tamaño de la red n , lo que permite obtener, respecto al caso anterior, una reducción de este costo proporcional al orden de la red N . Estos esquemas de procesamiento son eficientes tanto en casos donde *i) gran parte de los datos son relevantes para el procesamiento que se quiere realizar* (por ejemplo, el cómputo del valor medio), como en *ii) redes que no presentan cambios temporales trascendentes en su topología*. Para remediar el caso en que esta última situación no se cumpla, la mayoría de las soluciones propuestas asignan a cada nodo la tarea de explorar temporalmente su entorno, obteniendo información actualizada de sus nodos vecinos, necesaria para las próximas transmisiones de datos. Sin embargo, esto demanda un costo elevado de energía. Sin embargo, la mayor parte de las técnicas y protocolos propuestos en la literatura para reducción de tráfico en WSNs se focalizan en el uso de procesamiento de datos basado en agregación. Se ha demostrado analíticamente que la máxima ganancia (respecto a la reducción en el uso de energía) obtenida en la agregación de datos se obtiene cuando las fuentes de información están juntas entre ellas y lo más distante posible del nodo sink [49]. Pero en general, la formación de un árbol de agregación óptima de datos es un problema NP-hard, y en la literatura se proponen versiones subóptimas.

Sin embargo, en el caso del cómputo de cierto tipos de funciones de los datos obtenidos en la red, como por ejemplo *funciones de umbral (funciones máximo, mínimo, rango, etc)*, el uso de la agregación o fusión de datos es ineficiente desde el punto de vista del consumo energético. Ello se debe a que, para el cómputo de este tipo de funciones se necesita obtener solamente algunos pocos argumentos relevantes, pudiendo filtrar los demás datos. En este sentido, si la cantidad de argumentos relevantes m provistos por los nodos sensores al nodo sink es baja, y mucho menor al tamaño de la red ($m \ll n$), o que solo m nodos poseen datos útiles para la aplicación, y ante el desconocimiento de cuales son estos nodos, resulta más eficiente aplicar mecanismos de búsqueda para encontrarlos y dirigir las consultas hacia ellos, haciendo que solo ellos reporten sus datos al sink (Figura 2.5c). No obstante, para garantizar no cometer errores de cómputo por la pérdida de argumentos, se deben implementar esquemas de búsqueda que permitan llegar siempre a nodos con datos relevantes para la aplicación, pero también con capacidad de detectar nuevos datos relevantes entregados por otros nodos de la red. Para ello se utiliza una combinación de heurísticas de búsqueda basadas en gradientes de información, con computación distribuida y estrategias simples de aprendizaje y filtrado de datos y de nodos. Además, para reducir los costos de comunicación, se realizan inicialmente búsquedas en toda la red, y posteriormente se reduce gradualmente el espacio de búsqueda, mientras no se pierdan argumentos útiles para el cómputo. Este es el caso para el cual se desarrollan los protocolos propuestos en esta Tesis.

Hay que tener en cuenta, que los dos primeros casos están orientados a esquemas de reporte

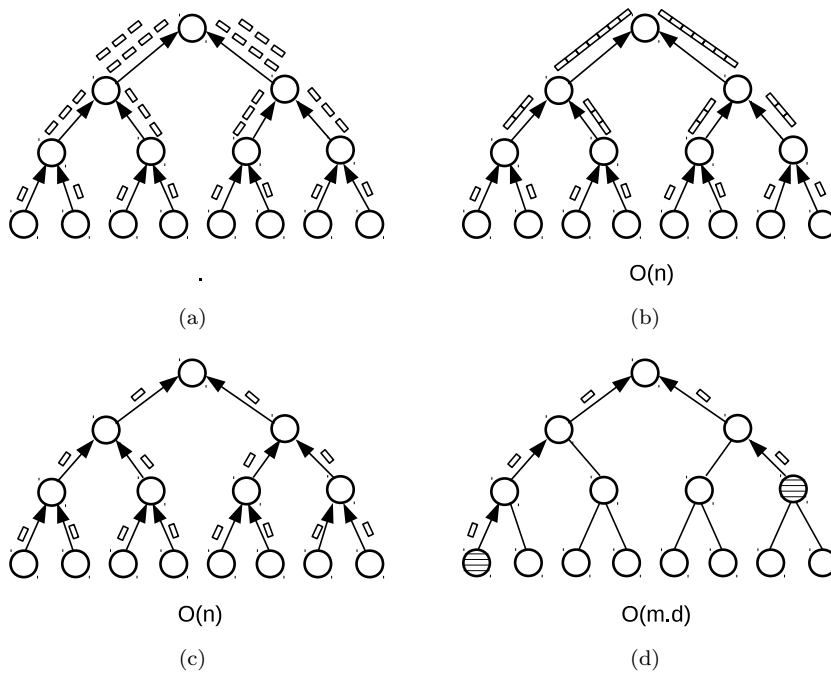


Figura 2.5: Reporte de datos en WSNs: a) sin procesamiento para la reducción de datos, con agregación/fusión de datos b) sin reducción de tamaño, y c) con reducción de tamaño, y d) con filtrado de datos y de nodos.

iniciados por los nodos (esquemas continuos o manejados por tiempo). En cambio, el último caso está más orientado a esquemas de reporte iniciados por requerimiento de un usuario, o basados en consulta/respuesta. Ello se debe a que, para detectar que nodos poseen información útil para la aplicación, hay que primero explorar la red. En este caso, se puede propagar el requerimiento de información a toda la red, y dejar seleccionados que nodos reportarán sus datos al nodo sink.

Como se describió anteriormente, lo que distingue principalmente a las WSNs de los demás tipos de redes de comunicaciones, es la capacidad de sus nodos y la de la red, para procesar información dentro de la misma, mientras los paquetes son dirigidos hacia el nodo sink. En base a esta característica, se puede analizar a una WSN desde dos enfoques diferentes. Por un lado, considerarla como una red inalámbrica con nodos que intercambian archivos, y por otro lado, debido a que las WSNs son diseñadas para obtener información de la red (tal como valores medios, máximos, mínimos, etc.), es verlas como un sistema computacional distribuido, es decir, como una computadora con muchos procesadores de limitadas prestaciones comunicados inalámbricamente (definida como una computadora de Maxwell). Sin embargo, las WSNs presentan una combinación de ambos enfoques, ya que tienen ciertas características similares a las redes de comunicaciones de datos, como así también a las computadoras tradicionales [51].

En general, el procesamiento o computación que puede ser aplicado a los datos que provienen desde diferentes nodos sensores en una WSN, puede ser clasificado en:

- **agregación:** es el proceso de *combinar* datos relacionados de diferentes nodos, usando funciones de agregación como suma, promedio, máximo, mínimo, etc.
- **fusión:** es el proceso de fusionar los datos de difentes nodos, basado en un criterio de decisión,

como votación, consenso, o reglas probabilísticas de decisión Bayesiana. Está relacionada con combinar datos heterogeneos a información más abstracta.

- **compresión:** involucra la reducción de los datos del nodo sensor antes de la transmisión.
- **filtrado:** es el proceso de descartar datos que no son relevantes para la aplicación.

Si bien en muchos trabajos se tiende a utilizar los términos anteriores como sinónimos, debido a que en ciertos casos están solapados, existen diferencias sustanciales entre ellos. Sin embargo, el objetivo de todas estas técnicas es el mismo, el de reducir el intercambio de paquetes, o el tráfico de paquetes o de bytes en la red. Sin embargo, el método utilizado para cumplir con este objetivo es sustancialmente diferente. Tanto la agregación como la fusión de datos involucran *combinación* de datos provenientes de diferentes nodos mientras son enrutados hacia el sink. Además, las tres primeras técnicas tienen como desventaja que introducen cierta demora o delay para obtener la información buscada, teniendo que ser considerado este aspecto en la calidad de servicio de la red (QoS).

Siguiendo con el esquema de basado en pregunta/respuesta, para la computación de una función en una WSN, el usuario de la red establece inicialmente un interés por cierta función estadística de los datos obtenidos por los nodos de la red. Por ejemplo, realizar el computo del valor máximo, mínimo, rango, k-máximos, k-mínimos, media, moda, promedio, histograma, varianza, conteo, suma, o de funciones lógicas como identidad, paridad, indicador, y cualquier tipo de funciones complejas. Sin embargo, las funciones pueden ser clasificadas en general según su tipo en: *divisibles, simétricas, sensibles o no a duplicados, sensibles al tipo, y sensibles a umbral* [50].

En este sentido, el cómputo de *funciones divisibles* presenta una gran ventaja para su aplicación en las WSNs, debido a que el mismo puede ser paralelizado fácilmente. Sin embargo, bajo este tipo de funciones, existen *funciones sensibles y no sensibles a duplicados*. Ejemplo del primer tipo son, funciones media, mediana, histograma, etc. En este caso, los nodos de la red deben utilizar algún mecanismo para evitar que multiples copias del mismo mensaje sean consideradas como argumentos independientes, lo cual afectaría al resultado. Estrategias de este tipo pueden ser, por ejemplo, filtrar mensajes repetidos, en el caso de no utilizar tecnicas de procesamiento en la red como agregación de datos. Otro tipo de estrategia para evitar mensajes duplicados es utilizar envío de datos por unicast, lo cual se verá mas adelante.

Otro tipo de función de interés en las WSNs es el cómputo de *isocontornos*, es decir, obtener información de que nodos poseen los mismos datos, o que estos estén dentro de un rango acotado. Sin embargo, este es un caso particular de computación de la función *rango*. Por otro lado, la obtención de *histogramas* de los datos obtenidos de la red posibilita mediante el post-procesamiento obtener otros tipos de funciones, como por ejemplo, funciones máximo, mínimo, promedio, etc.

Los primeros trabajos propuestos en el área de la computación de funciones en WSNs estaban centrados en el análisis asintótico del número de transmisiones necesarios para computar cada tipo de función. En este sentido, el cómputo de cada función demanda diferentes costos de comunicación. Por ejemplo, el cómputo de funciones tales como promedio, media, histograma, etc, requiere de una gran cantidad de argumentos (valores sensados) y por lo tanto, demandan un elevado costo de comunicación. Por otro lado, el cómputo de funciones tales como rango, conteo, etc, requieren de un menor uso de la comunicación. En general, cualquier tipo de procesamiento puede ser aplicado a los datos obtenidos por los nodos sensores. En este sentido, se puede ver a la red como una caja negra con múltiples entradas,

que representan los datos obtenidos por cada nodo de la red, y una única salida, que representa el resultado que se entrega al usuario de la red. En esta caja negra se puede computar una función determinada de los datos sensados.

Uno de los trabajos precursores en el procesamiento de los datos o de la información en WSNs es el que propuso el *Tiny Aggregation Service (TAG)* [50]. Este servicio, desarrollado para aplicar agregación de datos en WSNs, provee una abstracción de programación de alto nivel genérica y fácilmente invocable por usuarios que no son expertos en redes. Esto es, TAG se enfoca en liberar a las aplicaciones de la idiosincrasia de las capas inferiores de comunicación y de hardware. Para ello se basa en dos atributos esenciales: i) proveer de una interface declarativa simple para la recolección y agregación de datos, inspirada en lenguajes de requerimiento para bases de datos, y ii) distribuir y ejecutar inteligentemente requerimientos para agregación en WSNs de una manera temporal y energéticamente eficiente, sensible a las limitaciones de los recursos y a las propiedades de comunicación con pérdidas presentes en las WSNs. En este sentido, TAG realiza agregación en la red, computando los datos como flujos de datos, descartando datos irrelevantes y combinando lecturas relevantes en paquetes de datos de la forma más compacta posible.

En el desarrollo de esquemas de procesamiento distribuidos en WSNs a lo largo de los años, se ha enfatizado el uso de agregación/fusión de datos bajo un flujo de datos de muchos a uno, y presentado un sin fin de optimizaciones respecto a la reducción de la comunicación, de la latencia para extraer los datos de la red, etc. Sin embargo, no se ha considerado la reducción de los datos transmitidos en el caso de que solo pocos nodos provean datos útiles para la aplicación. El filtrado de datos y de la participación de ciertos nodos en la red para computar una función de los datos comienza a tomar preponderancia a medida que un menor número de nodos posee datos de interés para el usuario, o lo que es equivalente, en el caso de que los datos de interés sea perteneciente solo a un pequeño subgrupo de nodos. En este sentido, un caso extremo sería, por ejemplo, que la información de interés la posea solamente un nodo. En este caso, un esquema de recolección de la información basado en agregación, osea, que todos los nodos reporten sus datos cada un cierto tiempo, y se ejecute agregación en los nodos intermedios, termina siendo muy ineficiente desde el punto de vista del costo de comunicación y de procesamiento. Además, los protocolos de agregación propuestos en la literatura, suponen cierta estabilidad en la topología de los nodos que conforman la red, y hacen uso de estrategias activas de actualización constante del estado de la red, lo cual insume costos extras de energía de comunicación.

A continuación, se introduce a los tipos de funciones analizadas en la bibliografía, y de interés para los protocolos de procesamiento propuestos en esta Tesis. Estas son **funciones divisibles, simétricas, no sensibles a duplicados, y sensibles a umbral**.

2.2.3. Tipo de Funciones Analizadas

Las *funciones simétricas* son el tipo de funciones estadísticas de mayor interés para ser computadas en las WSNs, y sobre las cuales se enfocan los protocolos propuestos en esta Tesis. Ello se debe a dos cuestiones fundamentales: i) estas funciones son *invariantes a la permutación de sus argumentos*, y ii) *la mayoría de las funciones estadísticas pertenecen a este tipo de funciones*.

Se dice que una función es invariante a la permutación de sus argumentos si:

$$f(x) = f(\sigma \hat{x}'). \quad (2.2)$$

donde $\hat{x}' = [\hat{x}'_1(i), \hat{x}'_2(i), \dots, \hat{x}'_n(i)]$ es el vector de mediciones tomado por n nodos en la iteración i , y σ es cualquier permutación de los elementos del vector \hat{x}' . Esta propiedad es muy importante, debido a que en una WSN los paquetes de datos normalmente arriban a un nodo por diferentes caminos y en diferente orden, cumpliéndose la Ecuación 2.2.

Ejemplos comunes de funciones estadísticas simétricas son: *media*, *mediana*, *moda*, *histograma de frecuencias*, *máximo/mínimo*, y *rango*. Sin embargo, la mayoría de las funciones simétricas pueden ser clasificadas en dos subgrupos o subclases disjuntas de funciones [52, 53]:

- *sensibles al tipo*: media, mediana, moda, promedio.
- *sensibles al umbral*: máximo, mínimo, k-máximos, k-mínimos, rango.

A continuación, se define a cada una de estas funciones.

Funciones Sensibles al Tipo

Se dice que una función simétrica $f(\cdot)$ es *sensible al tipo* si existe algún valor γ entre $0 < \gamma < 1$ y un entero \bar{n} , tal que $n \geq \bar{n}$ y cualquier $j \leq n - \lceil \gamma n \rceil$, dado cualquier subconjunto $\{x_1, x_2, \dots, x_j\}$, hay dos subconjuntos de valores $\{y_{i+1}, y_{i+2}, \dots, y_n\}$ y $\{z_{j+1}, z_{j+2}, \dots, z_n\}$, tal que:

$$f(x_1, \dots, x_j, z_{j+1}, \dots, z_n) \neq f(x_1, \dots, x_j, y_{j+1}, \dots, y_n)$$

Si la desigualdad anterior es cierta para $j \leq n - \lceil \gamma n \rceil$, esta es automáticamente cierta para todos los valores de j .

Funciones Sensibles al Umbral

Se dice que una función simétrica $f(\cdot)$ es *sensible al umbral* si existe un vector no negativo $\theta|X|$, llamado vector de umbral, tal que $f(\underline{x}) = f'(\tau(\underline{x})) = f'(\min(\tau(\underline{x}), \theta))$, para todos los $\underline{x} \in X^n$, con *min* significando el mínimo elemento por elemento.

Intuitivamente, una función sensible al tipo no puede ser determinada si una gran cantidad de argumentos es desconocida, mientras que una función sensible al umbral puede ser determinada por un número fijo y reducido de argumentos conocidos. **En esta Tesis nos centramos en protocolos de computación de funciones de tipo umbral, ya que se puede reducir en gran medida el costo de comunicación, y que además incluye a las funciones de mayor interés para el usuario de una WSN.**

En este contexto, Giridhar et al. han propuesto modelos matemáticos de computación de datos en red [51]. Estos trabajos tienen como objetivo el desarrollo de estrategias de computación y comunicación. Los autores proponen y analizan diferentes modelos de computación. El primero de ellos se basa en el concepto de que los nodos poseen porciones de información, y del intercambio de mensajes entre los nodos para computar una función determinada. El segundo modelo analiza el concepto de reuso espacial en redes inalámbricas, con la posibilidad de acelerar la eficiencia computacional mediante el uso de computación de bloque, considerando la idea de modelado geográfico de WSNs tanto como

redes yuxtapuestas como de redes inalámbricas de múltiples saltos aleatorias. Además, se presenta un estudio de las clases funciones con respecto a cuán eficientemente pueden ser computadas en tiempo. Finalmente, se estudia un tercer modelo que considera un escenario más realista, es decir, el hecho de que las comunicaciones inalámbricas son en sí mismas ruidosas, y la noción de computar una respuesta que es correcta con un cierto grado de probabilidad. Todo el análisis desarrollado en este trabajo se enmarca dentro del *modelo de complejidad computacional*. Sin embargo, todo el análisis en este trabajo se enfoca en cómo obtener la tasa máxima de computación y comunicación.

En las WSNs, a diferencia de otro tipo de redes, y como se describió anteriormente, los nodos no solo tienen la capacidad de retransmitir paquetes en la red, sino también la de procesar en tránsito los paquetes de datos. En este sentido, los datos pueden ser agregados, filtrados (descartados), etc, con el fin de disminuir la cantidad de paquetes que viajan a través de la red, ahorrando energía en los nodos por medio de la reducción de los costos de comunicación [3]. Mientras que la agregación de datos, que ayuda a fusionar datos redundantes, ha sido extensamente estudiado [30], el filtrado de datos ha recibido mucha menos atención por parte de la comunidad científica. A continuación se tratarán los conceptos que describen al filtrado en red.

2.2.4. Funciones de Umbral

El cómputo de funciones de umbral ha sido muy estudiado en el caso de redes yuxtapuestas (collocated networks). Sin embargo, no ha sucedido lo mismo para el caso de redes aleatorias con múltiples saltos [52]. Este tipo de funciones pueden ser divididas en dos categorías, basado en cómo es definido el umbral. Estas pueden ser:

- *Funciones de umbral fijo*: definen un umbral que es conocido a priori y que es embebido en el requerimiento *como un valor constante*. Los nodos sólo leen este umbral para determinar si sus lecturas son relevantes o no para el cómputo de la función. Funciones como *rango* e *iso-contorno* son buenos ejemplos de esta categoría.
- *Funciones de umbral dinámico*: definen un umbral que es embebido en el requerimiento *como una variable*. Los nodos pueden, no sólo leer este umbral, sino también modificarlo. Entre las funciones que pertenecen a esta categoría, la función *máximo* es el ejemplo más interesante, ya que puede ser usada para implementar otras funciones tales como *mínimo*, *k-máximos*, *k-mínimos*, etc.

El cómputo de funciones de umbral dinámico no es trivial, y requiere de esquemas de filtrado que se adapten y aprendan de la distribución de datos en la red. Es sobre este tipo de problema en el que se enfoca esta Tesis.

2.3. Modelos de Enrutamiento

El enrutamiento de paquetes de datos es una de las tareas más importantes que tienen los nodos que conforman una red de datos. Esta consiste en el *proceso de dirigir o enrutar paquetes de datos, desde un nodo origen hacia un nodo destino, a través de múltiples saltos, y por un único o por múltiples*

*camino*s. La decisión de a través de que ruta o nodo vecino enviar un paquete, puede ser tomada individualmente por cada nodo, o impuesta externamente por un nodo coordinador. El trabajo en conjunto de todos los nodos de la red permite cumplir con la tarea de enrutamiento de los datos.

En este sentido, cada nodo posee su propio conocimiento local de la red, obtenido en base a su experiencia, o impuesto externamente, y almacenado en tablas contenidas en memoria. Estas contienen información actualizada acerca de sus nodos vecinos, y a través de que cual de ellos conviene enviar un paquete de datos para que este llegue a un destino determinado. Sin embargo, debido a la dinámica de las redes de comunicaciones, y en particular de las WSNs, en lo que respecta a cambios de topología (movilidad de nodos, falla de nodos, inclusión de más nodos, etc), se deben actualizar dichas tablas con el estado actual de la red cada cierta cantidad de tiempo. Esta actualización puede ser realizada por el mismo nodo, o impuesta externamente.

Para analizar el enrutamiento de paquetes dentro de una red, es indispensable contar con un *direccionamiento adecuado de los nodos*. En este sentido, el uso de direccionamiento basado en direcciones IP no es una buena solución para ser aplicada en WSNs, dado el elevado costo que introduce su uso, debido a la extensión de las mismas. Por ejemplo, en este caso, cada paquete lleva información de dirección de origen y de destino del paquete, con un tamaño de 4 Bytes o de 8 Bytes para cada dirección IPv4 o IPv6, respectivamente. La alternativa más utilizada en WSNs para direccionamiento es el uso de la dirección MAC de los nodos (o de parte de ella), la cual es una dirección única dada por el fabricante del transceptor de radio. Sin embargo, se pueden definir dos enfoques diferentes respecto al *direccionamiento de los nodos*, y a su *capacidad de enrutar paquetes*, los cuales están basados en las características del hardware de los nodos.

- *Nodos con recursos muy limitados*: no utilizan protocolos de enrutamiento ni información de rutas, debido a la demanda de procesamiento, memoria, etc, que esto requiere. En su lugar, estos nodos utilizan protocolos simples de disseminación o de retransmisión de datos, como por ejemplo inundación controlada (controlled flooding), búsqueda por expansión por anillos (expanding ring search, ERS), o basados en decisiones probabilísticas, como por ejemplo caminata aleatoria (random walk), o rumores (gossip). La *retransmisión de un paquete* no implica necesariamente el enrutamiento del mismo, sino que debe existir cierta información extra en un nodo para determinar a que nodo vecino enviar dicho paquete, para que luego de múltiples retransmisiones el mismo llegue a su destino. Sin embargo, se han propuesto soluciones innovadoras para optimizar tanto la búsqueda como el reporte de los datos en WSNs conformadas por nodos con recursos limitados, los cuales serán descritos posteriormente.

- *Nodos con mayores capacidades*: utilizan protocolos de enrutamiento tradicionales, por ejemplo, basados en direcciones IP, como los utilizados en las redes de área local (local area networks, LANs). Es decir, utilizaban técnicas de enrutamiento basadas en direcciones, conocidas como *address-centric* o *address-based* [10]. En este caso, cada nodo posee una identificación *id* única, y una *tabla de enrutamiento* en memoria, con información de nodos vecinos, y de través de que nodo vecino se puede llegar con un paquete a un nodo de destino distante. Cada nodo intermedio utiliza su información local para encontrar el mejor camino hacia este destino bajo un cierto criterio. En este contexto, criterios comunmente utilizados en redes de datos pueden ser: enrutar paquetes por el camino más corto, el de menor tráfico, el de mayor capacidad, el de menor latencia, etc. Cada nodo posee su propio conocimiento local de la red. Sin embargo, como se describió anteriormente, debido a la dinámica de las redes de comunicaciones en lo que respecta a cambios de topología (movilidad de nodos, falla de nodos, inclusión de más nodos, etc), se deben actualizar dichas tablas cada cierta cantidad de tiempo

con el estado actual de la red. Para este fin, se pueden utilizar *esquemas proactivos*, en los que cada nodo inyecta mensajes cada cierto tiempo para obtener información actualizada de la red y actualizar su tabla, lo cual demanda elevados costos de comunicación, o *esquemas reactivos*, en los cuales se inicia este proceso de actualización cuando un nodo quiere reportar datos.

En resumen, los protocolos de enrutamiento de paquetes basados en direcciones IP son eficientes en aplicaciones en las que los nodos disponen de recursos de procesamiento y memoria adecuados, y en las que cada nodo de la red transmite datos a cualquier otro nodo de la red (comunicación uno-a-uno), como en el caso de las LANs. En cambio, en el caso de las WSNs, se tiene generalmente un flujo de comunicación de uno-a-muchos (en el envío de paquetes de requerimiento o de actualización desde el nodo sink a los nodos sensores), o de muchos-a-uno (en el reporte de datos hacia el sink).

Con el avance en la investigación y desarrollo en el área de las WSNs, se planteó alrededor del año 2000, que *era necesario un cambio de paradigma respecto a los protocolos de comunicación*, debido a que estas redes están orientadas específicamente a aplicaciones de búsqueda y recolección de datos o de información, y no a comunicación punto a punto con en otro tipo de redes de comunicaciones. Además, se determinó que *el valor está en los datos sensados y en la información que los nodos proveen al usuario, y no en la identificación de quienes generan estos datos*. Bajo este enfoque, se pudo considerar al enrutamiento de paquetes basado gradientes de información, y al procesamiento de datos dentro de la red, a medida que estos son retransmitidos de un nodo a otro, con lo cual se puede reducir el uso de la comunicación. De este modo, se comprobó que es más eficiente en estas redes aplicar estrategias de enrutamiento basadas en datos, conocidas como *data-centric o data-based*. Bajo este nuevo paradigma, cada nodo que retransmite un paquete en la red puede acceder a la información o dato que el mismo transporta, y llegar a modificarlo bajo un cierto criterio.

El primer protocolo que propuso el *paradigma de enrutamiento centrado en datos* para las WSNs, como alternativa al enrutamiento centrado en direcciones, fué *Directed Diffusion (DD)* [8, 54, 55]. Bajo este paradigma, se propuso que los datos pueden ser almacenados en los nodos, y procesados en forma distribuida en la red ante el interés de un usuario. Una característica de DD es que la diseminación de paquetes de interés, y el reporte y agregación de los datos, son determinados por interacciones localizadas resultantes del intercambio de paquetes entre nodos vecinos. El foco de DD está centrado en el diseño de mecanismos de diseminación para tareas de detección de eventos, que sean escalables, robustos a fallas en nodos, y que minimizen el uso de energía. DD es un esquema basado en requerimientos de información, y utiliza gradientes, resultantes de la transmisión de los paquetes, para reforzar caminos de respuesta desde aquellos nodos que detectaron un evento definido en el mensaje de interés hacia el nodo sink. Además, en el mensaje de interés se hace una descripción de la tarea, es decir, se especifica el tipo de evento que se quiere detectar, la localización de los nodos a interrogar, y el tiempo de expiración del requerimiento. Para asegurar que el mensaje de requerimiento o interés sea transmitido a toda la red en forma confiable, el nodo sink reenvía el mismo interés a la red con una marca de tiempo (o timestamp) monótonicamente creciente. Se define también el concepto de formación de gradientes de interés en la red, y la capacidad de un nodo para decidir si retransmite o no el interés a un grupo de nodos vecinos. En este sentido, se pueden definir tres alternativas: i) lo más simple es transmitir el interés a todos los nodos, siendo equivalente a inundar la red, en caso de ausencia de información acerca de que nodos son probables candidatos a satisfacer dicho interés, ii) realizar enrutamiento geográfico en caso de conocer la posición exacta de los nodos, iii) en caso de una red estática, un nodo puede utilizar datos en memoria, basado en escucha activa de respuestas

previas de nodos vecinos, para dirigir los intereses.

En resumen, la propagación de interés define estados en la red o en parte de la misma, para facilitar la posterior extracción de datos hacia el nodo sink. Las reglas de propagación de interés son locales a los nodos, y pueden ser diferentes para diferentes tipos de tareas, como por ejemplo, obtener la localización estimada de un evento en la red, o la cantidad de eventos que se producen en un período de tiempo determinado.

Sin embargo, DD no es un esquema basado en *pregunta/respuesta*, sino que el interés, incluyendo una notificación de evento de bajo rate, es propagado en la red, y cada nodo almacena este requerimiento. Solo en casos de que se produzca un evento, cada nodo que lo detecta analiza si tiene un interés predefinido por ese tipo de información en una tabla local, y si es así, comienza el proceso de reporte de datos hacia el nodo sink. En este sentido, el nodo envía sus datos a cada nodo vecino por el cual recibió el interés. Basado en una estrategia de reducción en las demoras de las respuestas o incremento de la tasa de transferencia de datos, se refuerzan ciertos caminos entre el nodo que detecta el evento y el sink. Sin embargo, en caso de que se refuercen varios caminos, el nodo sink aplica una estrategia de refuerzo negativo para limitar el envío de datos por caminos más desfavorables o de mayor latencia. Sin embargo, nodos intermedios también pueden implementar reglas de refuerzo. Esto es útil para posibilitar reparaciones locales de caminos con fallas o degradados, producidos como consecuencia de fallas en nodos, y factores externos que afectan a la comunicación (obstáculos, fading, etc).

Por otro lado, Krishnamachari et al. [56], han modelado y simulado ambos tipos de paradigmas, los basados en direcciones (address-centric, AC) y los basados en datos (data-centric, DC), y han comparado su performance en aplicaciones de WSNs. Además, se enfatiza que la *agregación de datos puede ser utilizada en los esquemas DC como un mecanismo muy útil para reducir la comunicación de paquetes de datos*. La idea es combinar datos que provienen de diferentes nodos y que tienen cierta redundancia, con el fin de minimizar el número de transmisiones y ahorrar energía. Otro de los aportes de este trabajo es entender el contexto en el cual la agregación de datos es útil. En este sentido, en los esquemas AC los datos son generalmente enrutados por el camino más corto, en cambio, en los esquemas DC, estos son enrutados por el camino que permite obtener una mayor agregación de los datos. Para comprender mejor este segundo enfoque, se analizan 3 tipos de escenarios con diferentes niveles de correlación o redundancia en los datos, desde un extremo sin redundancia (todos los datos diferentes), un caso intermedio con cierta redundancia, hasta uno con redundancia completa. Adicionalmente, los autores determinaron que en un gran rango de escenarios, los esquemas de enrutamiento DC ofrecen una ganancia de performance significativa respecto a los AC, debido fundamentalmente a la posibilidad de utilizar agregación de datos. Sin embargo, se concluye con que esta ganancia se da solo en aquellos casos en que se tienen una gran cantidad de fuentes de datos en la red, y que la inclusión de la misma introduce una latencia significativa.

La idea de introducir una *medida de utilidad de información para guiar dinámicamente el enrutamiento de paquetes* fué propuesta por Chu et al. [57]. Esto permite maximizar la ganancia de la información mientras se minimiza la latencia de detección y el consumo de ancho de banda para tareas como localización y seguimiento. Los autores de este trabajo proponen dos técnicas denominadas *information-driven sensor querying (IDSQ)* y *constrained anisotropic diffusion routing (CADR)* para requerimiento y enrutamiento de paquetes en forma eficiente energéticamente en WSNs con tareas de procesamiento colaborativas. Estas técnicas surgen de la necesidad de desarrollar nuevos protocolos de procesamiento y de enrutamiento colaborativos para manejar los datos recolectados por la red, y

procesar la información para la tarea dada. IDSQ es desarrollada para optimizar la selección de sensores, en tanto que CADR es desarrollada para dirigir el enrutamiento de paquetes e incrementalmente combinar mediciones de sensores para minimizar el costo total. Se plantea como estrategia para el caso de WSNs de gran escala, el manejo selectivo de las tareas asignadas a los nodos. Para ello se utilizan medidas de utilidad de la información y una actualización incremental de la creencia de cada nodo. Cada nodo puede evaluar un objetivo información/costo, tomar una decisión, actualizar su estado de creencia, y enrutar los datos basado en el gradiente información local/costo y el requerimiento del usuario final de los datos.

Afortunadamente, en los últimos años, la comunidad científica relacionada a las WSNs, ha visto con gran interés el *uso de gradientes de información como estrategia para el enrutamiento de paquetes a zonas de la red con mayor información* [35, 36, 58, 59]. En este sentido, no existe un único criterio de cómo definir y utilizar estos gradientes, sino que esto depende de la aplicación particular. Estos gradientes pueden ser *naturales* o *artificiales* desde el punto de vista de los datos sensados. Un ejemplo del primer caso [35, 36] considera el problema de cómo direccionar requerimientos en aplicaciones de detección de eventos. Para ello se utiliza el gradiente de los datos sensados para direccionar la búsqueda. Un ejemplo del segundo caso [59] considera el caso de cómo los paquetes reportados por un nodo pueden encontrar rápidamente algunos de los nodos sink existentes en la red. Para ello se considera que cada paquete tiene asociada una temperatura, y que estos paquetes son retransmitidos en forma direccionada siguiendo gradientes de temperatura hacia alguno de los sinks, posiblemente el más cercano, considerados como fuentes de calor. En este contexto, el tipo de dato que pueden ser utilizado por cada nodo para el enrutamiento basado en gradientes, puede ser clasificado entre:

- *dato del estado del nodo*, por ejemplo, su energía remanente, su distancia al sink, etc.
- *dato sensado que es de la aplicación*, por ejemplo, valor de temperatura obtenido en una aplicación definida para encontrar la temperatura máxima sensada en la red.
- *dato sensado que no es de la aplicación*, por ejemplo, el valor de la presión atmosférica obtenida por cada nodo sensor en una red desplegada en una ladera de una montaña, y cuyo nodo sink se encuentra en la base de la misma, para sensar algún parámetro de interés diferente a la presión [60].

Hasta la actualidad, se han propuesto una gran cantidad de protocolos de enrutamiento para WSNs. Sin embargo, cada uno de ellos es más adecuado para un tipo de aplicación concreta. Existen varios trabajos que clasifican el gran abanico de protocolos de enrutamiento en función de ciertas características importantes, como por ejemplo [22, 61]:

- *Centrado en datos o en direcciones*: normalmente el primero es el más utilizado en WSNs.
- *Jerárquico o plano*: ciertos nodos pueden tener mayores funcionalidades que otros, o todos los nodos tener las mismas.
- *Por camino único o por múltiples caminos*: relacionado a como son direccionados los paquetes entre los nodos.
- *Basado o no en localización*: muchos protocolos requieren de información de la localización de los nodos, la cual es utilizada para calcular la distancia entre nodos.

- *Con calidad de servicio (QoS) o no:* relacionado con la demora introducida por el protocolo de enrutamiento.
- *Envío de datos basado en requerimientos, en tiempo, o en eventos:* descritos en la Sección 2.3.1.
- *Posibilidad de agregar datos o no.*

2.3.1. Modelos de Reporte de Datos

El *reporte de datos* es una tarea enmarcada dentro del enrutamiento de paquetes, y es realizada de forma conjunta por la totalidad o por parte de los nodos que conforman la red. En general, se definen cuatro modelos de reporte de datos, cada uno mas adecuado para cada tipo de aplicación [22, 61]. Estos modelos pueden ser:

- *basados en eventos (event-driven)*
- *basados en tiempo (time-driven, continuous, periodic)*
- *basados en requerimientos o bajo demanda (query-drive, on-demand)*
- *híbridos*

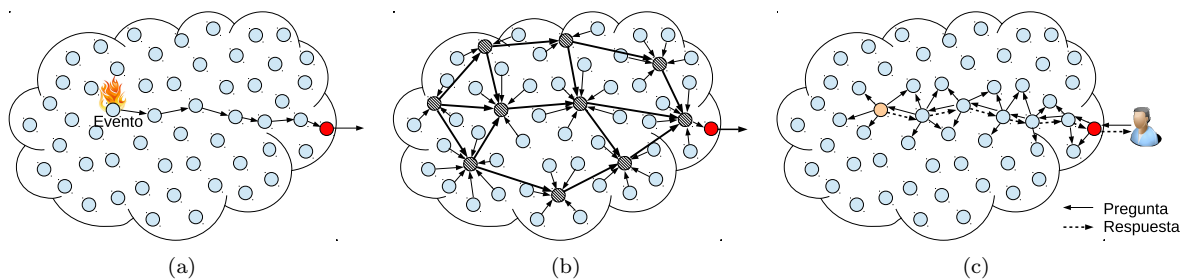


Figura 2.6: Modelos de reporte de datos: basados en eventos (a), en tiempo (b), y en requerimientos (c).

A continuación se describe cada uno de ellos con mayor detalle, con sus ventajas y desventajas, y se citan ejemplos de aplicación.

2.3.1.1. Modelo basado en eventos

En este modelo, los nodos reportan los datos al sink cuando detectan un evento, basado por ejemplo en un valor sensado que supera un cierto umbral pre-definido (por ejemplo la detección de fuego basado en temperatura). Este modelo es adecuado para la detección de eventos (explosiones, incendios, etc), localización, y seguimiento de objetos móviles (vehículos, animales, personas, etc) sobre un área específica. Cuando un evento es detectado en la red, los nodos sensores próximos al evento generan datos y los envían al nodo sink. Una de las características de este modelo es su requerimiento de tiempo-real, por lo que uno de los objetivos es minimizar la latencia de envío de datos.

2.3.1.2. Modelo basado en tiempo

En este modelo, los nodos recolectan datos del entorno a intervalos de tiempo periódicos predefinidos, y los envían al nodo sink. En el modelo anterior, el envío de datos es usualmente iniciado por aquellos nodos sensores próximos al evento detectado, en cambio en este modelo, utilizado mayormente para monitoreo de magnitudes físicas continuas (como humedad y temperatura ambiente), los datos son enviados al sink por todos o por una gran parte de los nodos que conforman la red. En este caso, la agregación de datos durante el reporte de los mismos hacia el nodo sink es clave, lo que permite reducir la cantidad de paquetes transmitidos, y por lo tanto el uso de energía. Ello se basa en que los datos recolectados por los nodos sensores pueden ser redundantes, estar correlacionados, y/o ser inconsistentes.

2.3.1.3. Modelo basado en requerimientos

En este modelo, basado en un *esquema de pregunta/respuesta*, el usuario decide cuando quiere obtener datos de la red. Para ello, el mismo inyecta una solicitud o requerimiento a la red, a través del nodo sink, indicando su interés por cierto tipo de datos o información provista por los nodos, y luego espera a que la red se los reporte. Este modelo es más adecuado para calcular funciones estadísticas de los datos obtenidos por los nodos de una magnitud física en las que no se requiere conocer todos los valores. Los nodos sensores pueden enviar datos relevantes hacia el sink, para computar la función deseada de manera centralizada, o pueden computar la función en forma distribuida y colaborativa mientras los datos son reportados hacia el nodo sink. Como las WSNs son redes poco fiables, debido a que los nodos son propensos a fallar, nodos pueden ser agregados en cualquier momento, nodos pueden cambiar sus posiciones geográficas, etc, este modelo puede ser más eficiente que los anteriores, debido a que en cada diseminación del interés se muestrea el estado de la red.

2.3.1.4. Modelo híbrido

En este último caso, se utilizan combinaciones de los modelos anteriores para optimizar alguna métrica, como por ejemplo, reducir el tiempo hasta que el que el usuario obtiene los datos, o disminuir el costo de comunicación. Un ejemplo de modelo híbrido es *Rumor Routing* [62], el cual es un mecanismo basado en agentes que combina un modelo basado en requerimientos con uno basado en eventos. El objetivo de este mecanismo es enrutar los requerimientos hacia aquellos nodos que han detectado un evento particular e iniciado la notificación del mismo. En este sentido, los autores buscan un compromiso lógico entre enrutar requerimientos y enrutar notificaciones de eventos.

2.4. Modelos de Energía

El modelo del consumo de energía en un nodo sensor puede ser tan preciso y complejo como se quiera, considerado todos los factores en juego (sensado, procesamiento, comunicación, etc). En general, se considera que cada nodo tiene inicialmente un nivel finito de energía E_T en Joules, determinado por el tipo de batería utilizado, y que este nivel se va consumiendo en el sensado (E_S), en el procesamiento (E_P), y en la comunicación, considerando tanto la transmisión (E_{TX}) como en la recepción

de datos (E_{RX}), como se indica en la Ecuación 2.3.

$$E_T = E_S + E_P + E_{TX} + E_{RX} \quad (2.3)$$

Sin embargo, en la literatura, en general, se considera solo a la comunicación inalámbrica (transmisión y recepción de datos) como fuente de consumo de energía, debido a que, en general, la misma es de varios órdenes de magnitud mayor al resto, y se considera que ambos procesos de comunicación demandan la misma cantidad de energía ($E_{TX} = E_{RX} \gg E_S, E_P$).

$$E \approx E_{TX} + E_{RX} \quad (2.4)$$

Considerando que no existe una fuente de carga de energía en los nodos, cuando el nivel de energía remanente (E_R) en el mismo es igual a cero, luego de q_t transmisiones y q_r recepciones de paquetes de datos, el nodo se queda sin energía, pasando a estar fuera de servicio.

$$E_R = E_T - (q_t * E_{TX}) + (q_r * E_{RX}) \quad (2.5)$$

A lo largo de esta Tesis, se considera este modelo de consumo de energía, y se analiza tanto el consumo de energía en la red como el tiempo de vida de la misma, para analizar y comparar la performance de los protocolos propuestos.

2.5. Modelos de Conectividad y Comunicación

En los sistemas de comunicaciones inalámbricos se considera que *existe conectividad o comunicación entre dos nodos*, es decir, que un nodo i puede recibir un paquete transmitido por otro nodo j , si la distancia euclídea o distancia geográfica entre ambos nodos $d_{i,j}$ es menor a su rango de comunicación r , osea, si se cumple $d(i, j) < r$. Esto es sólo una simplificación, dado que en realidad, existen efectos de frontera que pueden producir la pérdida de paquetes bajo una cierta probabilidad. Debido a la necesidad de los nodos sensores de conformar redes, se utilizan en general sistemas de comunicaciones omnidireccionales. Es decir, que se considera el uso de antenas que permitan establecer radio enlaces con nodos desde cualquier dirección, mientras se cumpla la condición de que el nodo receptor se encuentre en el rango de comunicación del nodo transmisor.

En general, existen 3 modos de comunicación de paquetes de datos entre nodos: *broadcast*, *multicast*, y *unicast*. En el primero, el paquete transmitido por un nodo es recibido por todos los nodos que se encuentran dentro de su rango de comunicación r , denominados nodos vecinos (Figura 2.7a). En el caso de multicast, si bien la transmisión del paquete es detectado por todos los nodos vecinos del nodo emisor, al igual que en el caso anterior, solo algunos nodos lo pueden recibir, y los demás nodos lo eliminan (Figura 2.7b). La información de a qué nodos va dirigido el paquete está contenida en el encabezado del paquete. Finalmente, la comunicación unicast considera que el paquete es solo dirigido a un solo nodo vecino (Figura 2.7c). Los dos últimos modos no aprovechan la naturaleza de broadcast de las comunicaciones inalámbricas, pero son más eficientes energéticamente desde el punto de vista

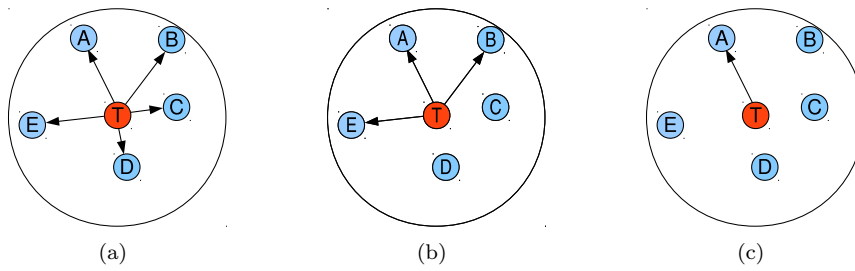


Figura 2.7: Modos de comunicación: broadcast (a), multicast (b), y unicast (c).

del costo de comunicación. Sin embargo, el modo broadcast es el más adecuado para aplicaciones de búsqueda de datos en redes de comunicaciones.

En los sistemas de comunicaciones inalámbricos tradicionales se utiliza el concepto de comunicación punto a punto, es decir, que se realiza el cálculo de radioenlace que posibilite una comunicación fiable entre ambos puntos de comunicación. Cuanto mas grande es el radioenlace, mayor nivel de señal es requerido en el transmisor. Sin embargo, en el caso de las WSNs, el concepto de comunicación es totalmente diferente. En las WSNs, al igual que en las redes Mesh, se utiliza la estrategia de incrementar la cantidad de dispositivos en la red, los cuales pueden retransmitir datos, con el fin de reducir los rangos de comunicación, y de este modo, el consumo de energía. En este sentido, el consumo de energía se incrementa exponencialmente a medida que se duplica el rango de comunicación. Por tal motivo, sistemas de comunicaciones que optimizan los recursos energéticos de los nodos reemplazan el uso de elevados rangos y potencias de transmisión, por esquemas de rangos cortos de comunicación basados en retransmisión por múltiples saltos, como se muestra en la Figura 2.8. En esta figura, se indica con P_{send} la potencia de transmisión, r es el rango de comunicación, N la cantidad de nodos que transmiten y retransmiten el paquete, y $P_{receive}$ el nivel de potencia o señal recibida en el receptor. Sistemas de comunicaciones que siguen esta filosofía son los que se han tenido un mayor desarrollo en los últimos años, dado que actualmente se busca contar con dispositivos móviles conectados en red que maximizen el uso energía.

La *conectividad* es una propiedad de una red de comunicaciones que define como están interconectados sus nodos. Su magnitud puede variar entre 0, para una red en la que ningún nodo se puede comunicar con otro, y 1, para el caso en que todos los nodos estén conectados a algún nodo de la red. En este sentido, se define el rango de comunicación r para asegurar una conectividad de la red cercana al 99%, para una red con nodos desplegados aleatoriamente. Dicha conectividad define la probabilidad de que cada nodo esté conectado a por lo menos un nodo vecino en la red. En este sentido, muchos trabajos se han enfocado en determinar cual es el número mínimo de nodos vecinos (o grado mínimo) que tiene que tener cada nodo para asegurar que la red este totalmente conectada. Giridhar et. al, basados en análisis matemáticos, concluyeron que una red con n nodos distribuidos uniformemente en un área puede ser considerada como asintóticamente conectada con probabilidad que tiende a 1, si cada nodo está conectado a más de $5,1774 \log(n)$ nodos vecinos [63], como se indica en la Figura 2.9.

De la Figura 2.9 se puede observar que para asegurar con alta probabilidad de que una red de 100 nodos esté totalmente conectada, se requiere de que cada nodo tenga más de 10 nodos vecinos. Sin embargo, para una red de 800 nodos se requiere de más de 15 nodos vecinos. Por ejemplo, si cada nodo tiene 16 nodos vecinos se cumple con la condición de conectividad para redes de hasta 1000 nodos.

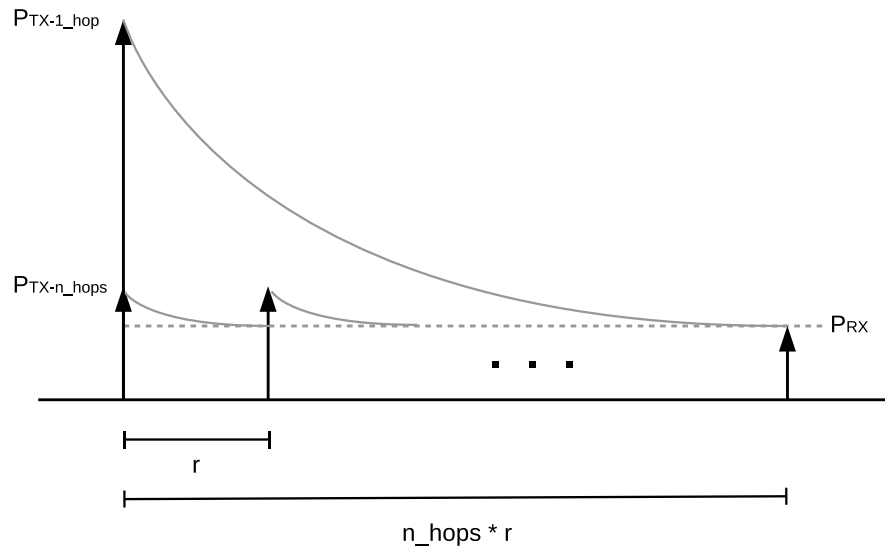


Figura 2.8: Comparación entre esquemas de transmisión con uno y con múltiples saltos.

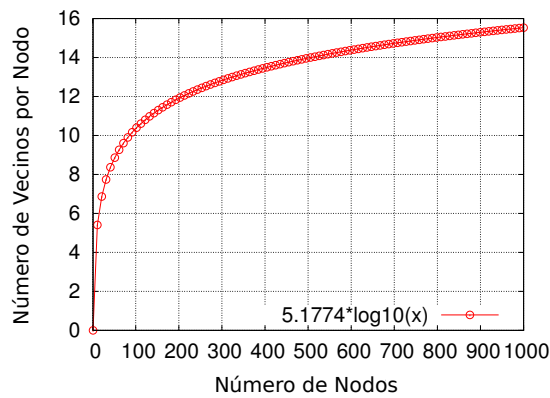


Figura 2.9: Cantidad mínima de vecinos (N_{min}) necesaria para asegurar la conectividad de la red.

2.6. Comentarios de Cierre

En este capítulo se describieron los modelos mas comunmente utilizados en la literatura para representar tanto a las magnitudes físicas a monitorear, al funcionamiento de los nodos sensores, como a la interrelación entre los nodos de la red. Los modelos fueron descriptos siguiendo una secuencia desde capas superiores hasta las capas más bajas de comunicación inalámbrica entre nodos. Como conclusión, se puede observar que el modelo mas adecuado es totalmente dependiente del caso de uso particular, o de manera equivalente, es función del tipo de aplicación en la que se va a utilizar la red.

Finalmente, los modelos seleccionados para el desarrollo de los protocolos de procesamiento y filtrado en red propuestos en esta Tesis son los siguientes. Para el modelado de la magnitud física a sensar, se utilizan modelos de distribución de datos conformados por la sumatoria de funciones que representan fuentes de datos. En este contexto, cada nodo sensor obtiene su lectura muestreando dicha distribución en su localización. Para el modelado del funcionamiento de los nodos, se utiliza un modelo de procesamiento distribuido basado en búsqueda de información y en filtrado de datos y de nodos, para el cómputo de funciones divisibles, invariantes a la permutación de los datos, no sensibles

a duplicados, y de umbral fijo o dinámico, como por ejemplo funciones máximo, mínimo, rango, etc. El modelo de enrutamiento es centrado en datos, basado en gradientes de información, con un modelo de reporte de datos basado en requerimientos o bajo demanda del usuario de la red. El modelo de energía utilizado es el basado en las Ecuaciones 2.4 y 2.5, y el modelo de comunicación adoptado, es el basado en broadcast.

Capítulo 3

Procesamiento Local

Contenido

3.1	Introducción	51
3.2	Metaheurísticas	54
3.2.1	Tipos de Metaheurísticas	54
3.3	Aprendizaje por Refuerzo	56
3.4	Elección de Estrategia de Operación	57
3.5	Esquema de Procesamiento Propuesto	58
3.6	Protocolos de Filtrado en Red	60
3.7	Estrategias de Procesamiento	61
3.8	Procedimientos	61
3.9	Arquitectura	62
3.9.1	Reglas de Decisión y Filtrado	63
3.10	Comentarios de Cierre	64

3.1. Introducción

En este capítulo se describe todo lo desarrollado en esta Tesis, es decir, la arquitectura implementada en cada nodo para el procesamiento y filtrado local de datos, los procedimientos requeridos para el cómputo de funciones de umbral, y la introducción a los protocolos de filtrado de nodos vecinos que serán descritos en detalle en los Capítulos 4 y 5.

En la mayoría de las aplicaciones en que se utilizan las WSNs, se requiere computar alguna función estadística de los datos obtenidos por los nodos, como por ejemplo, *funciones máximo, mínimo, rango, etc.* Sin embargo, y en general, el usuario no está interesado en recibir los datos de todos los nodos, sino más bien, en algún tipo de información. Es por ello, que los datos deben ser procesados, ya sea de forma centralizada o distribuida, antes de ser entregados al usuario, ya que los datos por sí

solos no aportan información útil al usuario. En este contexto, el procesamiento de los datos se puede formalizar como computar una función $f(X)$, donde X es el conjunto de valores sensados desde x_1 hasta x_n , siendo n el número de nodos de la red.

En general, todos los valores x pueden ser reportados al nodo sink, para computar la función *de manera centralizada*. Este esquema ha sido el utilizado en las primeras implementaciones de procesamiento de datos en WSNs. Sin embargo, estas eran redes de pequeña escala en las que el nodo sink tenía mayores capacidades de procesamiento que la de los nodos sensores (los cuales eran más económicos y limitados). Bajo este esquema de computación centralizada, si el nodo sink recibe todos los paquetes de datos, no se cometen errores de cómputo, a expensas de un elevado costo de comunicación y de energía debido al envío de datos. Este costo de comunicación se incrementa en gran medida cuando aumenta el tamaño de la red.

Con la evolución de la tecnología, en los últimos años se logró un incremento sustancial de las capacidades de computación de los nodos sensores, lo que posibilitó realizar parte de esta tarea *de forma distribuida en la red*, descentralizando el procesamiento en el nodo sink. Este enfoque encuentra una analogía con los procesadores multi-cores, o con los GPUs (Graphical Processing Units), en el que los nodos de una WSN pueden ser considerados como procesadores comunicados entre sí por enlaces inalámbricos, denominada por ciertos autores como *computadora marxweliana* [51]. Esto se basa en el hecho de que en las WSNs, es mucho más eficiente energéticamente, desde el punto de vista del costo de comunicación, realizar computación distribuida en la red, que hacerlo en forma centralizada en el sink. Como se describió en la Sección 2.4, el costo de comunicación en una WSN es superior al costo de computación en varios órdenes de magnitud. Debido a esto, se tiene como principal objetivo, disminuir los comunicación de paquetes en la red, incluyendo tareas de procesamiento y filtrado de datos en los nodos. Sin embargo, como los nodos utilizan microcontroladores de bajo consumo, por lo general con arquitecturas de 8 bits, estos no tienen grandes capacidades de procesamiento y computación, por lo que solo se pueden aplicar técnicas simples y colaborativas.

En los últimos años, se han desarrollado varios protocolos de procesamiento distribuido de datos para WSNs, basados principalmente en la agregación y fusión de datos durante el proceso de reporte de los mismos [50, 30]. Mediante el uso de este tipo de protocolos, los nodos pueden agregar o fusionar los datos transmitidos en los paquetes, provenientes desde diferentes fuentes, a medida que son transmitidos hacia el nodo sink. Para realizar esto, se utiliza una topología lógica de árbol, denominado *árbol de agregación*, con el nodo sink como raíz. Cada nodo reporta sus datos hacia el sink, y los nodos intermedios van agregando o fusionando los datos o la información, reduciendo así la cantidad de paquetes y de datos comunicados en la red. Estos protocolos tienen como desventajas que: i) se requiere conformar y mantener actualizada una topología lógica de enrutamiento de los paquetes hacia el sink, ii) necesidad de una buena sincronización entre los nodos para el correcto funcionamiento, y iii) un incremento en la latencia de reporte de datos a medida que la red es de mayor tamaño, debido a las tareas de computación. Sin embargo, se han propuesto variantes de este último caso, las cuales permiten seleccionar solo a aquellos nodos que pertenecen a una cierta zona geográfica de la red para participar en el proceso de reporte y procesamiento de los paquetes, como es el caso de *Directed Diffusion* [55]. Sin embargo, esta selección es impuesta externamente, basada en algún conocimiento previo de la red, por ejemplo, de que los nodos que proveen información útil pertenecen a una cierta zona geográfica de la red. Además, de que cada nodo debe tener información de su localización.

En general, para computar ciertos tipos de funciones, se requiere conocer todos los argumentos

presentes en la red. Ejemplos de ellos son la computación de histogramas, valores promedios, medios, etc. Sin embargo, existen ciertas funciones de mucho interés en WSNs, como por ejemplo las *funciones de umbral*, las cuales pueden ser computadas sobre un pequeño subconjunto $X' \subseteq X$ de valores, tal que $f(X') = f(X)$. Por lo tanto, **el problema de encontrar en la red el subconjunto X' se vuelve relevante debido a que este tiene el potencial de posibilitar el computo del valor de la función con menores costos de comunicación que el en caso de la agregación o fusión.**

Bajo esta hipótesis, en esta Tesis se plantea el desarrollo de protocolos de procesamiento y filtrado en red para WSNs. Los aportes fundamentales de esta Tesis son los siguientes:

- Desarrollo de protocolos para computar funciones de umbral dentro de la red, basado en un esquema de pregunta/respuesta.
- Diseño de heurísticas de búsqueda basadas en gradientes de información, y en la reducción iterativa del espacio de búsqueda. Se implementa el filtrado de datos y de nodos para reducir el costo de comunicación, mediante el uso heurísticas de búsqueda basadas en procesos físicos y bio-inspiradas en colonias, considerando esquemas de aprendizaje tanto centralizados como distribuidos.

En esta Tesis, se considera el problema de filtrar en la red aquellos argumentos (datos sensados) que no contribuyen o que son irrelevantes para el computo de la función de umbral requerida por el usuario. Para este fin, solamente se selecciona un conjunto de argumentos que puede ser usado para determinar el valor de la función. En realidad, esta selección constituye un proceso de precomputación del valor de la función para funciones simétricas de umbral, el cual puede ser realizado dentro de la red, con el fin de minimizar el uso de energía. La red puede proveer los mejores candidatos de argumentos para una función dada al nodo sink, el cual puede procesar luego estos datos para obtener el valor de la función. Por ejemplo, para el cómputo de la función máximo (o mínimo), solo el argumento con valor máximo (mínimo) puede ser provisto; mientras que para el caso de computo de rango, todos los argumentos que pertenecen al rango definido. Sin embargo, en general, se pueden tener dos situaciones respecto a los argumentos que se pueden proveer al nodo sink:

- *un gran conjunto de argumentos*, lo cual puede demandar elevados costos de comunicación, pero sin cometer errores de computación.
- *un pequeño conjunto de argumentos*, lo cual puede generar errores de computación.

En consecuencia, se buscan desarrollar protocolos con un *equilibrio entre costo y error*, tratando de minimizar ambos parámetros. Para cumplir con este objetivo, se incluyen estrategias de inteligencia computacional en los nodos que conforman la red, las cuales requieran de una baja capacidad de procesamiento, y sean basadas en el paradigma de procesamiento distribuido. Esta Tesis presenta nuestra contribución a la resolución de este tipo de problemas.

A continuación, se describen las herramientas y estrategias computacionales utilizadas para llevar adelante estos objetivos, las cuales involucran algoritmos metaheurísticos, algoritmos de aprendizaje por refuerzo, computación de funciones de forma distribuida, y filtrado de datos en red. Esta elección se basa en que las mismas son, dentro de las estrategias de inteligencia computacional, las más adecuadas para ser implementadas en WSNs, desde el punto de vista de los requerimientos computacionales, de memoria, y de flexibilidad [64].

3.2. Metaheurísticas

Las *metaheurísticas* son un tipo de métodos de optimización, desarrolladas para resolver diferentes tipos de problemas complejos de manera subóptima, pero con requerimientos computacionales y de tiempo muy bajos respecto a los métodos exactos. En la Figura 3.1 se describe una clasificación de los métodos de optimización existentes [65]. Como se puede ver en la figura, existe una gran variedad de métodos de optimización. Sin embargo, en esta Tesis nos centramos en los métodos aproximados, y específicamente, en las metaheurísticas basadas en solución simple, específicamente, las *basadas en procesos físicos* y las *bio-inspiradas en colonias*.

La palabra *heurística* tiene su origen en la palabra del griego antiguo *heuriskein*, que significa el arte de descubrir nuevas estrategias para resolver problemas. El subfijo *meta*, también de origen griego, significa *metodología de alto nivel*. El término *metaheurística* fué introducido por F. Glover en 1986. Los métodos de búsqueda metaheurísticos pueden ser definidos como metodologías generales de alto nivel que pueden ser usadas como estrategias de guiado en el diseño de heurísticas de capas inferiores para resolver problemas de optimización y de búsqueda. Las primeras heurísticas para búsqueda de información, basadas en búsqueda local, fueron propuestas a fines de la década del 40.

Las metaheurísticas de búsqueda son procedimientos iterativos que permiten guiar una heurística subordinada, combinando adecuadamente y de forma inteligente la *exploración* y *explotación* de los datos en el espacio de búsqueda. En este sentido, se debe buscar la forma más efectiva de combinar diferentes niveles, tanto de exploración (diversificación) como de explotación (intensificación), a fin de lograr el objetivo deseado, reduciendo parámetros tales como tiempo, computación, memoria, etc, con el mínimo error posible. En este sentido, por un lado, la intensificación o explotación permite reducir los costos de búsqueda y de comunicación en la red, a costa de incurrir en posibles errores de búsqueda. Por otro lado, la diversificación o exploración permite reducir los errores de búsqueda a costa de una mayor demanda de comunicación. Por lo tanto, se debe buscar que la heurística utilizada tenga un equilibrio respecto a estas características.

3.2.1. Tipos de Metaheurísticas

Del gran abanico de heurísticas desarrolladas hasta la actualidad, solo algunas pueden ser implementadas en forma distribuida en una WSN, debido a los requerimientos computacionales de las mismas, y a las limitaciones de los nodos. En esta Tesis se considera que, debido a la naturaleza distribuida de este tipo de redes, la mejor estrategia para resolver problemas de búsqueda de información es la implementación paralelizada o distribuida de algoritmos metaheurísticos dentro de la red. Es decir, que los algoritmos puedan ser multi-instanciados en los nodos.

Basado en esto, en esta Tesis se proponen dos protocolos para el procesamiento y filtrado de datos basados en heurísticas de búsqueda de solución simple. El primer protocolo se basa en recocido simulado, el cual es descrito en el Capítulo 4. Por otro lado, el segundo protocolo, el cual es presentado en el Capítulo 5, se basa en inteligencia colectiva en colonias, bajo la estrategia de reforzar aquellos caminos de búsqueda más adecuados, mediante el depósito de alguna sustancia, realizado por ejemplo, por comunidades de hormigas. A continuación se presentan las características de cada una de las metaheurísticas utilizadas por los protocolos desarrollados en esta Tesis.

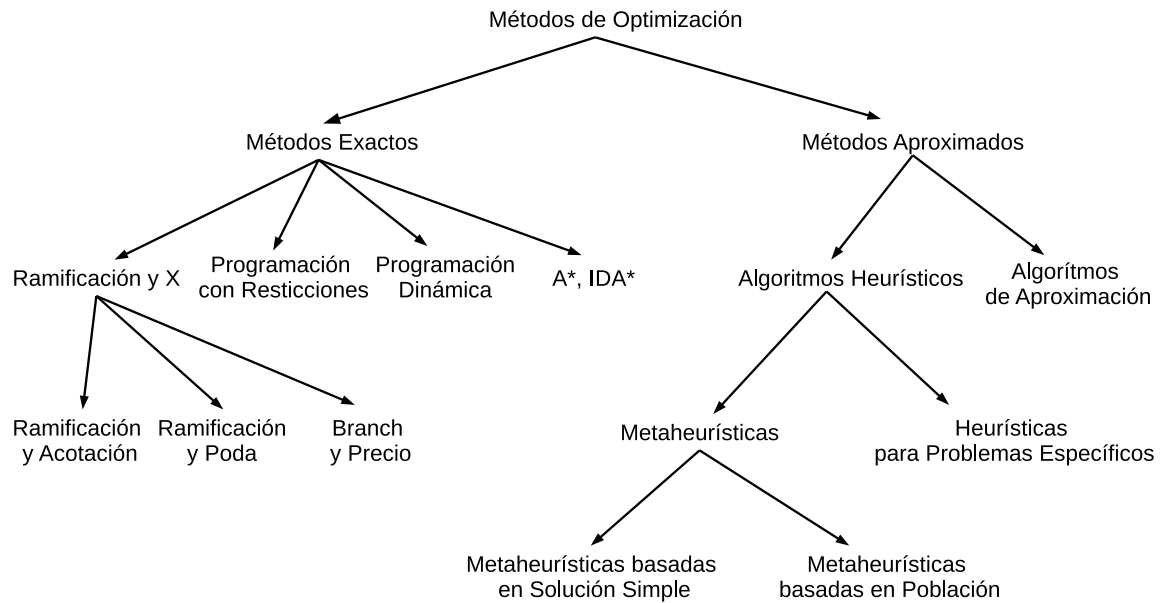


Figura 3.1: Métodos de optimización.

3.2.1.1. Heurísticas basadas en Recocido Simulado

Una heurística muy utilizada para resolver problemas complejos o intratables, debido principalmente al tamaño del espacio de búsqueda, es *Recocido Simulado* o *Simulated Annealing (SA)*. Esta heurística se basa en el proceso físico de recocido de metales, y fué propuesta por Kirkpatrick en 1983 [66]. El proceso de recocido de un metal consiste en calentarlo y luego dejarlo enfriar lentamente, volverlo a calentar a una temperatura menor y dejarlo enfriar nuevamente, y así sucesivamente, hasta cumplir con algún criterio de finalización. Bajo este proceso, se lleva a un metal a un estado de menor energía interna, es decir, una estructura cristalina más regular, lo que repercute en la resistencia mecánica del mismo. En analogía con la búsqueda de información, la función objetivo, como por ejemplo, el valor mínimo o máximo buscado en la red, representa a este estado de menor energía. En este sentido, en los últimos años, se han propuesto algunas implementaciones multi-instanciadas de SA para WSNs [35, 67].

3.2.1.2. Heurísticas basadas en Inteligencia de Colonias

En los últimos años, se han propuesto y desarrollado *algoritmos bio-inspirados basados en inteligencia de colonias* para ser aplicados en la resolución de una gran cantidad de problemas. Estas heurísticas tratan de imitar ciertos tipos de comportamientos ejecutados en forma colectiva por los seres vivos. Por ejemplo, la búsqueda de nuevas fuentes de alimento, de nuevos hábitats, la protección o mecanismos de defensa contra ataques de predadores, etc. Todos estos comportamientos colectivos están basados en el instinto de supervivencia de las especies. Algunos investigadores han demostrado que del comportamiento colectivo de seres vivos con inteligencia y capacidades simples y limitadas, y basado en algún mecanismo de comunicación determinado, emerge una inteligencia superior a la de la suma de las inteligencias individuales. Ejemplos de este tipo de heurísticas de optimización son las basadas en colonias de hormigas (*Ant Colony Optimization, ACO*), en enjambres de abejas (*Bee Colony*

Optimization, BCO), y en bandadas de pájaros o cardúmenes de peces (Particle Swarm Optimization, PSO), entre otras. Los mecanismos de comunicación entre individuos o agentes que componen la colonia, pueden ser directos (explícitos) o indirectos (implícitos). Los indirectos son basados, por ejemplo, en el depósito de alguna sustancia en el medio físico, que los individuos pueden interpretar, como en el caso del depósito de feromona en las colonias de hormigas. Este tipo de comunicación se denomina *estigmergía*. Por otro lado, la comunicación directa es como en el caso de las colonias de abejas, en el cual la comunicación es realizada explícitamente entre los individuos mediante un tipo de danza.

Particularmente, el algoritmo de optimización ACO, imita el comportamiento de colonias de hormigas durante la búsqueda de nuevas fuentes de alimento. Este tipo de problema está estrechamente relacionado con el problema de búsqueda de fuentes de información en WSNs que se considera en esta Tesis. Sin embargo, el protocolo propuesto en esta Tesis, basado en el concepto de depósito de feromona, difiere en muchos aspectos de los protocolos existentes derivados de ACO, lo que será descrito en los próximos capítulos.

3.3. Aprendizaje por Refuerzo

El *aprendizaje por refuerzo* (*reinforcement learning, RL*) es una sub-área del aprendizaje automático y de la inteligencia computacional o artificial. RL es un algoritmo que imita el comportamiento de un agente que aprende de su experiencia, mediante la interacción directa con su entorno, y sin la intervención de un supervisor que le indique que decisiones debe tomar. Este tipo de aprendizaje, en el que está contenido RL, se conoce como aprendizaje no-supervisado. Esta técnica computacional de aprendizaje por refuerzo nació a fines de los años '80, inspirada en modelos de aprendizaje provenientes de la psicología, principalmente de los estudios sobre condicionamiento Pavloviano basado en aprendizaje por premio y castigo. Su desarrollo aplicado a las Ciencias de la Computación ha llevado a importantes avances en los últimos años [68].

En RL, un agente aprende de su entorno y va mejorando su comportamiento mediante interacciones de prueba y error en un ambiente dinámico e incierto (Figura 3.2). En este contexto, el término comportamiento se refiere a la toma secuencial de decisiones por parte del agente.

La ventaja de esta técnica de aprendizaje, es la simplicidad de su implementación, y su bajo requerimiento computacional. Es por ello que se la ha propuesto, junto con las heurísticas inspiradas en procesos físicos y las bio-inspiradas, como las más adecuadas para ser aplicadas en WSNs [64, 69].

En los protocolos de procesamiento y filtrado desarrollados en esta Tesis, se aplican diferentes estrategias de aprendizaje por refuerzo. Por ejemplo, en la Sección 4.2.3, se utiliza esta técnica para que el nodo sink pueda adaptar la disseminación de requerimientos o búsqueda de información a la distribución de datos presentes en la red, o en la Sección 6.6.4, para que cada nodo pueda hacer frente a la pérdida de paquetes en el reporte de argumentos hacia el sink, denominado mecanismo de auto-curado (self-healing).

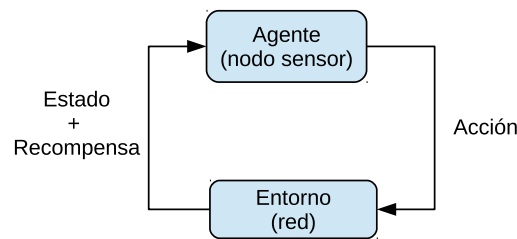


Figura 3.2: Aprendizaje por refuerzo.

3.4. Elección de Estrategia de Operación

En general, se pueden considerar dos estrategias o modos de operación desde el punto de vista de como es implementado el *proceso de pregunta/respuesta*. Al primero, lo denominaremos *de requerimiento simple o single-query, SQ*, y al segundo, *de reducción iterativa del área de búsqueda o del área consultada o iterative query with space reduction, IQSR*. Consideraremos al protocolo PASA, el cual se describirá en el Capítulo 4, como un protocolo enmarcado dentro del segundo caso. El primer caso está basado en el reporte inicial de todos los datos al sink, y la posterior actualización de aquellos datos de cada nodo que superen un cierto porcentaje respecto al valor ya reportado. Por otro lado, el segundo caso está basado en la búsqueda de datos en toda la red, y su posterior reducción iterativa de búsqueda hasta converger a un estado de menor energía en el que no se pierdan fuentes relevantes de datos. En este contexto, el hecho de buscar inicialmente en toda la red se debe a que no se quiere perder información para el cómputo de la función, y termina siendo un caso inverso al de la búsqueda de eventos en la red mediante la búsqueda por expansión de anillos (ERS).

En el caso de SQ, el nodo sink disemina inicialmente el requerimiento a todos los nodos de la red, basado en inundación controlada, y cada nodo reporta sus datos al nodo sink, la cual es almacenada en este último. Como cada nodo mantiene información del valor inicial reportado, solo se reportarán datos desde cada nodo, cuando haya algún cambio significativo en sus propios valores sensados, como para que sean actualizados en el nodo sink. Es decir, los datos se reportan solo desde aquellos nodos que detectan algún cambio significativo mayor a un cierto porcentaje predefinido. Por otro lado, en el caso de PASA, en cada iteración se realiza la diseminación de requerimientos y el reporte de los datos, pero se pasa de inundar la red, a ir reduciendo iterativamente el espacio de búsqueda, mientras se considere que no hay pérdida de información.

En la Figura 3.3 se muestra el costo total de comunicación, es decir, el costo de diseminación de requerimientos más el costo de reporte de los datos realizado en promedio por cada nodo, para ambos modos de operación, y para el caso de una red de 100 nodos. El porcentaje representa el cambio entre iteraciones en el dato obtenido por cada nodo. Si analizamos el costo total de comunicación podemos ver que si no se producen cambios en los valores sensados, es decir, que tenemos un 0% de cambio, el esquema SQ es el más conveniente, ya que una vez reportados todos los datos de los nodos al sink, no se producen reportes por actualizaciones. Sin embargo, ya con un cambio del 1% en los valores sensados por cada nodo se produce un costo de comunicación muy elevado respecto al caso de PASA, en el que dicha variación no lo afecta.

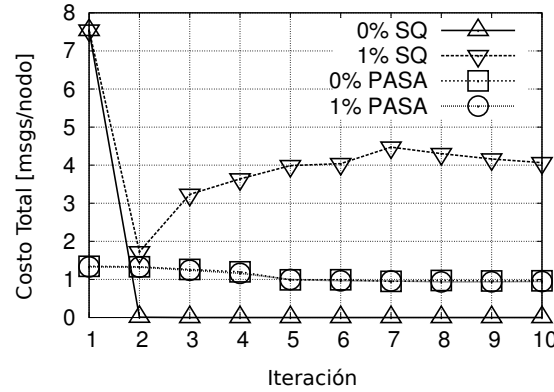


Figura 3.3: Comparación de costos de comunicación de diferentes estrategias de pregunta/respuesta.

3.5. Esquema de Procesamiento Propuesto

En la mayoría de las aplicaciones de WSNs, se busca computar una función sobre los datos sensados en la red. Esto se puede formalizar como computar una función $f_i(X)$ en la iteración i , donde X es el conjunto de valores sensados en la red desde x_1 hasta x_n , siendo n el número de nodos de la red. En general, todos los valores x podrían ser reportados al nodo sink, para computar el valor de la función en una manera centralizada. En este caso, el nodo sink no incurre en errores de computo a expensas de un elevado costo de comunicación, y como consecuencia de energía, resultante de tener que reportar todos los datos hacia el sink. Este elevado costo de comunicación es cada vez más significativo a medida que aumenta el tamaño de la red. Sin embargo, funciones simétricas de umbral (type-threshold functions) pueden ser computadas típicamente sobre un subconjunto $X' \subseteq X$ de valores tal que $f(X') = f(X)$. Por lo tanto, **el problema de encontrar en la red el subconjunto X' se vuelve relevante debido a que este tiene el potencial de posibilitar el computo del valor de la función a menores costos de comunicación.**

Se define $\hat{X}' = g(X)$ como el conjunto de argumentos provistos por una función de filtrado en red $g(\cdot)$ dada. Si $\hat{X}' \supseteq X'$, $g(\cdot)$ no introduce error de computación. Sin embargo, si $\hat{X}' \not\supseteq X'$, el proceso de filtrado introduce algún error de computación $e = \frac{|f(X') - f(\hat{X}')|}{f(X')}$. El costo de comunicación c es difícil de estimar, pero se espera que este sea inversamente proporcional al tamaño del subconjunto reportado $|\hat{X}'|$. Esto se debe al hecho que como $\hat{X}' \rightarrow X'$, mayor intercambio de mensajes es requerido por el esquema de filtrado $g(\cdot)$. Considere por ejemplo inundar la red con un requerimiento relacionado con computar un rango dado de valores. Cada nodo cuyo valor sensado pertenece o está incluido en el rango definido por el requerimiento puede reportar su valor hacia el nodo sink. Debido a que todos los nodos reciben el requerimiento, entonces $\hat{X}' = X'$ y $|\hat{X}'| = |X'|$. Sin embargo, si el esquema de requerimiento está basado en random walk, no todos los nodos con valores relevantes pueden recibir el requerimiento. Por lo tanto, se espera que $\hat{X}' \not\supseteq X'$ con $|\hat{X}'| < |X'|$, lo cual introduce algún error de computación pero a bajos costos de comunicación debido a que no es explorada toda la red.

La computación de una función $f(\cdot)$ puede ser realizada en dos partes. Una primera parte, dada por $\hat{X}' = g(X)$, la cual es computada dentro de la red; y una segunda, dada por $f(\hat{X}')$, la cual es realizada en el nodo sink. Ambos procesos son descriptos en la Figura 3.4. Debido a que $g(\cdot)$ es implementada de una manera distribuida, esta puede ser gobernada por dos reglas de filtrado: una

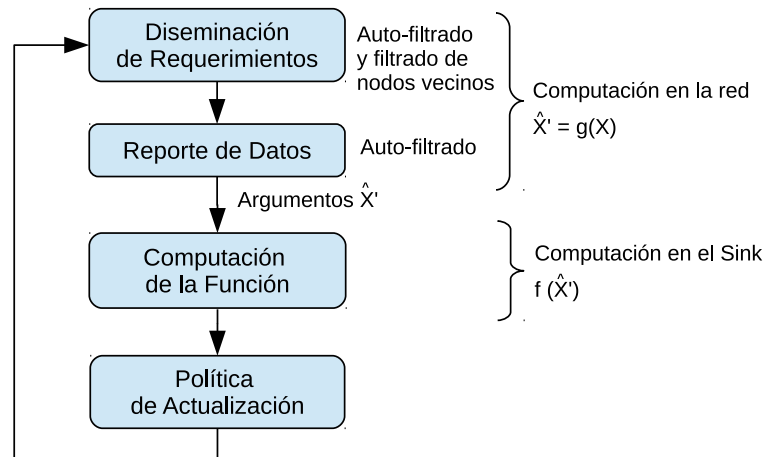


Figura 3.4: Procesos involucrados en la computación de una función.

regla de auto-filtrado, la cual permite determinar si el valor sentido por el nodo es relevante o no para la computación de la función $f(\cdot)$, y una *regla de filtrado de nodos vecinos*, la cual determina si conviene continuar retransmitiendo el requerimiento en la red. Tenga en cuenta que si el requerimiento no es retransmitido a los nodos vecinos, estos últimos pueden ser potencialmente excluidos del proceso de computación en red, debido a que sus valores sentidos no son considerados para ser reportados hacia el nodo sink.

La regla de auto-filtrado es simple de implementar, debido a que solo requiere comparar el dato local en el nodo contra un umbral contenido en el mensaje de requerimiento. Sin embargo, se puede considerar computar funciones con umbrales tanto fijos como dinámicos. Umbrales fijos [70, 71] son usados en la búsqueda de valores sentidos dentro de un rango específico de valores, mientras que los de umbral dinámico, son usados para valores sin un rango predefinido. Umbrales fijos pueden ser implementados localmente por cada nodo independiente de los valores disponibles en otros nodos, mientras los de umbral dinámico dependen de ellos para actualizar los valores de umbral [72]. El ejemplo canónico para el caso de umbral dinámico, es la función *máximo*, donde, dado un umbral inicial, todos los nodos cuyos valores están por encima del umbral lo actualizan sobre el mensaje de requerimiento. De esta forma, el umbral tiende a incrementarse a medida que el mensaje de requerimiento es diseminado a través de la red. En general, consideramos en nuestro análisis, el caso de computar funciones máximo/mínimo, debido a que estas hacen uso de umbrales dinámicos; sin embargo, el análisis puede ser también extendido a casos de umbral fijo.

Entre ambas reglas, la de filtrado de vecinos es la más desafiante, debido a que la misma requiere de un cierto aprendizaje en los nodos sobre en que lugares de la red hay presentes datos relevantes para el computo de la función. Debido a que los datos obtenidos por los nodos sensores pueden cambiar en el tiempo, este proceso necesita ser robusto y tener capacidad de detectar y seguir cambios que puedan conducir a considerar nuevos nodos y/o descartar nodos existentes. Es decir, soluciones al problema de computar \hat{X}' necesitan hacer frente a estos desafíos.

La computación final de la función es realizada por el nodo sink en base a los argumentos \hat{X}' reportados por los nodos. Por otro lado, la política de actualización puede ser realizada de forma centralizada, como se indicará en el protocolo propuesto en el Capítulo 4, o de forma distribuida,

como se indicará en el protocolo propuesto en el Capítulo 5

3.6. Protocolos de Filtrado en Red

La aplicación de filtrado en WSNs fue inicialmente propuesta en el desarrollo del protocolo Directed Diffusion [73]. Sin embargo, su aplicación estaba basada en la detección de eventos que coinciden con el interés de un usuario, y no para la computación de funciones estadísticas de los datos obtenidos por los nodos, como se considera en esta Tesis. En este contexto, una vez que un nodo recibe un mensaje, ya sea este un requerimiento o una respuesta, el mismo implementa *auto-filtrado*. Este tipo de filtrado es realizado por un nodo mediante la comparación de su lectura actual con respecto al umbral transportado en el mensaje de requerimiento o con el argumento transportado en el mensaje de respuesta. Sin embargo, este esquema de auto-filtrado no introduce errores de computación, pero puede incrementar el costo de comunicación debido al intercambio de mensajes para filtrar nodos sin datos relevantes para la aplicación. Por el contrario, el *filtrado de vecinos*, el cual decide si vale la pena transmitir un requerimiento a los nodos vecinos, puede introducir errores de computación, pero tiende a decrementar los costos de comunicación por evitar consultar a nodos que puedan tener datos no útiles.

En este sentido, en esta Tesis se proponen dos protocolos para implementar el proceso de filtrado de vecinos. Si bien los principios de funcionamiento respecto a la computación de funciones y al auto-filtrado son los mismos en ambos protocolos, existen diferencias significativas en lo que respecta a la decisión de retransmisión del requerimiento, es decir, la decisión de filtrado de nodos vecinos, y al esquema de aprendizaje. El primero de estos protocolos considera que los nodos no mantienen ningún estado sobre decisiones previas de filtrado, referido como *filtrado sin estados*. Sin embargo, el nodo sink puede aprender de la información obtenida de la red y modificar la regla de filtrado a lo largo del tiempo, mediante el proceso de embeber alguna información de estado en el mensaje de requerimiento, que puede ser usado para recolectar solamente datos relevantes a un menor costo de comunicación. Por otro lado, en el segundo protocolo, denominado *filtrado con estados*, los nodos mantienen información acerca de decisiones previas de filtrado, mientras el nodo sink siempre mantiene la misma regla de filtrado. Ambos protocolos pueden ser descritos por:

- Un algoritmo de retransmisión en bajada, el cual implementa la regla de filtrado de vecinos para enrutar los requerimientos a través de la red.
- Un proceso de aprendizaje en subida, el cual adapta la dinámica de la regla de filtrado de vecinos en el tiempo.

En los siguientes capítulos se describen en detalle estos protocolos propuestos. En el Capítulo 4 se describe en detalle el *protocolo de filtrado sin estados*, mientras que el *protocolo de filtrado con estados* se describe en el Capítulo 5.

Como caso de estudio, se analiza en esta Tesis la computación de funciones de umbral, descritas en el capítulo anterior. A continuación se describen las estrategias de procesamiento de la información consideradas en esta Tesis.

3.7. Estrategias de Procesamiento

En esta Tesis se proponen varias estrategias para procesamiento de datos para WSNs, diferentes a las de los esquemas existentes.

El mecanismo utilizado en esta Tesis está basado en un proceso de pregunta/respuesta iterativo, que permite realizar el pre-procesamiento de los datos dentro de la red, y posteriormente el procesamiento final de los mismos en el nodo sink. Además, se tiene la posibilidad de limitar la participación de los nodos basado en aprendizaje por refuerzo. Es decir, se aplica procesamiento basado en filtrado de datos, combinado con búsqueda heurística de nodos con información útil o filtrado de nodos vecinos, en el proceso de pregunta o diseminación de requerimientos, con el fin de seleccionar nodos para reportar sus datos como respuesta a la solicitud. En este sentido, cada nodo puede procesar y filtrar datos localmente cuando recibe un paquete, resultando en una pre-computación de los datos dentro de la red, y como consecuencia, una reducción de la cantidad de paquetes comunicados en la red.

Como se comentó en la Sección 2.3, las estrategias y protocolos de enrutamiento desarrollados para WSNs pueden ser basados o en direcciones o basados en datos. Actualmente, los protocolos para WSNs son en su mayoría basados en datos, fundamentado en que lo importante en este tipo de redes son los datos y no las identidades de quienes los reportan. Sin embargo, nuestra propuesta considera que la mejor opción es utilizar una combinación de ambos esquemas, aprovechando ambos beneficios. En este sentido, se utiliza para la búsqueda de datos relevantes para la aplicación el esquema basado en datos, utilizando gradientes de información para enrutar a los requerimientos hacia aquellos nodos de la red que provean información útil. El esquema basado en direcciones se utiliza tanto para los procesos de aprendizaje, que se aplican en los esquemas propuestos, como para el enrutamiento de las respuestas. A continuación se describen en detalle los procedimientos involucrados en el procesamiento de la información.

3.8. Procedimientos

Los esquemas utilizados para el procesamiento de funciones en WSNs, propuestos en esta Tesis, se basan en procedimientos instanciados de forma secuencial e iterativa cuando un usuario solicita información a la red. Es decir, se tiene una secuencia de procedimientos, que en conjunto, conforman una iteración de computación, la cual es repetida en el tiempo. Debido a la variabilidad de las WSNs (inclusión de nuevos nodos, nodos que quedan sin energía, movilidad, paquetes perdidos, etc), se considera que el esquema basado en requerimientos es más conveniente para este tipo de aplicación, debido a que este esquema permite muestrear el estado de la red en cada diseminación de requerimientos. Estos procedimientos incluyen:

- diseminación controlada de requerimientos a los nodos de la red.
- reporte de los datos obtenidos por los nodos hacia el nodo sink.
- computación de la función en el nodo sink.
- ajuste o actualización del funcionamiento de los nodos basado en aprendizaje.

La idea de la diseminación controlada es que, mediante algoritmos heurísticos de búsqueda y técnicas de aprendizaje, se pueda reducir el área de exploración, o sea, la cantidad de nodos consultados o que participen en la tarea de procesamiento, asegurando que no se pierdan argumentos relevantes para computar la función requerida. El término *argumento relevante* se refiere a datos que aporten información útil para la computación de la función dada. En resumen, se busca converger a configuraciones de búsqueda que, con el menor costo de comunicación y de error se puedan obtener los argumentos relevantes para computar la función deseada.

A continuación, se describe la arquitectura desarrollada para ser implementada dentro de cada nodo, para el filtrado, procesamiento de la información, y toma de decisiones necesarios para la aplicación.

3.9. Arquitectura

La *arquitectura* desarrollada en esta Tesis, para ser instanciada en cada nodo, permite realizar cierto tipo de procesamiento local de la información, con filtrado de datos y de nodos vecinos, basado en toma de decisiones. Este procesamiento se realiza en cada nodo cuando se reciben paquetes, estrategia conocida como *procesamiento de datos on-the-fly*. El filtrado de datos es sencillo de implementar, ya que solo consiste en comparar el dato transportado en el paquete, respecto al valor actual del nodo que lo recibe. Sin embargo, el filtrado de nodos vecinos no es trivial, debido a que los nodos desconocen los valores sensados en la red fuera de su entorno inmediato. El aporte de esta Tesis es el desarrollo de dos protocolos de filtrado de nodos vecinos, basados en control centralizado o distribuido, los cuales se describen en los siguientes capítulos.

Esta arquitectura, descrita en la Figura 3.5, está conformada inicialmente por una *etapa de auto-filtrado*, que determina si la lectura del nodo es relevante para la aplicación, seguida por una *etapa de filtrado de nodos vecinos*, que determina si continuar la búsqueda de datos mediante la retransmisión del requerimiento a nodos vecinos. El proceso de auto-filtrado no tiene complejidad, y consiste en comparar la información o umbral contenido en el payload del paquete recibido, respecto al dato sensado por el nodo. Sin embargo, el filtrado de nodos vecinos es más complejo, debido a que inicialmente no se conoce la distribución de los datos en la red. Para ello, se proponen dos protocolos de filtrado de nodos vecinos, uno basado en el proceso físico de recocido de metales con control centralizado, denominado *Parallel Adaptive Simulated Annealing (PASA)*, y otro totalmente descentralizado basado en procesos bio-inspirados de inteligencia en colonia, el cual es denominado *Pheromone-based In-Network Processing (PhINP)*.

Desde el punto de vista de que tipo de función simétrica se va a computar, y como se describió en la Sección 2.2.4, el *cómputo de funciones simétricas de umbral estático* es trivial, y consiste en propagar un requerimiento de computación con un valor de umbral predefinido. Cada nodo que recibe este requerimiento determina si sus datos son relevantes en base a este umbral. Sin embargo, el *cómputo de funciones simétricas de umbral dinámico* requiere de esquemas de búsqueda y filtrado con aprendizaje de la red, debido a que es un tipo de búsqueda desinformada. Para este fin, se aplican técnicas simples de inteligencia computacional a cada nodo de la red para dotarlos de cierta capacidad de toma de decisiones basado en su conocimiento local. Estas técnicas son un conjunto de metodologías computacionales utilizadas para hacer frente a problemas complejos del mundo real en los que las

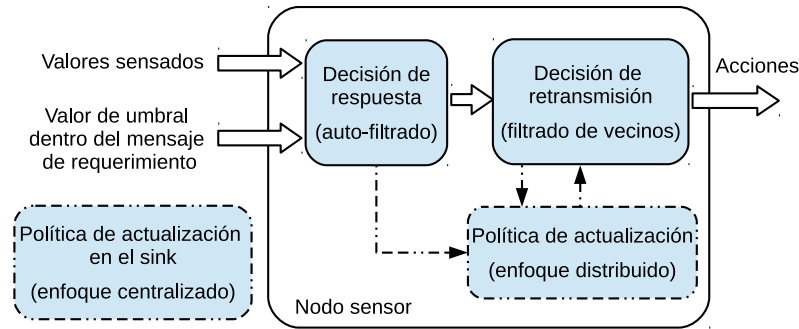


Figura 3.5: Arquitectura de filtrado, procesamiento y toma de decisiones instanciada en cada nodo.

metodologías tradicionales son ineficientes o inviables. En esta Tesis, estas técnicas son utilizadas para dotar a los nodos de dos reglas de filtrado, como se indica en la Figura 3.5. Las mismas son descritas en detalle a continuación.

3.9.1. Reglas de Decisión y Filtrado

Cuando un nodo sensor recibe un requerimiento para el cómputo de algún tipo de función de los datos sensados, la primera regla de decisión evaluada, llamada *regla de auto-filtrado*, determina si el dato presente en el nodo constituye un argumento relevante para ser reportado al nodo coordinador. Posteriormente, se evalúa una segunda regla, llamada *regla de filtrado de vecinos*, la cual determina si los nodos vecinos pueden tener argumentos relevantes para continuar retransmitiendo el requerimiento. Es decir, si se asume que los datos de los nodos vecinos son irrelevantes, el requerimiento puede llegar a ser retransmitido solo con una baja probabilidad. Sin embargo, estas dos reglas no son estáticas, es decir, no son preconfiguradas, sino que se construyen en la marcha durante la diseminación de requerimientos, mediante el uso de técnicas de aprendizaje. Debido a que el proceso de búsqueda y respuesta es iterativo y probabilístico, el comportamiento de la red resulta complejo de analizar, y puede ser analizado con herramientas de simulación de redes. La red evoluciona desde un estado sin filtrado a uno con filtrado por medio de decisiones locales tomadas por cada uno de sus nodos.

La *regla de auto-filtrado* es simple de implementar, debido a que solo requiere evaluar el dato local en el nodo contra un umbral transmitido en el paquete de requerimiento. Sin embargo, la complejidad en la toma de decisiones se concentra en la segunda regla de filtrado (*filtrado de nodos vecinos*). Este tipo de filtrado es desafiante, debido a que requiere aprender si pueden existir datos relevantes para el cómputo de la función dada más allá de la posición del nodo, a través de la dirección de diseminación de requerimientos. Debido a que los datos obtenidos por los nodos sensores pueden cambiar en el tiempo, este proceso necesita ser robusto y tener capacidad de seguir cambios que puedan conducir a considerar nuevos nodos y/o descartar nodos existentes. Es decir, soluciones al problema de computar \hat{X}' necesitan hacer frente a estos desafíos. Para hacer frente a esto, en esta Tesis se proponen dos protocolos diferentes. El primero está basado en el *concepto de recocido simulado (simulated annealing, SA)*, el cual es instanciado en cada nodo, o sea múltiplemente instanciado en la red, centralizando el aprendizaje en el nodo sink para actualizar la política global de retransmisión de requerimientos, la cual permita obtener datos de áreas relevantes para la aplicación. Por otro lado, el segundo protocolo distribuye el control y cierto aprendizaje en los nodos de la red para

actualizar localmente la política de diseminación de requerimientos. Esto es logrado por medio de una versión modificada del algoritmo de optimización por colonia de hormigas (ant colony optimization, ACO). Este es un algoritmo bio-inspirado, basado en el comportamiento de las hormigas cuando buscan fuentes de alimentos y forman caminos óptimos entre la colonia y dichas fuentes. Cuando estos protocolos son implementados iterativamente, se adaptan a la red y permiten proveer un conjunto de argumentos para la computación final en el nodo coordinador, con una buena performance en terminos tanto de costos de comunicación como de error de computación.

Las principales métricas analizadas a lo largo de esta Tesis son: el *error de cómputo* y el *costo de diseminación de requerimientos*, considerándose en los análisis simplemente como *error* y *costo*, respectivamente. Este costo es el de mayor interés, ya que es el de mayor proporción en protocolos de computación de funciones de tipo umbral, basados en requerimientos o de consulta/respuesta, en los cuales se reportan solo pocos datos relevantes al nodo sink. En este contexto, en el caso de tratarse del costo total de comunicación, es decir el costo de diseminación de requerimientos más el costo de respuesta, se describirá como *costo total*.

En los siguientes capítulos se describen en detalle los protocolos desarrollados en esta Tesis para el filtrado de nodos vecinos en WSNs. En el Capítulo 4 se presenta el *protocolo de filtrado sin estados*, el cual está basado en el concepto de recocido simulado, mientras que el *protocolo de filtrado con estados*, el cual está basado en inteligencia bio-inspirada de colonias de hormigas, es presentado en el Capítulo 5.

3.10. Comentarios de Cierre

En este capítulo se describieron los mecanismos de inteligencia computacional desarrollados en esta Tesis para ser aplicados en los nodos y en las WSNs en general. Estos mecanismos van desde las heurísticas de búsqueda basadas en procesos físicos, como recocido simulado, las basadas en inteligencia colectiva o en colonias de hormigas, hasta el aprendizaje por refuerzo de un agente. Además, se describe la elección de la estrategia de operación basada en reducción iterativa del espacio de búsqueda (o de filtrado iterativo de nodos), la cual es mas eficiente que la de requerimiento simple con actualización de datos ante la dinámica del parámetro físico. Se presentan las estrategias de procesamiento de funciones en WSNs, y los procedimientos utilizados para tal fin. Finalmente, se describe la arquitectura implementada en cada nodo para la computación de funciones, y la toma de decisiones de filtrado tanto local como de nodos vecinos. Además, se definen las reglas de decisión y filtrado local y de vecinos.

Protocolo de Procesamiento sin Estados

Contenido

4.1	Introducción	65
4.2	Funcionamiento	66
4.2.1	Diseminación de Requerimientos	67
4.2.2	Reporte de Respuestas	70
4.2.3	Aprendizaje	71
4.3	Comentarios de Cierre	76

4.1. Introducción

EL primer protocolo de procesamiento y filtrado, desarrollado en esta Tesis, se denomina *protocolo sin estados* [41]. Este nombre se debe a que no se mantienen estados dentro de la red, es decir, que los nodos que conforman la red no almacenan estados relacionados con tareas de procesamiento y filtrado ejecutadas previamente. Ello se debe a que el control de funcionamiento de este protocolo es definido de manera centralizada por el nodo coordinador o sink.

El objetivo de este protocolo es el de realizar procesamiento y filtrado de datos y de nodos vecinos dentro de la red que no son útiles para la aplicación, con el fin de reducir el elevado costo de comunicación de reportar al nodo sink todos los datos obtenidos en la red. Mas específicamente, el procesamiento de los datos es realizado en dos etapas independientes y consecutivas, como se describió en la Sección 3.7. Se realiza parte del procesamiento y filtrado dentro de la red, y la información resultante es provista al nodo sink, quién es el que realiza la etapa final de procesamiento para obtener el resultado buscado. Es decir, se busca preprocesar los datos dentro de la red, filtrando información y la participación de nodos no relevantes para la aplicación.

En este contexto, una vez que el usuario solicita a la red, a través del nodo sink, realizar la computación de una función dada de los datos presentes en la red, se ejecutan tres procesos o fases

secuenciales dentro de la misma, basados en un modelo de pregunta/respuesta. La primer fase consiste en la *diseminación de requerimientos* por parte del nodo sink a la red. Por otro lado, la segunda fase consiste en el *reporte de datos relevantes hacia el sink*. Finalmente, se implementa una fase de *aprendizaje por refuerzo* en el nodo sink para mejorar el funcionamiento del protocolo en función de las características de la red.

Esta secuencia de tres procesos, definidos para el procesamiento y filtrado de datos y de nodos, y que se denomina *iteración*, es realizada repetitivamente con el fin de que el nodo sink, mediante el aprendizaje por refuerzo, pueda ajustar en cada iteración la energía con la que es inyectado cada requerimiento a la red. En este sentido, y a diferencia de aplicaciones de detección de eventos, en los que se puede ir incrementando gradualmente el área de búsqueda o de diseminación de requerimientos, en este caso de computación de funciones estadísticas, el nodo sink realiza el proceso inverso. Es decir, este nodo realiza la primer búsqueda en profundidad, es decir, inyectando a la red un mensaje de requerimiento con una gran energía o temperatura, con lo cual se inunda la red, pudiendo llegar con el mismo a cada nodo. Posteriormente, el nodo sink va disminuyendo la energía de este mensaje en sucesivas iteraciones, con el fin de reducir el espacio de búsqueda, mientras considere que no hay pérdida de información. Con esto, y haciendo uso de búsqueda basada en gradientes, se trata de diseminar requerimientos solo a aquellos nodos con datos relevantes para la función que se está evaluando, filtrando la participación de nodos no relevantes, y minimizando errores de búsqueda. Es decir, no tiene sentido solicitar información a nodos que no aportan datos útiles para la aplicación, sino que lo que se busca es solicitar información solo a aquellos nodos que contribuyan con la aplicación requerida. Cada nodo sensor que recibe un requerimiento, ejecuta localmente acciones basadas en reglas de decisiones simples, gobernadas por el nivel de energía de los mensajes de requerimiento recibidos, la cual es función de la secuencia o historia seguida por los mismos, debido a que estos sufren cambios a medida que son diseminados a través de la red.

En resumen, este protocolo de procesamiento y filtrado permite adaptar el proceso de diseminación de requerimientos en función de la distribución de los datos en la red con un control centralizado en el nodo sink. En este sentido, se filtra la participación de nodos no relevantes para la aplicación, convergiendo a un estado de baja energía que asegura obtener argumentos relevantes minimizando el error. A continuación se describe en detalle el funcionamiento de este protocolo de procesamiento y filtrado, y la secuencia de fases implementadas en cada iteración.

4.2. Funcionamiento

Este protocolo, denominado *sin estados*, implementa una versión de la metaheurística de recocido simulado (simulated annealing, SA) propuesta por Kirkpatrick et al. en 1983 [66]. SA tiene la capacidad de permitir la exploración de datos mas allá de áreas sin información relevante para la aplicación. Ello se debe a su comportamiento estocástico basado en la probabilidad de transición para pasar de un estado s a otro nuevo estado s' es especificado por una cierta función de probabilidad de aceptación ($P(e, e', T)$) que depende de las energías $e = E(s)$ y $e' = E(s')$ de los dos estados, y de una variable T llamada temperatura que permite controlar el funcionamiento. La probabilidad de aceptar la transición a un nuevo estado es dada por:

$$P(\Delta E, T) = e^{-\frac{\Delta E}{T}}. \quad (4.1)$$

donde T es la variable temperatura del algoritmo SA, y ΔE representa la diferencia de energía entre los dos estados considerados (e, e') .

Para un valor grande de T ,

Para un valor bajo de T ,

SA aplica un mecanismo de enfriamiento del parámetro T , para lograr la convergencia del algoritmo.

La versión propuesta en esta Tesis, denominada *Parallel Adaptive Simulated Annealing (PASA)* [39], resulta en una paralelización natural del algoritmo SA, debido a que cada paquete de requerimiento es transmitido a más de un nodo a la vez mediante broadcast. En este sentido, se generan múltiples instanciaciones de este algoritmo en la red, es decir, cada nodo instancia este algoritmo para tomar decisiones locales. Para agilizar la convergencia de este algoritmo, la cual es una de las desventajas de SA, se agregan dos estrategias de enfriamiento, una *estrategia de enfriamiento lineal dentro de cada iteración*, y una *estrategia de enfriamiento logarítmico entre iteraciones sucesivas*.

PASA es un protocolo basado en requerimientos (query-driven), conformado por tres fases ejecutadas secuencialmente, las cuales son repetidas iterativamente. La primera fase es de diseminación de requerimientos en bajada o downstream forwarding (desde el sink hacia los nodos sensores), la segunda fase es de reporte de respuestas en subida o upstream learning (desde nodos sensores seleccionados hacia el sink), y finalmente la última fase de aprendizaje en el nodo sink. En este sentido, la suma de estas fases constituye una iteración de computación. La primera fase es utilizada para diseminar o transmitir el requerimiento a los nodos sensores presentes en la red, siguiendo un modelo de flujo de datos de uno-a-muchos. Por otra parte, la segunda fase es implementada para reportar los datos desde nodos seleccionados con información relevante hacia el nodo sink, siguiendo un modelo de flujo de datos de uno- o muchos-a-uno, dependiendo del número de fuentes detectadas en la red. Estos nodos seleccionados proveen de valores o argumentos relevantes para la aplicación particular. Este proceso es ejecutado en forma iterativa, con la finalidad de obtener soluciones óptimas haciendo uso de reducción del espacio de búsqueda. Las fases descritas anteriormente son detalladas a continuación.

4.2.1. Diseminación de Requerimientos

Se considera una WSN de múltiples saltos, en la que los nodos están distribuidos uniformemente en un área de interés, y donde el nodo sink, que es el encargado de inyectar requerimientos de información y de recibir los datos para ser reportados al usuario, es localizado en el centro de dicha área. Cuando se solicita a la red computar alguna función de los datos sensados, por ejemplo calcular el valor máximo, mínimo, medio, etc, el nodo sink inyecta a la red un requerimiento de información específico mediante broadcast. Cada nodo que recibe este requerimiento *puede retransmitirlo al menos una vez en cada iteración de computación*. El mensaje de requerimiento está conformado por un valor de temperatura T del algoritmo, y por el mejor valor encontrado hasta el momento a lo largo del camino seguido por

el requerimiento (denominado umbral o threshold, T_h). En este contexto, ambos parámetros pueden ser modificados por cada nodo que retransmita este mensaje. Como consecuencia, los requerimientos diseminados en la misma iteración pueden tener diferentes valores de T y T_h , dependiendo de la secuencia seguida por cada uno de ellos.

Para limitar el problema de explosión de comunicación debido al uso de inundación de mensajes usando broadcast, se define como regla obligatoria, *que cada nodo no puede retransmitir más de una vez el mismo mensaje en la red*. Es decir, que independientemente si un nodo retransmitió o no un mensaje de requerimiento con un *id* dado, si lo vuelve a recibir a través de otro nodo vecino, este lo tiene que eliminar, ya que fué procesado por él anteriormente. En este sentido, en cada iteración de computación el nodo sink inyecta mensajes de requerimiento con un *id* diferente, modificándolo en forma incremental.

El comportamiento de los nodos puede ser dividido en tres casos, según la decisión de retransmisión tomada por los mismos. En este contexto, **cuando hablamos de decisión de retransmisión de requerimientos por parte de un nodo, estamos implícitamente hablando de filtrado de nodos vecinos, ya que la decisión tomada por un nodo de no retransmitir un requerimiento implica el filtrado o la no participación de sus nodos vecinos en la computación de la función solicitada, y viceversa**. Esta decisión de retransmisión del mensaje de requerimiento está basada tanto en el valor de ΔE , como en el valor de temperatura T transportado en el mensaje. El primer parámetro es la diferencia entre el umbral T_h transportado en el mensaje y la lectura actual en el nodo \hat{x}' , es decir $\Delta E = T_h - \hat{x}'$, y el segundo es el valor actual de temperatura que tiene el mensaje, debido a que el mismo puede ir sufriendo degradaciones (enfriamiento lineal) a medida que pasa por condiciones desfavorables, es decir, cuando el gradiente de información es negativo. A continuación, se describen cada uno de los tres tipos de comportamiento que puede tener un nodo durante la diseminación de requerimientos.

4.2.1.1. Selección de Nodo a Responder y Retransmisión del Requerimiento

Aquellos nodos que tienen valores relevantes para la aplicación (es decir, $\Delta E < 0$) son seleccionados para reportar sus datos al sink después de un cierto período de tiempo, y siempre retransmiten el requerimiento recibido a sus nodos vecinos mediante broadcast (Figura 4.1 superior). En este sentido, antes de continuar con la retransmisión del requerimiento, estos nodos reemplazan el contenido original del mensaje, con su propio valor que es mejor. Dado que ΔE es la diferencia entre el valor de umbral T_h transportado en el mensaje de requerimiento y el valor sentido por el nodo, la condición $\Delta E < 0$ determina que existe un valor relevante en el nodo, es decir, que existe un gradiente positivo de información, o ganancia de información, ya que el nodo tiene un mejor valor que el que es transportado en el mensaje. Es por ello que dadas estas condiciones, el nodo decide siempre continuar retransmitiendo el requerimiento a los nodos vecinos. Sin embargo, si su nodo vecino tiene un mejor valor que él, este último termina seleccionado para responder, y mediante la diseminación por broadcast del requerimiento, termina desactivando la respuesta del primer nodo, y así sucesivamente.

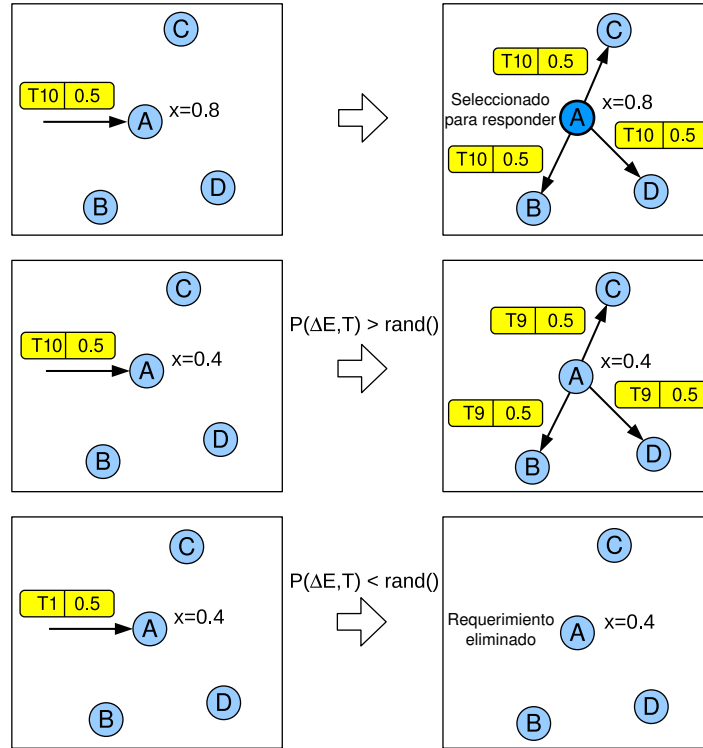


Figura 4.1: Tipos de comportamiento de los nodos durante la diseminación de requerimientos.

4.2.1.2. Filtrado del propio Nodo, posible Filtrado de Vecinos

Por otro lado, si se cumple la condición $\Delta E > 0$, quiere decir que el nodo no tiene datos útiles para la aplicación (filtrado propio) y solo puede retransmitir el mensaje de requerimiento a sus vecinos a fin de explorar más allá (diversificación) con una probabilidad P calculada como:

$$P(\Delta E, T) = e^{-\frac{\Delta E}{T}}. \quad (4.2)$$

donde T es la variable temperatura del algoritmo SA, el cual es transportado dentro del requerimiento (Figura 4.1 centro). El nodo retransmitirá el mensaje a sus nodos vecinos solo si el valor de P calculado es mayor que un valor entre 0 y 1 generado aleatoriamente, es decir, si se cumple la condición $P(\Delta E, T) > \text{rand}()$.

4.2.1.3. Filtrado del propio Nodo y de Nodos Vecinos

Cuanto más bajo el valor de T o más grande la diferencia ΔE , menor es la probabilidad P de continuar retransmitiendo el requerimiento a los nodos vecinos, debido a que la temperatura de SA dentro del mensaje es baja o los datos cercanos al nodo son probablemente irrelevantes para la aplicación, respectivamente. Como consecuencia, el mensaje de requerimiento es probablemente descartado por el nodo (Figura 4.1 inferior), debido a que se cumple la condición $P(\Delta E, T) < \text{rand}()$. De esta manera se filtran o descarta la participación de los nodos vecinos en la computación de una función.

Como conclusión, se puede notar que existe un compromiso entre *diversificación* e *intensificación* en la búsqueda de valores sensados relevantes para la aplicación. El primer término se refiere a la capacidad de exploración del protocolo, mientras que el segundo se refiere a la capacidad de dirigir la búsqueda solo a zonas adonde hay información útil. En este sentido, un valor grande de T fomenta la exploración de datos y disminuye el filtrado de nodos vecinos, en lugar de intensificar la búsqueda a solamente datos relevantes, es decir, incrementa la probabilidad P de retransmitir los requerimientos aunque existan condiciones desfavorables de ganancia de información. En nuestra propuesta, un valor grande de $T = T_0$ es inicialmente utilizado a fin de poder explorar toda la red y no cometer errores en el computo de la función requerida (*diversificación*), y luego, durante la diseminación del requerimiento, es linealmente decrementado por un factor D por nodos intermedios que poseen datos irrelevantes, es decir, que el mensaje va perdiendo temperatura (se va enfriando), energía, o crédito solo cuando se producen condiciones desfavorables para la aplicación (es decir, para $\Delta E > 0$). Este proceso de decremento de temperatura es conocido como *enfriamiento adaptativo*.

En las siguientes iteraciones, un decremento en forma logarítmica de T_0 es aplicado a los requerimientos a ser transmitidos a la red, con el fin de acelerar la convergencia del algoritmo. Sin embargo, el proceso de enfriamiento (*intensificación*) es ejecutado solamente al comienzo de cada iteración basado en un mecanismo de aprendizaje por refuerzo que se detallará más adelante. El valor de D es computado por el nodo sink, como $\frac{T_0}{maxDist}$, donde $maxDist$ es la máxima distancia estimada desde el nodo sink hasta el nodo mas distante de la red, en cantidad de saltos. Esto es debido al hecho de que podría existir una probabilidad no nula de explorar los bordes de la red aunque se asuman datos irrelevantes para la aplicación, es decir, es requerido que $T > 0$ después de atravesar una cantidad de saltos $maxDist$. Nótese que típicamente D se decrementará con el tamaño de la red. Sin embargo, otras estrategias de enfriamiento pueden ser consideradas. Una descripción detallada del algoritmo de retransmisión de PASA es mostrado en el Algoritmo 1, en el cual la Ecuación 4.2 es el corazón del mismo.

4.2.2. Reporte de Respuestas

El proceso de enrutamiento de las respuestas de los nodos seleccionados en la fase de diseminación de requerimientos para reportar sus datos, descrito en el Algoritmo 2, se inicia estratégicamente desde los nodos que se encuentran a mayor distancia del nodo sink. Es decir, los nodos que quedaron seleccionados para responder comienzan con este proceso un tiempo inversamente proporcional a su distancia en saltos respecto al sink. En este sentido, los nodos más alejados comienzan primero a enviar sus datos. Esta estrategia es utilizada para filtrar la respuesta de nodos previamente seleccionados a responder que se encuentran mas cercanos al sink.

Sin embargo, se han desarrollado dos variantes del mecanismo de respuesta a lo largo del desarrollo de esta Tesis. En la primera versión de PASA (Figura 4.2a) se utiliza un esquema de envío de respuestas por múltiples caminos hacia el sink. Un nodo seleccionado para responder hace un broadcast del mensaje, y lo retransmiten solo aquellos nodos que tengan una menor distancia respecto al sink. Con esta estrategia se pueden direccionar fácilmente los datos relevantes hacia el sink, teniendo como ventaja, una gran robustez ante pérdida de paquetes, o de fallas en nodos, debido a que se transmite la misma información por múltiples caminos. Además, se obtiene una gran capacidad de filtrado en el proceso de respuesta de nodos pre-seleccionados a responder. Sin embargo, el costo de comunicación

Algorithm 1: Algoritmo de decisión de diseminación de requerimientos del protocolo sin estados (PASA)

```

1: Input: Mensaje de requerimiento recibido en un nodo
2: Output: Decisión de si el nodo responde y retransmite el requerimiento
3:  $id \leftarrow getId(Msg)$ 
4: ...
5:  $T \leftarrow getTemperature(Msg)$ 
6:  $thresholdValue \leftarrow getThreshold(Msg)$ 
7:  $\Delta E \leftarrow thresholdValue - sensedValue$ 
8: // Descarta requerimientos ya enviados
9: if  $idTable[id] == true$  then
10:    $delete(Msg)$ 
11: else
12:    $updateIdTable(id)$ 
13: end if
14: // Decide envío de nuevas consultas
15: if  $\Delta E \leq 0$  then
16:    $Msg \leftarrow setValue(sensedValue)$ 
17:    $Msg \leftarrow setTemperature(T)$ 
18:    $send(Msg)$ 
19: else
20:   if  $P = e^{-\Delta E/T} > rand()$  then
21:      $Msg \leftarrow setTemperature(T - D)$     D: Decremento
22:      $send(Msg)$ 
23:   else
24:      $delete(Msg)$ 
25:   end if
26: end if

```

puede ser elevado, principalmente si los nodos a responder se encuentran alejados del nodo sink, y si la densidad de nodos es elevada. Posteriormente, en la segunda versión de PASA, se desarrolló una estrategia diferente de respuesta, más eficiente energéticamente, en la que cada nodo almacena durante la fase de diseminación de requerimientos, desde que nodo vecino recibió primero dicho mensaje, el cual representa el camino de menor latencia (Figura 4.2b).

4.2.3. Aprendizaje

Debido a que el protocolo PASA implementa las fases de requerimiento de información y de respuesta de forma iterativa, este esquema permite obtener una vista actualizada o un muestreo del estado de los nodos de la red. Esto permite hacer frente a nodos que pueden fallar previo a una iteración de cómputo. En caso de falla dentro de una iteración, una estrategia simple de auto-curado ha sido incluida para hacer frente a este problema, la cual es descrita en detalle en la Sección 6.6.4.

El esquema de aprendizaje implementado en este protocolo es de tipo *centralizado*, y es ejecutado solo en el nodo sink. Dicho aprendizaje es descripto a continuación.

- Para un valor grande de T_0 , la retransmisión en bajada puede comportarse como Flooding, debido a que cada nodo retransmitirá siempre el requerimiento a sus nodos vecinos, independientemente de la distribución de los datos en la red; es decir, existe certeza de encontrar siempre los datos más relevantes pero a mayores costos de comunicación.
- Para un valor bajo de T_0 , el comportamiento de la retransmisión de los requerimientos es equiva-

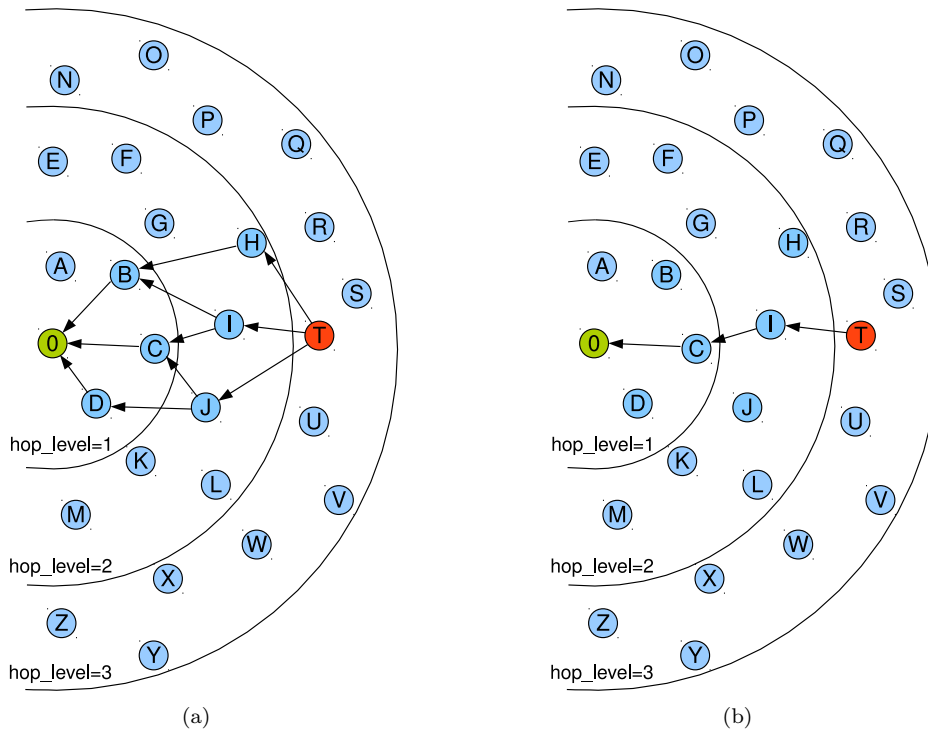


Figura 4.2: Mecanismos de reporte de datos: por múltiples caminos (a), y por camino único (b).

lente al comportamiento del algoritmo gradient descent, donde el requerimiento es retransmitido solamente a través de nodos con algunos datos relevantes, o a través de un gradiente marcado. Como resultado, la probabilidad de encontrar datos significativos es baja, lo que puede introducir errores de computación.

Estas características son descriptas en la Figura 4.3 para el caso de un escenario con 10 fuentes de datos, considerando diferentes valores de T_0 . En dicha figura, se puede ver el impacto en términos del error de computación, la probabilidad de éxito en computar el valor actual de la función, y el costo de diseminar el requerimiento.

De este análisis, se puede concluir que existe algún valor T_0 , el cual puede proveer un error de computación nulo o bajo, así como también costos de comunicación bajos. Por lo tanto, es necesario encontrar un valor T_0 óptimo para inicializar T tal que cuando sea linealmente decrementado este ofrezca un buen equilibrio entre costos de comunicación y errores de computación. Para este propósito, un algoritmo de aprendizaje por refuerzo es implementado en el nodo sink. El objetivo de este algoritmo es que el nodo sink puede aprender de la distribución de datos en la red basado en su experiencia al recibir los mensajes de respuesta de los nodos. Esta estrategia es implementada basada en el *id* de los nodos que responden al requerimiento de información. El nodo sink puede aplicar una política de actualización de temperatura para adaptar el valor de $T_0(i)$ en cada iteración sucesiva de diseminación de requerimientos, para mejorar el proceso de búsqueda, reduciendo el consumo de energía. El objetivo del nodo sink es reducir la profundidad de búsqueda, o lo que es equivalente, reducir el área de diseminación de requerimientos, o incrementar el filtrado de nodos vecinos, iterativamente mediante la actualización del valor de $T_0(i)$. Con esto, se puede llegar con los requerimientos solamente a aquellos nodos que provean de argumentos relevantes al nodo sink, para computar la función requerida. Este

Algorithm 2: Algoritmo completo de decisión de diseminación de requerimientos y de reporte de datos del protocolo sin estados (PASA)

```

1: Input: Paquete recibido en un nodo
2: Output: Decisión completa del comportamiento del nodo
3:  $id \leftarrow getId(Msg)$ 
4:  $type \leftarrow getType(Msg)$ 
5:  $hops \leftarrow getHops(Msg)$ 
6: if  $type == downstream$  then
7:   // Actualiza distancia de hop mínima hasta el nodo coordinador
8:   if  $hops < minHops$  then
9:      $minHops \leftarrow hops$ 
10:  end if
11:   $Msg \leftarrow setHops(minHops + 1)$ 
12:  ... (Algoritmo 1: Algoritmo de diseminación de requerimientos) ...
13:  // Actualiza el estado de la respuesta
14:  if  $\Delta E \leq 0$  then
15:     $replyQuery \leftarrow true$ 
16:  else
17:     $replyQuery \leftarrow false$ 
18:  end if
19: else
20:   // Descarta respuestas ya recibidas
21:   if  $idTable[id] == true$  then
22:      $delete(Msg)$ 
23:   else
24:      $updateIdTable(id)$ 
25:   end if
26:   // Decide envío de nuevas respuestas
27:   if  $hops > minHops$  then
28:      $Msg \leftarrow setHops(minHops - 1)$ 
29:      $send(Msg)$ 
30:   else
31:      $delete(Msg)$ 
32:   end if
33: end if

```

proceso de aprendizaje es descrito en la Figura 4.4.

Si bien los nodos se encuentran desplegados sobre un área A, como es ilustrado en la figura, una primera iteración de requerimiento de información alcanza a nodos que se encuentran sobre el área B. Nodos sobre el área $(A - B)$ no son consultados debido a que no fueron encontrados datos relevantes en las proximidades, y por lo tanto el valor de la probabilidad de retransmisión P se vuelve muy pequeña. En la segunda iteración, el valor inicial $T_0(i)$ es decrementado hasta un valor $T_0(i + 1)$, el cual reduce el espacio de búsqueda a un área C. Este proceso es repetido mientras el algoritmo encuentre el valor de $T_0(i)$, en alguna iteración i , que habilite consultar a todos los nodos con datos relevantes al menor costo de comunicación posible, el cual en la Figura 4.4 es representado por el área E. A cada iteración de computación, el nodo sink toma la decisión de si decrementar el valor previo $T_0(i - 1)$ o no. Posterior al envío del requerimiento en la primera iteración, con un valor T_0 elevado, en la primera versión del protocolo PASA desarrollado, el nodo sink almacena el número de respuestas (es decir, el número de nodos que reportan datos relevantes). Mientras el nodo sink reciba el mismo número de respuesta, este asume que el valor previo $T_0(i - 1)$ puede ser reducido, es decir, el nodo sink decrementa el valor $T_0(i)$ inyectado en la siguiente iteración i . Un valor bajo de T_0 tiende a reducir el espacio de búsqueda de información en la red, ahorrando energía en los nodos. En el caso de que

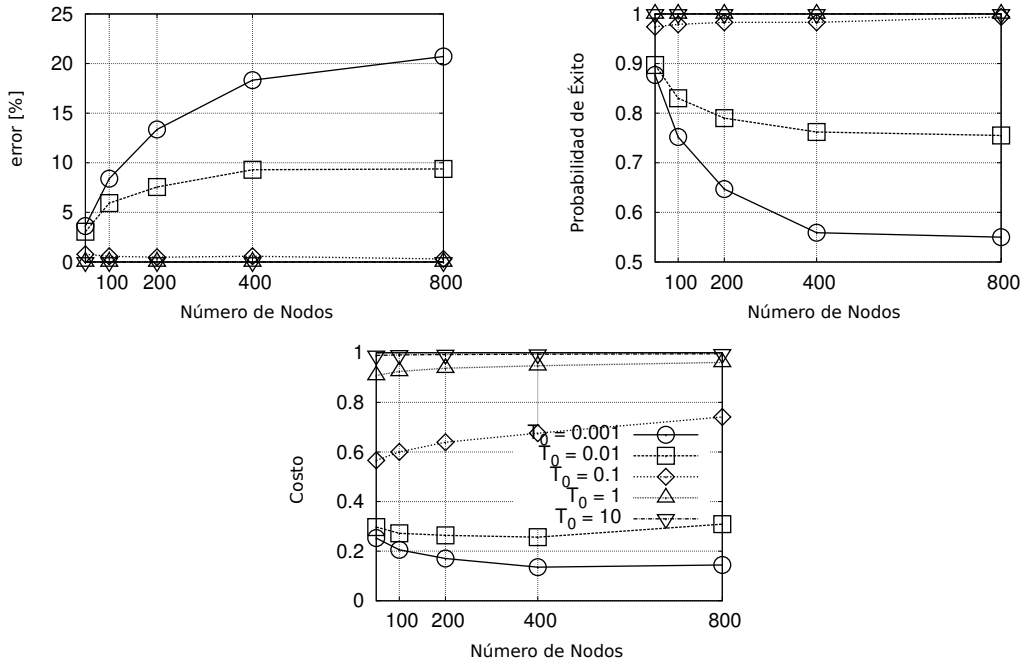


Figura 4.3: Performance del protocolo sin estados para diferentes valores de T_0 .

el número de respuestas disminuye luego de cierta iteración ($i - 1$), el valor $T_0(i)$ en la iteración i es incrementada a un valor intermedio entre $T_0(i - 2)$, la última vez que no hubo pérdida de información, y $T_0(i - 1)$, la última vez sin pérdida de información. Por ejemplo, $T_0(i - 2) = 100$, $T_0(i - 1) = 10$, $T_0(i) = 45$ muestra este concepto.

Una descripción de los procesos de enfriamiento logarítmico (entre cada iteración) y lineal (dentro de una iteración) es esquematizada en la Figura 4.5 para el caso de computación de la función máximo. El nodo sink es indicado como nodo s , y un nodo seleccionado a reportar su argumento al sink como nodo n . Los mensajes de requerimiento embeben tanto el valor de temperatura actual T , como el valor de umbral T_h . Después de i iteraciones, T_0 converge a un valor fijo $T_0(i)$ el cual es usado en requerimientos sucesivos, como se muestra en la Figura 4.6 para el caso de un valor de $T = 0,02804688$.

Nótese que si el valor de T_0 es decrementado mas allá de un valor dado, el error de computación se incrementa debido a que el espacio de búsqueda es significativamente reducido, y la retransmisión de requerimientos se limita solo a áreas con gradientes de datos (es decir, $\Delta E \leq 0$). Este último efecto puede notarse por el número de saltos malos que se producen en la red durante la propagación de un requerimiento, que es definido como el número promedio de requerimientos retransmitidos por cada nodo cuando los datos son considerados no relevantes ($\Delta E > 0$). A diferencia de implementaciones tradicionales del algoritmo SA, no se define un criterio o condición de detención del algoritmo PASA. Tanto un valor bajo de T_0 o un límite en el número de iteraciones son normalmente utilizados como condiciones de detención en las soluciones basadas en SA propuestas. En nuestro protocolo, es posible bajo ciertas condiciones que $T_0(i) \rightarrow 0$ para valores grandes de i , por ejemplo, en el caso de la detección de un evento único en la red. Debido a que tanto la topología de red como los datos obtenidos por los sensores pueden variar en el tiempo, el proceso de aprendizaje puede ser ejecutado periódicamente para adaptar la diseminación de requerimientos a esas variaciones. En esta versión actualizada del protocolo PASA se aplica una regla de decisión mas robusta de la política de actualización de temperatura por

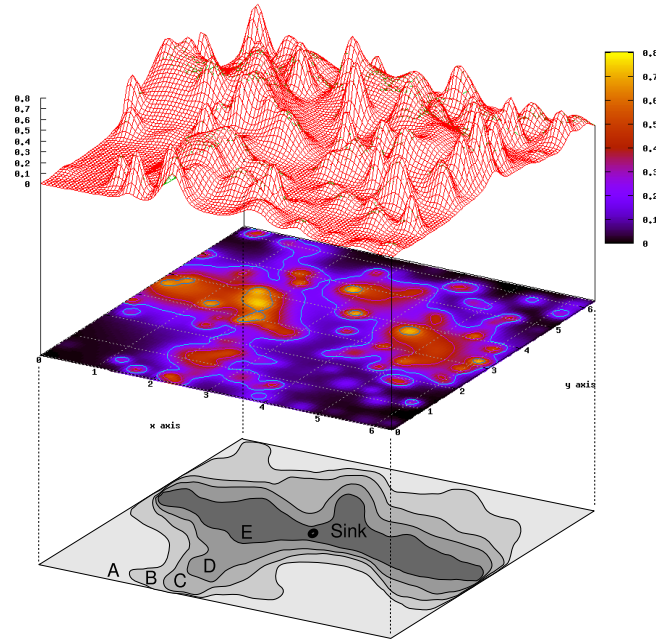


Figura 4.4: Filtrado iterativo con reducción del espacio de búsqueda desde A hasta E.

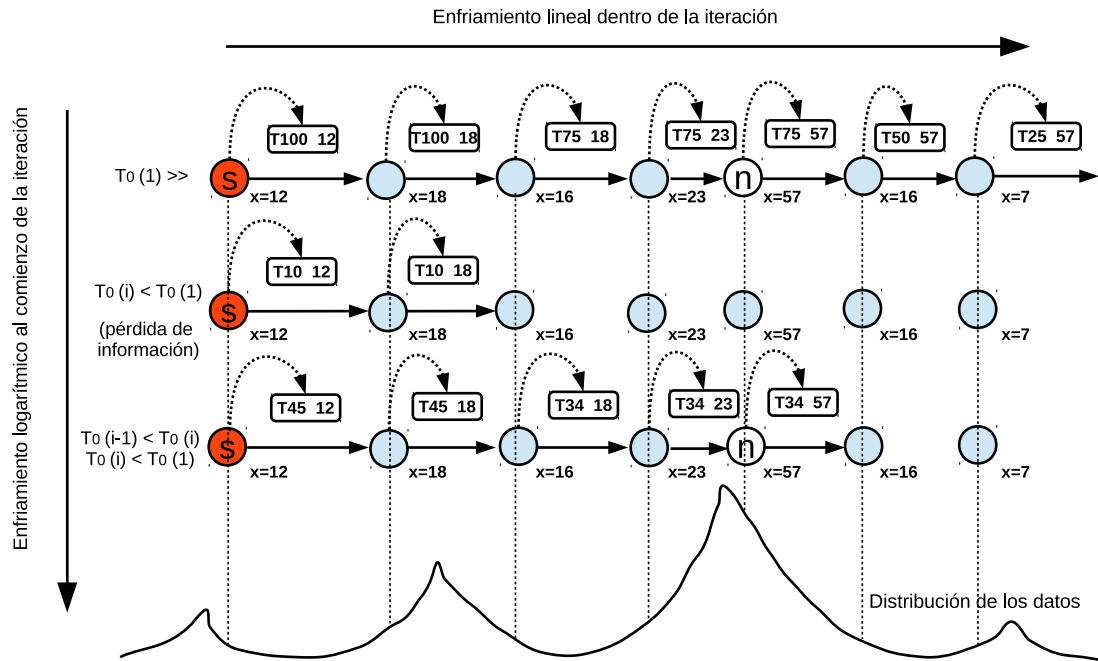


Figura 4.5: Procesos de enfriamiento utilizados por el protocolo PASA.

parte del nodo sink, respecto a la versión previa de PASA. En la primer versión del protocolo PASA, se consideraba que el nodo sink aprovecha el número de respuestas \hat{X}' para aplicar la política de adaptación de T . Sin embargo, en la segunda versión, el nodo sink aplica una política de actualización de temperatura basada en el *id* de aquellos nodos sensores que reportan sus argumentos en la primer iteración (que es la más profunda). En este sentido, el nodo sink decrementa el valor de temperatura T de los mensajes de requerimientos inyectados en la red, y lo incrementa en caso contrario, para evitar pérdida de información.

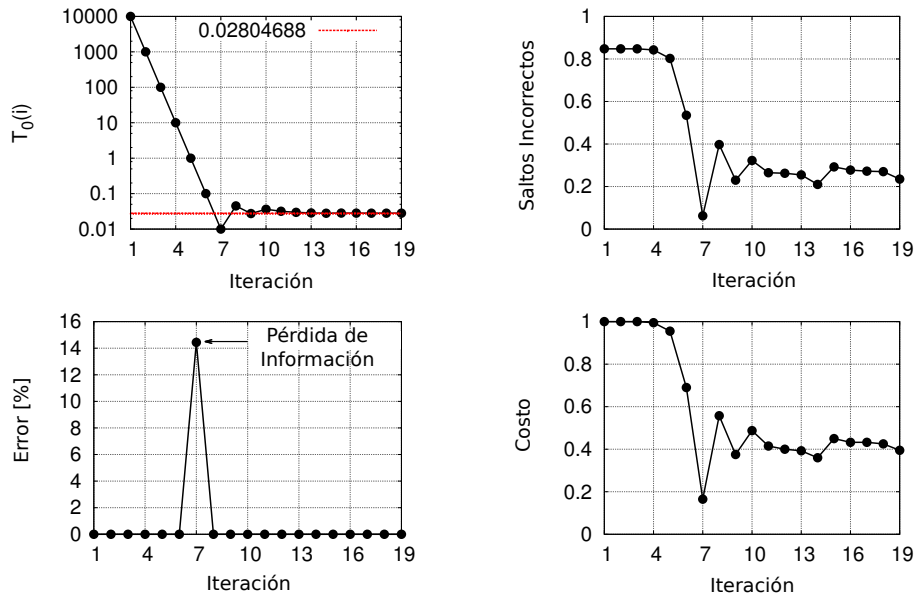


Figura 4.6: Convergencia del protocolo sin estados (para 400 nodos, y 10 fuentes de datos).

4.3. Comentarios de Cierre

En este capítulo se introdujo al funcionamiento del protocolo de filtrado de nodos vecinos sin estados, basado en una versión multi-instanciada del algoritmo recocido simulado, denominado protocolo PASA. En este contexto, se describieron sus fases de funcionamiento, conformadas por la diseminación de requerimientos, por el reporte de las respuestas, y por el aprendizaje, y se describieron los detalles de implementación. Además, se especificaron las estrategias de enfriamiento, tanto lineales dentro de una iteración de cómputo, como de logarítmicas entre cada iteración, desarrolladas en esta Tesis para agilizar el proceso de convergencia del filtrado, ya que esta es una de las desventajas del algoritmo de recocido simulado. Se analizó el comportamiento de la búsqueda de información y del filtrado de datos y de nodos respecto a diferentes valores de temperatura de búsqueda T_0 . Se pudo determinar que es necesario encontrar un valor del parámetro temperatura del protocolo propuesto que permita obtener un equilibrio, o que garantice minimizar tanto el costo de búsqueda como el error de computación. Para ello, se implementó en el nodo sink un mecanismo de aprendizaje por refuerzo que permite cumplir con este objetivo, independientemente de las características del escenario considerado (parámetro físico y red).

Protocolo de Procesamiento con Estados

Contenido

5.1	Introducción	77
5.2	Funcionamiento	78
	5.2.1 Diseminación de Requerimientos	80
	5.2.2 Reporte de Respuestas	80
	5.2.3 Aprendizaje	82
5.3	Comentarios de Cierre	83

5.1. Introducción

EL segundo protocolo de procesamiento y filtrado, desarrollado en esta Tesis, se denomina *protocolo con estados* [42]. Este nombre se debe a que se mantienen estados en cada uno de los nodos de la red, los cuales están relacionados con tareas de procesamiento y filtrado ejecutadas previamente, y que son utilizados por cada nodo para definir su funcionamiento.

El objetivo de este protocolo, al igual que en el caso del protocolo sin estados, es el de realizar procesamiento y filtrado dentro de la red, de aquellos datos y nodos vecinos que no son útiles para la aplicación, con el fin de reducir el elevado costo de comunicación al reportar al sink todos los datos obtenidos por los nodos. Sin embargo, en este protocolo, la decisión de diseminar mensajes de requerimientos para realizar la computación de una cierta función de los datos sensados, y el aprendizaje y control, o toma de decisiones, es manejada de forma local por cada nodo. Esta es la principal diferencia con respecto al protocolo sin estados, en el cual esta decisión era impuesta por el nodo sink a través de la temperatura del mensaje de requerimiento. En resumen, en el protocolo con estados, los nodos sensores que reciben las tareas de requerimiento de computación actúan en función de reglas de decisión simples basadas en estados locales que son función de tareas ejecutadas previamente. Este protocolo, al igual que el protocolo sin estados, utiliza dos fases de funcionamiento

ejecutadas de forma secuencial, conformadas por la diseminación de requerimientos a la red, y por el reporte de respuestas hacia el nodo sink. Ambas fases constituyen una iteración de procesamiento, como se describió en la Sección 3.7. En este contexto, se realiza parte del procesamiento y filtrado dentro de la red, y la información resultante es provista al nodo sink, quién es el que realiza la etapa final de procesamiento para obtener el resultado buscado. Es decir, se busca preprocesar los datos dentro de la red, filtrando información y la participación de nodos no relevantes para la aplicación. Sin embargo, al final de cada iteración, cada nodo utiliza un algoritmo de aprendizaje para actualizar su estado local. En resumen, la decisión de diseminación de requerimientos en la red, es determinada en forma local por cada nodo, no existiendo un control centralizado, como en el caso del protocolo PASA (Capítulo 4).

Afortunadamente, en los últimos años, diferentes protocolos de enrutamiento bio-inspirados han emergido como una eficiente y robusta solución. Ellos están basados en sistemas biológicos, los cuales son el resultado de millones de años de evolución, con características tales como *adaptabilidad a condiciones de entorno adversas, resistencia a fallas y daños, operación colaborativa, y auto-organización*. Observando a la naturaleza como una fuente de inspiración, podemos desarrollar nuevas estrategias y hacer frente a los problemas que existen en las redes de comunicaciones inalámbricas de gran escala, en las cuales *la ausencia de un control centralizado y la resolución desatendida de posibles fallas son factores claves* [38, 74]. La *inteligencia en enjambre o de colonias (swarm intelligence, SI)*, es un campo de investigación relativamente nuevo, el cual estudia la conducta colectiva de sistemas multi-agente que se auto-coordinan mediante algún tipo de control descentralizado (auto-organización) para realizar alguna tarea. En este tipo de inteligencia distribuida y colaborativa, un individuo particular no tiene la capacidad de realizar una acción eficientemente. Sin embargo, cada miembro, usando algún tipo de comunicación directa o indirecta, contribuye a la ejecución de una tarea específica de forma eficiente. Este fenómeno, que se puede encontrar en los sistemas biológicos, se conoce como *estigmergía*.

5.2. Funcionamiento

Este protocolo de procesamiento y filtrado, denominado *Pheromone-based In-Network Processing (PhINP)*, implementa una versión modificada de la metaheurística basada en colonia de hormigas (ant colony optimization, ACO). El protocolo PhINP utiliza un modelo de comunicación por broadcast, y combina las ventajas de los algoritmos basados en feromona (derivados de ACO) y de aprendizaje automático instanciados en cada uno de los nodos o agentes que conforman la red. Por simplicidad, en esta Tesis se analiza la computación de funciones máximo. Se enfatizan la robustez y la adaptabilidad del mecanismo de enrutamiento debido a la variabilidad de condiciones existentes en las WSNs (por ejemplo, nodos que fallan, introducción de nuevos nodos, pérdida de paquetes, etc.). Otra motivación es la de desarrollar esquemas cooperativos o colaborativos que permitan descentralizar el control del nodo sink presente en el protocolo sin estados PASA.

Los protocolos de enrutamiento basados en feromona imitan, por ejemplo, la conducta cooperativa de colonia de hormigas durante la búsqueda de fuentes de alimento. Para llevar adelante esta tarea, las hormigas realizan la exploración de su entorno, depositando feromona a través de los caminos que recorren. Cuando una hormiga encuentra una fuente de alimento, retorna a su colonia, reforzando los caminos pre-establecidos con más cantidad de feromona. Otras hormigas tenderán a seguir más probablemente estos caminos, reforzándolos con más feromona. En el transcurso del tiempo, la feromona

tiende a evaporarse, eliminando caminos no usados. Para evitar la estagnación, que es el problema que surge cuando todas las hormigas tienden a seguir los mismos caminos, perdiendo la capacidad de explorar y descubrir nuevas fuentes, se considera una cierta probabilidad de exploración de nuevos caminos. El enfoque de búsqueda basado en feromona ha inspirado varios mecanismos de enrutamiento de datos, los cuales pueden ser clasificados bajo tres criterios, en base a:

- *como se inicia el reporte de datos.* Existen tres enfoques, definidos como: *proactivos* [82, 83] (ante un evento), *reactivos* [84] (bajo demanda o por requerimiento), e *híbridos* (combinación de los anteriores). En general, los esquemas reactivos tienen mejor escalabilidad que los proactivos, debido a que requieren menor overhead de control, y son más adecuados para escenarios muy dinámicos.
- *cual es el modelo de comunicación.* Normalmente se consideran esquemas de comunicación *unicast*, aunque también modelos *broadcast* e híbridos han sido propuestos [84]. La comunicación unicast no aprovecha la naturaleza de la comunicación inalámbrica, en la que un paquete es recibido por todos los nodos vecinos dentro del rango de comunicación del transmisor. Además, el modelo broadcast permite realizar el enrutamiento por múltiples caminos, incrementando la robustez del protocolo ante la pérdida de paquetes y falla de nodos.
- *quién es el iniciador del proceso.* Tres enfoques han sido propuestos: *iniciado por los nodos sensores* [82, 85, 98, 86], *iniciado por el nodo sink* [87], e híbridos.

En esta Tesis se propone un protocolo de dos fases iniciado por el nodo sink, y bajo demanda, y que utiliza comunicación por broadcast. El protocolo toma ventaja de los niveles de feromona de los nodos de la red para dirigir los mensajes de requerimiento de información a nodos con información relevante, filtrando la participación de nodos no útiles para la aplicación. Las principales diferencias de este protocolo propuesto con respecto a los protocolos existentes son:

- Exploración, disseminación de requerimientos, computación de la función de interés, y filtrado integrados en la misma fase.
- El nivel de feromona no es ni transportado en el mensaje de requerimiento ni intercambiado entre nodos vecinos, sino que es un parámetro local de cada nodo.
- El nivel de feromona es un valor simple representando la probabilidad de que un nodo continúe retransmitiendo un mensaje de requerimiento. En los protocolos tradicionales, se utiliza una *matriz o un vector de feromonas* con información de los niveles de feromona entre todos o algunos nodos vecinos.
- Se utiliza un enfoque de enrutamiento o disseminación de requerimientos basado en datos y no basado en direcciones, el cual es más adecuado para realizar procesamiento en red.
- No se incluye en el mensaje de requerimiento información de los nodos sensores ya visitados.

El protocolo PhINP, al igual que el protocolo PASA, utiliza dos fases de funcionamiento, disseminación de requerimientos a la red, y reporte de respuestas hacia el nodo sink. Ambas fases constituyen una iteración de procesamiento. Al final de una iteración, se produce la actualización del nivel de feromona en cada nodo. Estas fases son detalladas en las siguientes subsecciones.

5.2.1. Diseminación de Requerimientos

Esta fase está basada en un proceso de búsqueda iterativa con reducción del espacio de búsqueda, adaptando la diseminación de requerimientos a la distribución de los datos. Esta fase combina eficientemente dos estrategias, la primera es búsqueda basada en feromona, para dirigir los requerimientos a nodos con valores sensados relevantes, y la segunda filtrado en red, para limitar la participación de nodos no relevantes, como es descrito en el Algoritmo 3. El mensaje de requerimiento es conformado por:

- tipo de mensaje (requerimiento o respuesta).
- identificación del nodo que origina el mensaje.
- valor de umbral (por ejemplo, valor máximo).
- Número de saltos realizados.

Vale la pena mencionar que, con el fin de computar la función máximo, los nodos necesitan actualizar el valor de umbral del mensaje de requerimiento si su valor sensado es mayor (umbral dinámico). Para reducir el costo de comunicación, cada nodo puede retransmitir el mismo mensaje solamente una vez. En la recepción de un mensaje de requerimiento, cada nodo debe tomar dos decisiones:

- si responder o no al requerimiento mediante el reporte de los datos al nodo sink (auto-filtrado)
- si continuar o no retransmitiendo el requerimiento a sus nodos vecinos (filtrado de vecinos)

La primera decisión es basada tanto en el valor de umbral transportado en el mensaje de requerimiento, como en el valor sensado en el nodo que recibe el mensaje, y su determinación no presenta complejidad. Sin embargo, la segunda decisión es basada en el nivel de feromona del nodo, y puede conducir a cometer algún tipo de error en el proceso de cómputo, que puede ser remediado luego de algunas iteraciones adicionales para readaptar la búsqueda, y que será descrito más adelante. Esta última situación se puede producir en el caso de que el protocolo ya haya adaptado la diseminación y filtrado a un escenario determinado, y que aparezcan, a partir de un momento, nuevas fuentes de datos.

5.2.2. Reporte de Respuestas

Al final de la primera fase de diseminación de requerimientos, considerando que la misma abarca a la totalidad de la red (por inundación), cada nodo aprende cuál es su distancia en saltos de comunicación respecto al nodo sink (hop level), actualizando el mínimo valor de saltos de los mensajes de requerimiento recibidos en él a través de diferentes caminos. La fase de reporte de datos o de respuesta es iniciada por aquellos nodos que quedaron seleccionados durante la fase de diseminación de requerimientos, considerando que el tiempo de respuesta es inversamente proporcional a la distancia del nodo respecto al sink. Esta estrategia permite deshabilitar o filtrar aquellos nodos más cercanos al sink seleccionados previamente en la misma iteración para responder. En este contexto, el reporte de datos es realizado utilizando comunicación por unicast, basado en el camino inverso de primer arribo del mensaje de requerimiento, el cual, representa el camino de menor latencia.

Algorithm 3: Algoritmo de decisión de retransmisión de requerimientos del protocolo con estados (PhINP)

```

1: Input Nodo recibe mensaje de requerimiento
2: Output Se determinan las decisiones de respuesta y de retransmisión del requerimiento
3:  $id \leftarrow Msg.getId()$ 
4:  $thresholdValue \leftarrow Msg.getThreshold()$ 
5:  $\Delta E \leftarrow thresholdValue - sensedValue$ 
6: // Discard already forwarded queries
7: if  $idTable[id] == true$  then
8:    $delete(Msg)$ 
9: else
10:   $updateIdTable(id)$ 
11: end if
12: // Query forwarding decision
13: if  $\Delta E \leq 0$  then
14:   $Msg \leftarrow setValue(sensedValue)$ 
15:   $send(Msg)$ 
16: else
17:  if  $\lambda > rand()$  then
18:     $Msg \leftarrow setValue(thresholdValue)$ 
19:     $send(Msg)$ 
20:  end if
21: else
22:   $delete(Msg)$ 
23: end if

```

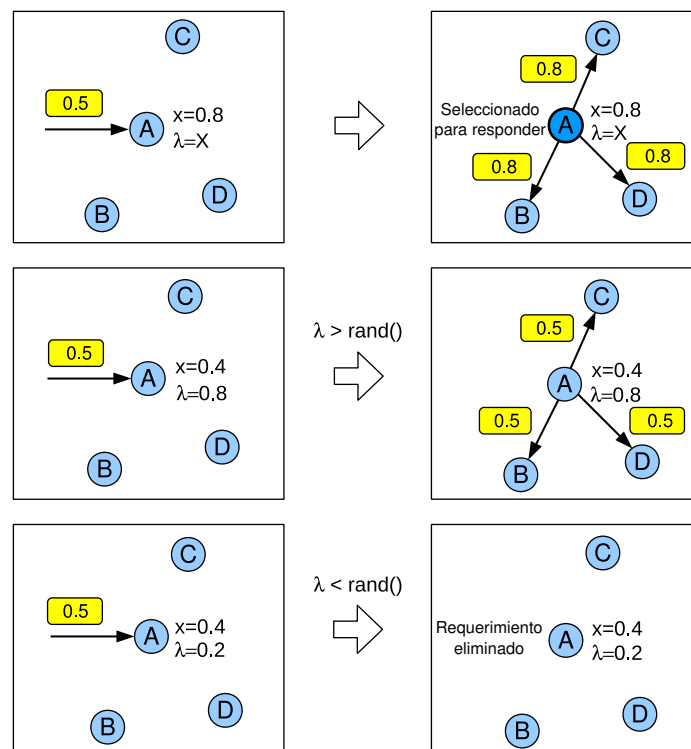


Figura 5.1: Tipos de comportamiento de los nodos durante la disseminación de requerimientos.

Algorithm 4: Política de actualización de feromona

```

1: if  $\lambda > \lambda_{min}$  then
2:    $\lambda = \lambda - \lambda_{dec}$ 
3: end if
4: if answerQuery == true or forwardResp == true then
5:   if  $\lambda < \lambda_{max}$  then
6:      $\lambda = \lambda + \lambda_{inc}$ 
7:   end if
8: end if

```

5.2.3. Aprendizaje

Al final de cada iteración de computación, cada nodo sensor actualiza su nivel de feromona, bajo ciertos criterios que son descritos a continuación. En este sentido, la frecuencia de actualización del nivel de feromona en cada nodo es definida por el proceso de diseminación de requerimientos, y es en general asumido igual a la frecuencia de monitoreo. El primer mensaje de requerimiento incentiva o estimula a todos los nodos a participar en esta fase, mediante la configuración de los niveles máximos de feromona ($\lambda = 1$) en los nodos. Esto garantiza que todos los nodos reciben el requerimiento (por inundación o flooding), y que todos los nodos con lecturas relevantes son adecuadamente seleccionados. Después de cada iteración, osea, después de finalizadas las fases de diseminación de requerimiento y reporte de datos, todos los nodos decrementan su nivel de feromona un nivel λ_{dec} (evaporación) hasta un valor mínimo permitido dado por λ_{min} . Solo aquellos nodos que han sido seleccionados para responder el requerimiento reportando sus datos sensados, o que han participado en el proceso de retransmisión de datos de nodos vecinos hacia el sink, pueden incrementar su nivel de feromona un nivel λ_{inc} hasta un límite máximo dado por λ_{max} (tal como en el caso de MAX-MIN Ant System [88]). En el caso de PhINP, el valor de λ_{max} es limitado a 1, y el valor de λ_{min} puede ser ajustado basado en los requerimientos de la aplicación particular. En este último caso, existe un balance entre el costo de diseminación de requerimientos y el error de computación, debido a que para valores bajos de λ_{min} el costo tiende a ser bajo pero el error se incrementa, y viceversa. El objetivo de esta fase es reforzar aquellos caminos formados entre nodos con argumentos útiles para computar la función requerida, y el nodo sink, como es descrito en el Algoritmo 4.

En general, se utiliza en las simulaciones realizadas a lo largo de esta Tesis, un valor de $\lambda_{min} = 0,3$, que posibilita obtener un filtrado aceptable de nodos en la red, mientras que se minimiza el error de cómputo al surgir nuevas fuentes de datos en la red. La distribución de los niveles de feromona presentes en cada nodo de la red, luego de la convergencia del protocolo PhINP a un escenario determinado, es mostrado en la Figura 5.2. En esta figura se puede ver que los nodos que participan en el reporte de los argumentos relevantes para la aplicación, es decir, que reportan valores máximos al sink para el cómputo del valor máximo en la red, mantienen un nivel máximo de feromona igual a 1. Es decir, que siempre retransmiten el requerimiento. Por el contrario, los nodos que quedan excluidos o filtrados de este proceso, mantienen un nivel de feromona mínimo igual a 0.3.

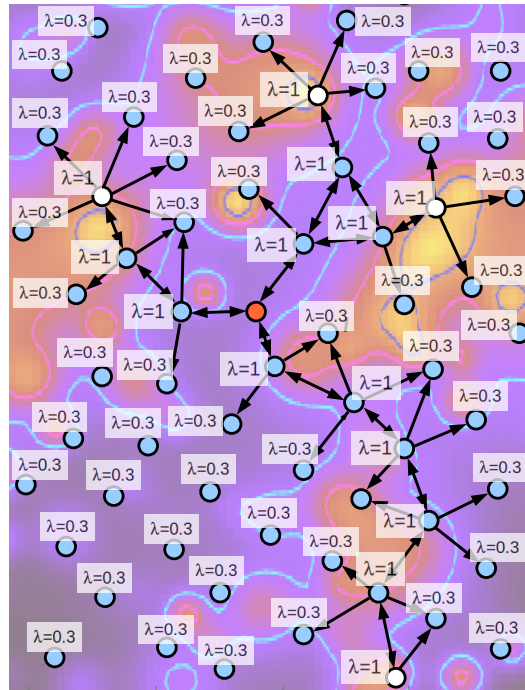


Figura 5.2: Distribución de los niveles de feromona en cada nodo luego de la convergencia del algoritmo, para $\lambda_{min} = 0,3$ y $\lambda_{max} = 1$.

5.3. Comentarios de Cierre

En este capítulo se introdujo al funcionamiento del protocolo de filtrado de nodos vecinos con estados, basado en inteligencia en colonia, denominado protocolo PhINP. En este contexto, se describieron sus fases de funcionamiento, conformadas por la diseminación de requerimientos, por el reporte de las respuestas, y por el aprendizaje, y se describieron los detalles de implementación. Además, se especificaron las estrategias de diseminación de requerimientos y de filtrado de nodos vecinos en función de una variable local a cada nodo denominada feromona (λ). Sin embargo, se definieron los parámetros de control que definen el comportamiento del protocolo, como son λ_{dec} , λ_{inc} , λ_{max} , y λ_{min} . Los dos primeros parámetros definen la velocidad tanto de convergencia como de readaptación del protocolo, mientras que los dos últimos definen los límites máximos y mínimos que puede tomar la variable local λ , y que define totalmente el filtrado de nodos vecinos. Además, se analizó el comportamiento de la búsqueda y del filtrado respecto a diferentes valores de estos parámetros de control, los cuales deben ser definidos para cada tipo de escenario, según la distribución de los datos.

Capítulo 6

Análisis de los Resultados

Contenido

6.1	Métricas Analizadas	86
6.2	Parámetros de Configuración	86
6.3	Protocolos Utilizados para Comparación	87
6.4	Análisis del Protocolo sin Estados	88
6.4.1	Comparación con otros Protocolos	88
6.4.2	Cambio de Amplitud	89
6.4.3	Falla de Nodos	90
6.4.4	Paquetes Perdidos	90
6.5	Análisis del Protocolo con Estados	90
6.5.1	Comparación con otros Protocolos	91
6.5.2	Falla de Nodos	92
6.5.3	Paquetes Perdidos	92
6.5.4	Tiempo de Vida de la Red	92
6.5.5	Monitoreo con Resolución Finita	94
6.6	Análisis de la Comparación entre Protocolos	94
6.6.1	Convergencia	95
6.6.2	Análisis de Eventos	97
6.6.3	Detección de Evento Único	98
6.6.4	Robustez ante Fallas de Nodos	98
6.7	Comparación de los Protocolos	101
6.8	Comentarios de Cierre	101

EN este capítulo se describen los resultados obtenidos del análisis de las diferentes métricas evaluadas de los protocolos de procesamiento y filtrado propuestos en esta Tesis. Los protocolos fueron evaluados en entornos de simulación desarrollado en OMNeT++ [75], mediante los cuales se evaluaron diferentes métricas para comparar las performances de los mismos en diferentes escenarios (Figura 6.1).

6.1. Métricas Analizadas

Las principales métricas analizadas son: *error de computación*, *probabilidad de éxito*, y *costo de comunicación*. El *error de computación* representa el error relativo resultante de la función computada en el nodo sink respecto al óptimo. Por otro lado, la *probabilidad de éxito* representa la probabilidad de computar el valor óptimo (es decir, error relativo cero). En este sentido, en el caso particular del cálculo de la función máximo, esta métrica representa la probabilidad de encontrar el valor máximo obtenido por los nodos de la red. Finalmente, el *costo de comunicación* c_T considera el número promedio de nodos involucrados en el proceso de comunicación, tanto la parte correspondiente a la retransmisión de mensajes de requerimiento (*costo de requerimiento*, c_q), como en el de retransmisión de mensajes de datos hacia el nodo sink (*costo de respuesta*, c_r) en cada iteración de computación. Es decir, se define $c_T = c_q + c_r$. El máximo costo de requerimiento es igual a 1, lo que implica que todos los nodos retransmiten el requerimiento una vez dentro de una iteración, es decir, se hace una inundación o flooding limitada a una sola retransmisión por nodo. Por otro lado, si cada nodo seleccionado para responder envía su medición al sink por un único camino, y si la cantidad de estos nodos es baja, este costo de respuesta es en general despreciable respecto al costo de requerimiento. Por esta razón, este trabajo se enfoca en la reducción de este último costo, es decir, del c_q . Además, este parámetro es considerado como *comunicación* en todos los análisis en esta Tesis. Bajo la asunción de que cada nodo sensor puede retransmitir solo una vez el requerimiento en cada iteración, el costo de comunicación asociado puede ser también definido como el número de paquetes transmitidos en cada iteración. En este sentido, esta métrica está estrechamente relacionada a la energía consumida por los nodos de la red.

Adicionalmente, se han realizado análisis del *tiempo de vida de la red*, lo que permite determinar las ventajas de usar un protocolo respecto a los demás. El tiempo de vida se puede definir de diferentes formas. La más comunmente utilizada es la duración en tiempo de funcionamiento de la red hasta que el primer nodo queda sin energía. Otra variante es la duración de tiempo hasta que un porcentaje dado de nodos queda sin energía, o inversamente, la duración de tiempo en la que un porcentaje específico de nodos estén aún operacionales y conectados, cumpliendo la red con requerimientos de calidad de servicio. En esta Tesis se analiza como varían las métricas error de computo, costo de requerimiento, y cantidad de nodos activos en función de la cantidad de iteraciones de computo.

6.2. Parámetros de Configuración

Los simuladores desarrollados permiten establecer, mediante un archivo, los diferentes parámetros de configuración para evaluar el funcionamiento de los protocolos propuestos. La configuración de los parámetros usados en las simulaciones se detalla en la Tabla 6.2. Se consideran diferentes configuraciones para cada uno de los análisis realizados. Desde el punto de vista de los datos, se considera en

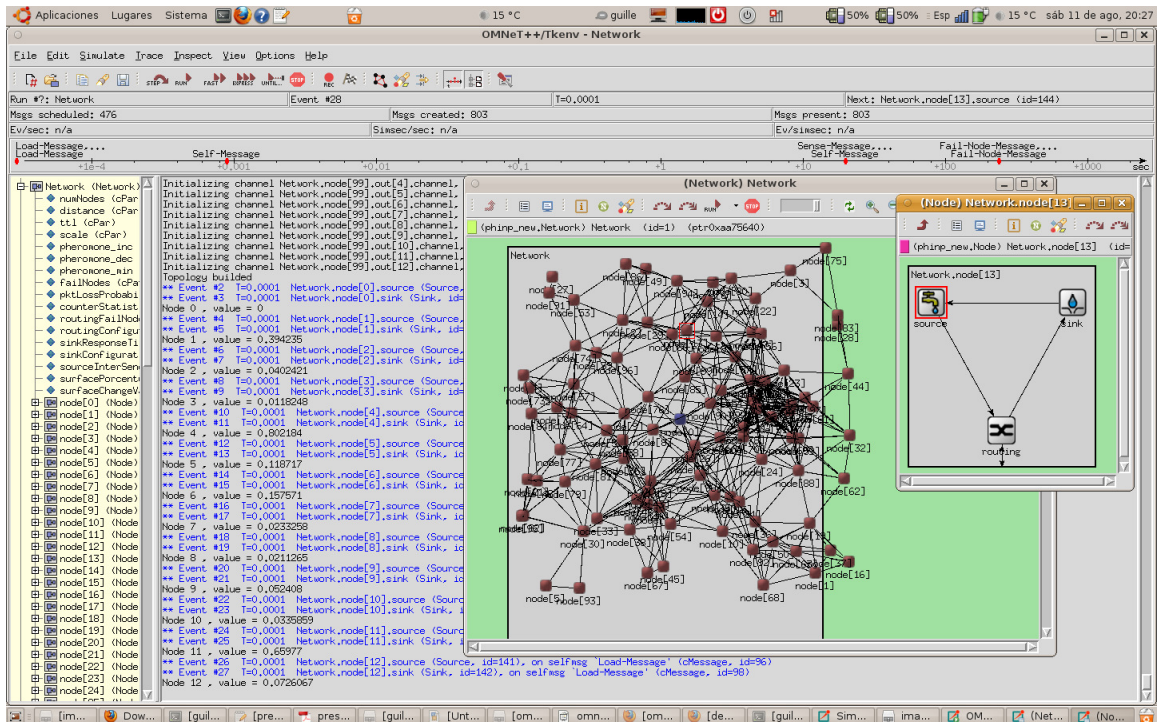


Figura 6.1: Entorno de simulación desarrollado en OMNeT++.

las simulaciones la representación de los datos obtenidos del sensado de la magnitud física en punto flotante. Esto hace que la correlación espacial de los datos tienda a cero. En general, se implementan dos tipos de análisis, considerando en ambos casos un rango de comunicación de los nodos constante y circular. Se evalúa la performance de los protocolos desarrollados según las diferentes métricas. El primer tipo de análisis considera que varía el tamaño de la red manteniendo constante la densidad de nodos. El segundo tipo considera que varía la densidad de nodos manteniendo constante el tamaño de la red. A fin de analizar la robustez de los protocolos propuestos, los simuladores desarrollados permiten configurar a partir de un cierto momento, tanto el porcentaje de nodos que fallan, como la probabilidad de pérdida de paquetes. Para analizar la capacidad de readaptación de los protocolos propuestos, se consideran diferentes valores tanto de la cantidad como de la movilidad de los eventos. El número de simulaciones es un parámetro fundamental para obtener resultados sin una variabilidad significativa. Empíricamente, se determinó que con la ejecución de 2000 simulaciones se cumple en todos los casos analizados con un buen valor de confianza. En el caso de la generación de las simulaciones, si bien es posible configurar diferentes generadores de números pseudo-aleatorios (PRNGs), el utilizado por defecto en OMNeT++ y en esta Tesis, por considerarlo el más adecuado es el *Mersenne Twister*. Este generador presenta buena uniformidad multidimensional y propiedades estadísticas, además que es muy rápida comparado a otros algoritmos, y útil en simulaciones en las que se requiere una gran cantidad de números aleatorios.

6.3. Protocolos Utilizados para Comparación

Para comparar la performance de los protocolos propuestos, se utilizaron protocolos tradicionales tales como Flooding, Gossip, y Random Walk. En la Tabla 6.1 se resumen las características mas

relevantes de los protocolos tradicionales que son usados en esta Tesis para compararlos con los protocolos propuestos.

Caract. / Protocolo	Flooding	Gossiping	Random Walk
Comunicación	Broadcast	Broadcast	Unicast
Decisión de Tx		Tx o no	A que nodo Tx
Tipo de Decisión	Determinística	Probabilística	Probabilística
Optimización	Controlado		No contra lazos
Detención	1 Tx cada nodo	Probabilística	Cantidad de saltos

Tabla 6.1: Características de los protocolos tradicionales utilizados.

Flooding es el primer protocolo utilizado en esta Tesis para comparación con los protocolos propuestos. En este sentido, se implementa una versión controlada de este protocolo, basada en la condición de que cada nodo puede retransmitir solo una vez el paquete recibido. De esta forma se limita el problema de explosión de comunicación propio de Flooding. El segundo protocolo utilizado es Random Walk. Como su nombre lo indica, se basa en el envío del paquete por caminos aleatorios, basado en unicast. En este caso, cada nodo que recibe un paquete, decide aleatoriamente a que nodo vecino se lo va transmitir. Sin embargo, este protocolo no ha sido optimizado en esta Tesis para evitar el problema de lazos cerrados, que es un problema común de Random Walk. Como condición de terminación, cada paquete transmitido lleva asociado el número de saltos actual, y el máximo posible. Cada nodo que recibe un paquete, incrementa el número de saltos, y analiza si se cumplió la condición de finalización. En las simulaciones realizadas en esta tesis, se consideraron dos situaciones de terminación, una que considera que el paquete puede ser retransmitido una cantidad de veces igual a la cantidad de nodos presentes en la red, y otra que considera a la mitad de estos nodos.

6.4. Análisis del Protocolo sin Estados

En esta sección se analiza el protocolo sin estados, analizando tanto el *error de computación* como el *costo de comunicación para la diseminación del requerimiento*. Se verifica el desempeño del protocolo sin estados respecto a protocolos existentes en un escenario con datos estáticos, a cambios en la amplitud de los eventos, a fallas en nodos, y a probabilidad de pérdida de paquetes. El modelo de distribución de datos utilizado en estas simulaciones consiste en una superficie formada por la sumatoria de 160 funciones exponenciales con diferentes posiciones y amplitudes. Se simulan redes desde 50 hasta 800 nodos sensores.

6.4.1. Comparación con otros Protocolos

En este caso se compara la performance de PASA respecto a dos protocolos tradicionales, como son Flooding y Random Walk, considerando que no hay variación en los datos sensados en el proceso de adaptación del protocolo PASA. La versión utilizada de Flooding es la controlada, es decir, que cada nodo puede retransmitir solo una vez el requerimiento. Por otro lado, se considera en el caso de Random Walk, que el valor máximo de saltos permitido es igual a la cantidad de nodos en la red, donde un nodo puede retransmitir más de una vez el mismo requerimiento, y no se implementan estrategias para evitar lazos cerrados. Es decir, se considera que la elección del camino es totalmente

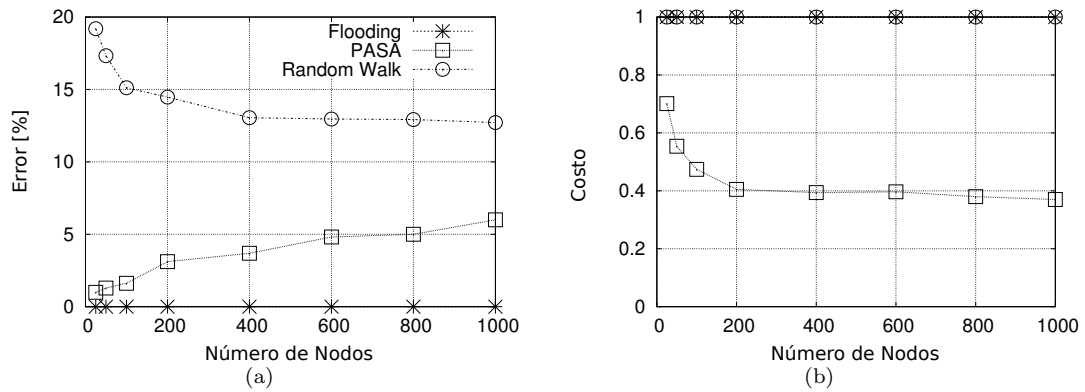


Figura 6.2: Comparación con protocolos existentes: error de cómputo (a) y costo (b).

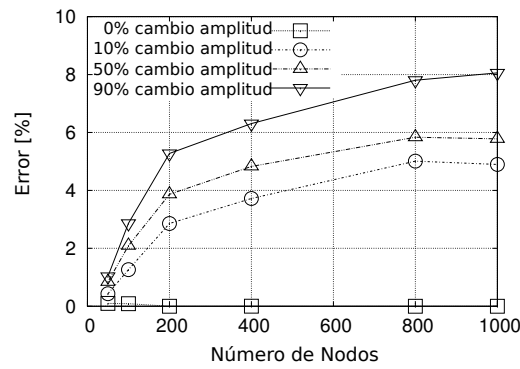


Figura 6.3: Error de cómputo ante cambios de amplitud de los eventos.

aleatoria, sin optimización. De los resultados obtenidos descritos en la Figura 6.2 se puede ver que PASA tiene un comportamiento intermedio a Flooding y a Random Walk. Debido a que el objetivo es minimizar tanto el error de cómputo como el costo de comunicación, PASA es el que tiene el mejor desempeño de los tres protocolos. Por un lado, utilizando Flooding no se comete error de cómputo, a expensas de un elevado costo de comunicación. Por otro lado, Random Walk incurre en el mismo costo de comunicación, con errores que varían entre el 13 y el 19%.

Respecto al comportamiento del protocolo PASA respecto al tamaño de la red, se puede observar que si bien existe cierta dependencia del error con el tamaño de la red, esta no crece en forma exponencial.

6.4.2. Cambio de Amplitud

En este caso se analiza el error de cómputo en el que incurre PASA cuando se produce un cambio de amplitud de los eventos que representan la magnitud sensada. Para ello, una vez que el protocolo PASA se ha adaptado a la distribución de los datos en la red, convergiendo a un estado de baja energía, se aplica un cierto porcentaje de cambio en las amplitudes de todos los eventos, y se verifica el error de cómputo de PASA luego de este cambio. En la Figura 6.3 se puede observar que este error se incrementa a medida que aumenta el tamaño de la red.

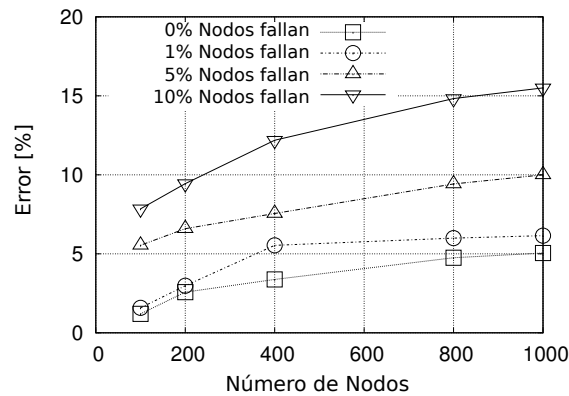


Figura 6.4: Error de computo ante fallas de nodos.

6.4.3. Falla de Nodos

Una de los aspectos más importantes para un protocolo de procesamiento en WSNs es su robustez a la falla de nodos. Esta es la capacidad para llevar adelante su tarea ante nodos que de un momento a otro fallen o se queden sin energía. Para implementar esta condición, se mantiene constante la superficie que modela los datos, y se deja que PASA se adapte a los mismos, convergiendo a un estado de mínima energía. Una vez que se cumplió con el criterio de convergencia de PASA, se hace fallar a un cierto porcentaje de nodos de forma uniforme. En la siguiente iteración, es analizado el comportamiento de PASA. En la Figura 6.4 se muestran los resultados obtenidos considerando que fallan hasta un 10% de los nodos que forman la red.

6.4.4. Paquetes Perdidos

Una de las condiciones más comunmente encontradas en WSNs, principalmente cuando estas son densas y de gran tamaño, es la probabilidad de pérdida de paquetes. Esto es consecuencia de que en el proceso de diseminación de requerimientos no se utilizan mecanismos de confirmación de recepción de los paquetes, ni de retransmisión. Se asume en la diseminación, que si hay algún paquete perdido, una copia del mismo puede llegar a un nodo a través de otro camino. En la simulación se considera que la magnitud sensada no varía en el tiempo, se deja adaptar a PASA a esta distribución de datos, y cuando está adaptado, se aplica un cierto porcentaje de paquetes perdidos, y se analiza el error cometido en el computo. En la Figura 6.5 se puede observar como se incrementa este error a medida que aumenta el tamaño de la red. El máximo error de computo obtenido para el caso de una pérdida de paquetes del 10% es de alrededor del 5%.

6.5. Análisis del Protocolo con Estados

En este análisis del protocolo con estados se analiza tanto el *error de computación*, como el *costo de comunicación para la diseminación del requerimiento*. Se verifica el desempeño del protocolo con estados respecto a protocolos existentes, a cambios en la amplitud de los eventos, a fallas en nodos, y a probabilidad de pérdida de paquetes. El modelo de distribución de datos utilizado en estas simulaciones

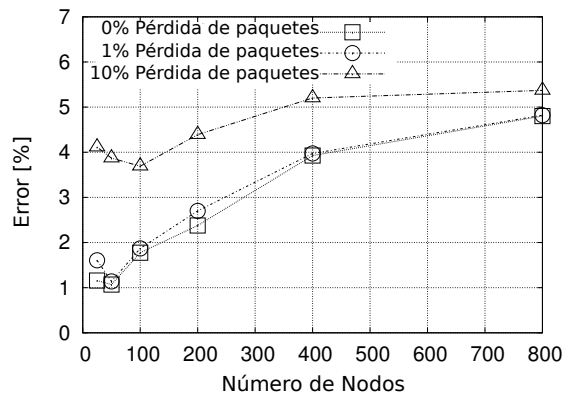


Figura 6.5: Error de cómputo ante pérdida de paquetes.

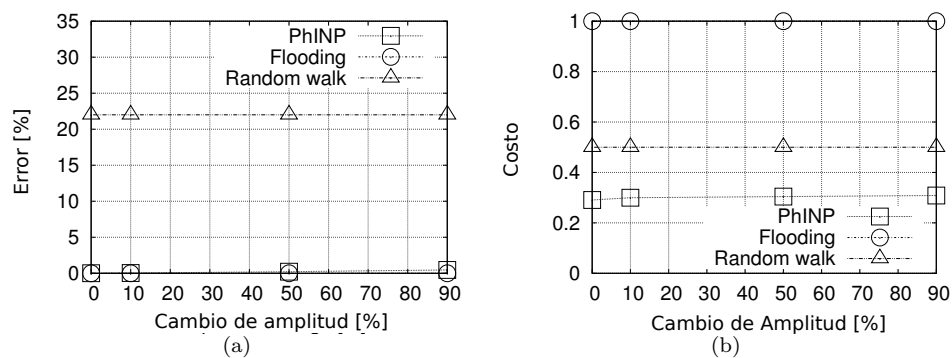


Figura 6.6: Comparación con protocolos existentes: error de cómputo (a), y costo de requerimiento (b).

consiste en una superficie formada por la sumatoria de 160 funciones exponenciales con diferentes posiciones y amplitudes. Se simulan redes desde 50 hasta 400 nodos sensores. La configuración utilizada para el protocolo PhINP es de $\lambda_{min} = 0,1$, $\lambda_{max} = 0,9$, y $\lambda_{dec} = 0,1$ y $\lambda_{inc} = 0,2$.

6.5.1. Comparación con otros Protocolos

Un análisis comparativo de la performance del protocolo con estados respecto a protocolos tradicionales es resumido en la Figura 6.6. En esta figura se puede observar por un lado el error cometido en el cómputo de la función de interés, y por otro lado el costo de comunicación en el que se incurre cuando se aplica un determinado porcentaje máximo de cambio en las amplitudes de las funciones que conforman la magnitud física medida luego que el algoritmo con estados PhINP está adaptado en su búsqueda. Se puede ver que Random Walk es el mecanismo que incurre en un mayor error, debido a que su búsqueda es de naturaleza aleatoria, no basada en gradientes como el caso PhINP, y que Flooding si bien no incurre en errores tiene un costo muy elevado de comunicación. En resumen, el esquema con estados PhINP permite minimizar en gran medida los errores de búsqueda y de computación que se pueden producir ante la dinámica de la magnitud sensada debido a su naturaleza de búsqueda basada en gradientes de información, y por otro lado minimizar el costo de comunicación respecto a esquemas tradicionales debido a su estrategia de reducción iterativa del espacio de búsqueda y readaptación en caso de ser necesario.

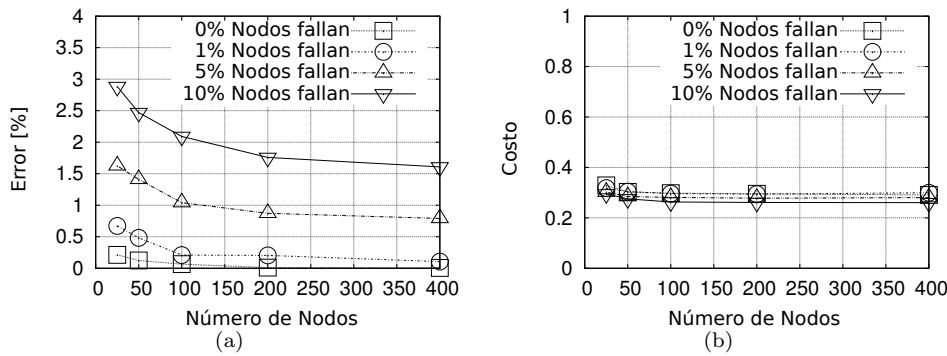


Figura 6.7: Error de cómputo ante fallas de nodos.

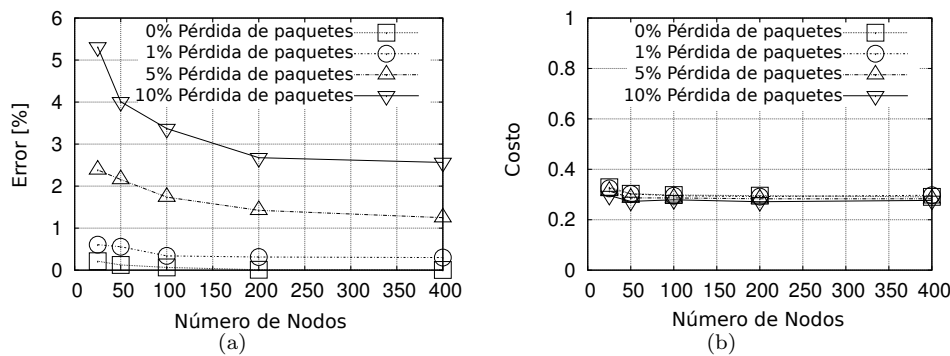


Figura 6.8: Error de cómputo ante pérdida de paquetes.

6.5.2. Falla de Nodos

En este caso, se analiza la robustez del protocolo con estados respecto a la probabilidad de fallas de nodos, ocurrido luego de la adaptación del esquema al escenario, osea, luego de la convergencia. Los resultados obtenidos son mostrados en la Figura 6.7. En la misma se puede ver que para falla de hasta el 10% de los nodos, y para diferentes tamaños de red, nunca se supera el 3% de error relativo de cómputo.

6.5.3. Paquetes Perdidos

Un análisis muy importante es el de la robustez del protocolo ante la pérdida de paquetes. En la Figura 6.8 se describen los resultados obtenidos de las simulaciones realizadas. De la misma se puede observar que con una probabilidad de pérdida de paquetes de hasta el 10%, no se supera el 5% de error relativo de cómputo, para diferentes tamaños de red.

6.5.4. Tiempo de Vida de la Red

En este caso, se busca analizar no solo el tiempo de vida de la red, como la cantidad de tiempo que transcurre hasta que un nodo o un cierto porcentaje de nodos quedan sin energía, sino comprender la dinámica de la misma. Para ello se considera que los nodos tienen una cantidad de energía finita

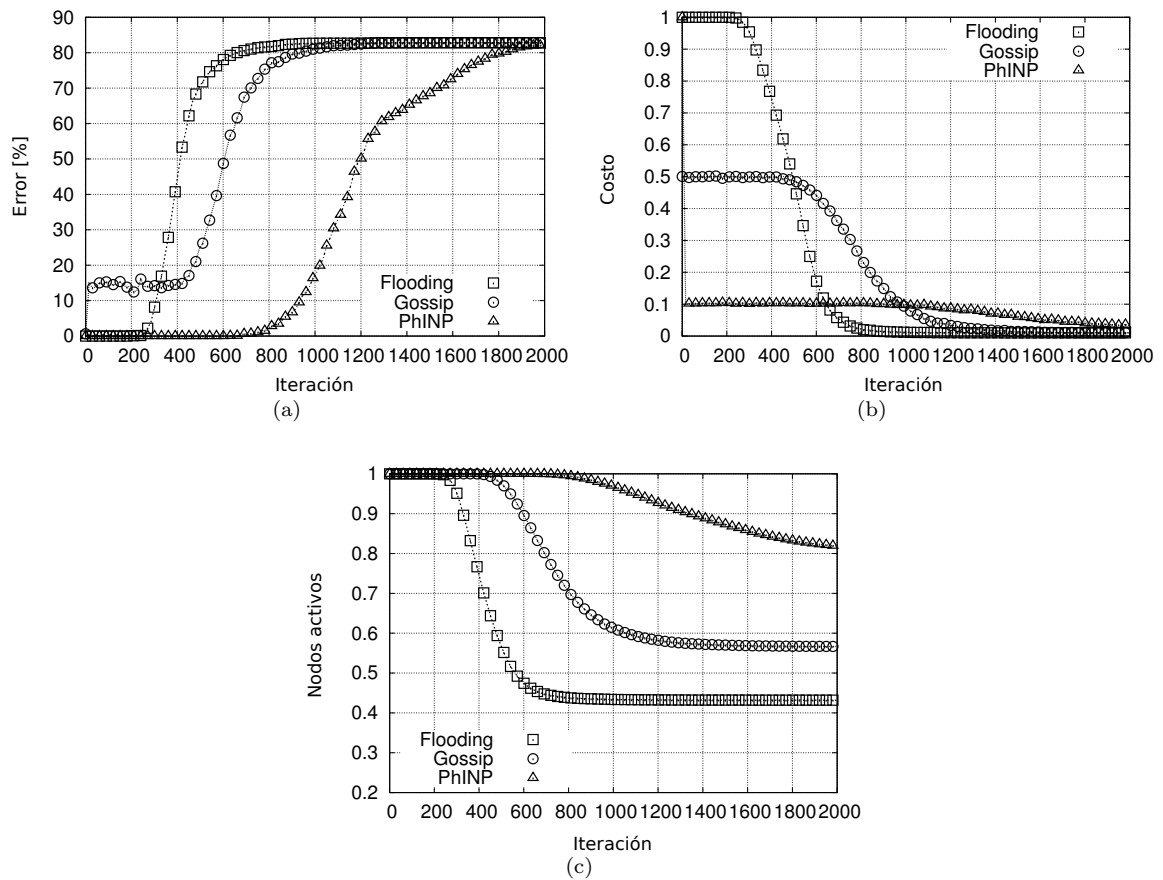


Figura 6.9: Tiempo de vida de la red: error de computo (a), costo de requerimiento (b), y cantidad de nodos activos (c).

E_T en Joules equivalente a la suministrada por dos baterías alcalinas AA. Bajo las consideraciones de la sección 2.4, esta energía es equivalente a 7800 tareas de transmisión o recepción de paquetes, ya que se considera que se consume la misma cantidad de energía para la transmisión y la recepción de un paquete, y que el nodo sink no tiene limitación de energía. Se analizaron tres métricas, para comprender la dinámica de Flooding, Gossip, y PhINP respecto al tiempo de vida de la red. Estas son el error de computo, el costo de requerimiento, y la cantidad de nodos activos en la red en función del número de iteraciones de computación, que es equivalente al tiempo de funcionamiento de la red. Para ello se considera que los datos son constantes, y que no se producen pérdida de paquetes. Se analiza el caso de Flooding controlado, en el que cada nodo solo puede retransmitir una vez el mensaje de requerimiento, el caso de Gossip con una probabilidad de retransmisión de 0,5, y PhINP configurado con $\lambda_{min} = 0,1$, $\lambda_{max} = 1$, y $\lambda_{dec} = 0,1$ y $\lambda_{inc} = 0,2$. Un nodo puede recibir y transmitir paquetes mientras tenga energía remanente. Cuando esta es nula, el nodo es apagado. Debido a la gran cantidad de iteraciones de computo en cada ejecución, los tiempos de simulación son elevados. Se puede observar en la Figura 6.9 que si bien con Flooding no se cometen errores de computo, tiene el inconveniente de que a partir de las 220 iteraciones de computo los nodos comienzan a quedarse sin energía rápidamente.

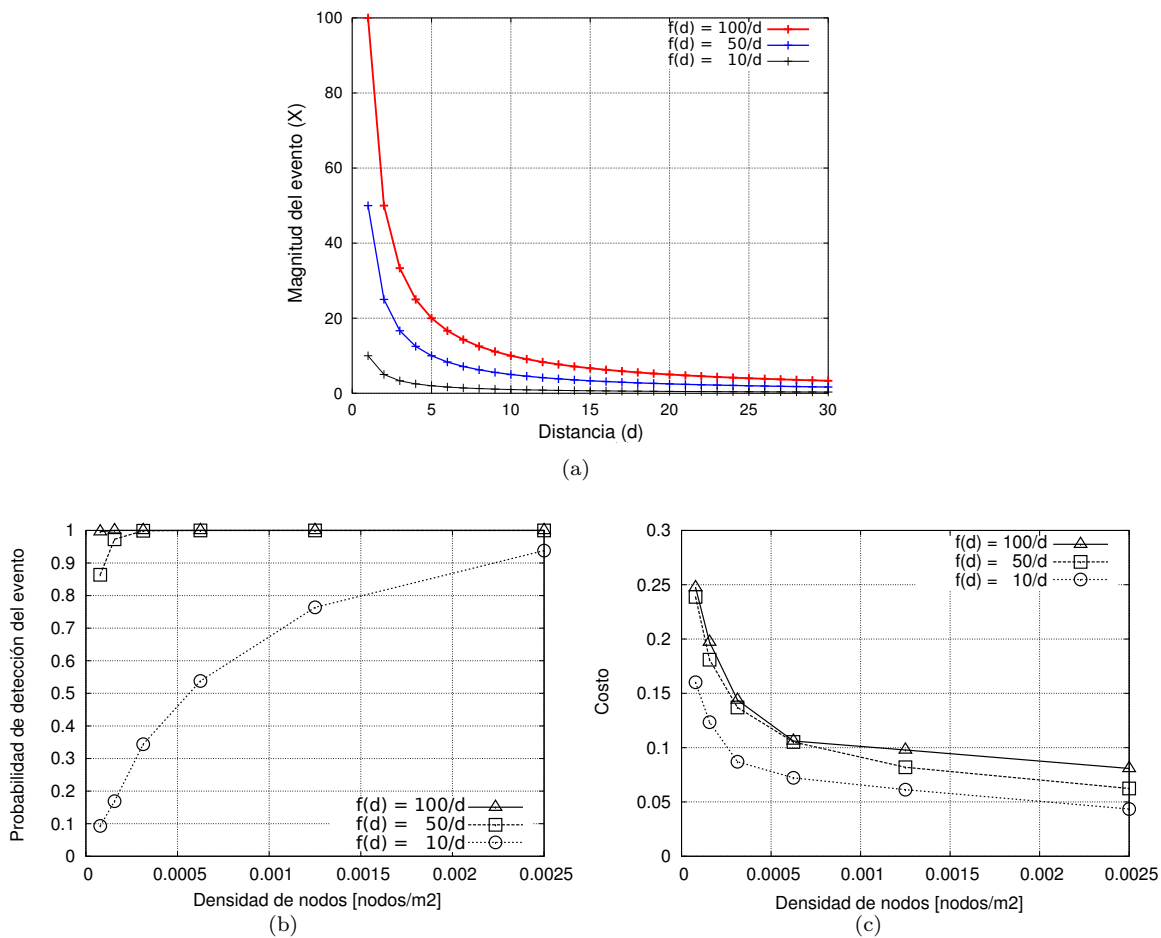


Figura 6.10: Capacidad de detección con resolución finita: magnitud de los eventos (a), probabilidad de detección del evento (b), y costo de requerimiento (c).

6.5.5. Monitoreo con Resolución Finita

En todos los casos analizados en esta Tesis, se considera que los nodos sensores tienen una resolución de punto flotante, con 5 dígitos después de la coma. Debido a ello, se puede considerar que la correlación de los datos es prácticamente nula. En este caso, se busca determinar la capacidad de detectar eventos por parte de los nodos de la red, y el costo de requerimiento asociado, en función de la densidad de nodos en la red.

Para ello se considera un solo evento en la red con posición aleatoria. Se modela a dicho evento con un coeficiente de difusión $\alpha = 1$, representando por ejemplo, a una fuente de calor, y se considera que el mismo puede tener tres amplitudes diferentes, de 10, 50, y 100, como se muestra en la Figura 6.10.

6.6. Análisis de la Comparación entre Protocolos

Diferentes aspectos de los protocolos propuestos fueron evaluados durante este trabajo, con el fin de comparar su performance y determinar los mejores campos de aplicación para cada uno de

Parámetro	Valor
Área de la red desplegada	200x200, 283x283, 400x400 m
Tamaño de la red	100, 200, 400 nodos
Densidad de nodos	2.5×10^{-3} nodos/m ²
Rango de Comunicación	Circular, 50 m
Nodos vecinos promedios por nodo	19.635
Probabilidad de falla de nodos	0, 1, 5, 10 %
Número de eventos	1 to 400
Movimiento de eventos	20 to 200 % del rango de comunicación
Número de simulaciones	2000
Generador de números aleatorios	Mersenne Twister

Tabla 6.2: Parámetros de configuración usados en las simulaciones.

los protocolos. Nuestro análisis incluye la convergencia del algoritmo, análisis de eventos, robustez a falla de nodos, a pérdida de paquetes y a la dinámica de los eventos, y finalmente la capacidad de readaptación a cambios de los eventos presentes en la red. Finalmente, se resumen los resultados comparativos de los protocolos propuestos.

6.6.1. Convergencia

Debido a que ambos protocolos propuestos, sin estado y con estado, aprenden en el tiempo, es de esperar que sus métricas experimenten variaciones durante las primeras iteraciones de computo. En el caso del protocolo sin estados, este necesita converger a un valor T_0 , mientras que en el protocolos con estados, el nivel de feromona en los nodos necesita estabilizarse. Para simplificar el análisis, y entender como se comportan estos protocolos, se considera una red en donde tanto los nodos como los enlaces de comunicación son ideales, es decir, los nodos no pueden fallar, y los enlaces están libres de pérdidas de datos. Además, se asume que el campo sensado de datos está conformado por 10 fuentes de información o eventos distribuidos en forma aleatoria y con la misma amplitud, los cuales siguen la ley de difusión del calor ($\alpha = 1$). Durante la convergencia de los protocolos propuestos, se considera que este campo es estático, es decir, no se cambia ni sus amplitudes ni sus posiciones.

En el *protocolo sin estados*, el nodo sink configura un valor grande T_0 en el requerimiento inyectado a la red en la primer iteración ($T_0(1) = 10000$ es considerado en todas las simulaciones), para alcanzar a cada nodo en la red. Basado en el conjunto de nodos que responden el requerimiento con sus datos en la primera iteración, el nodo sink puede estimar, basado en técnicas de aprendizaje por refuerzo, si existe pérdida de información en la siguientes iteraciones, y ajustar el valor de $T_0(i)$ en los siguientes envíos de requerimientos para evitar dichas pérdidas, como se describió en la Sección 4.2.3. El objetivo de esta estrategia es reducir el costo de diseminación de requerimientos en la mayor medida posible, mientras no exista pérdida de información. Debido a la naturaleza probabilística del protocolo sin estados propuesto, este puede incurrir en errores de computación relativamente bajos una vez que el algoritmo ha convergido, cercanos al 3 % en caso de los escenarios analizados. Las métricas de error de computación y de costo de requerimiento obtenidas por simulación son mostradas en la Figura 6.11a,b, respectivamente.

Por otro lado, dado que el *protocolo con estados* también inunda la red con requerimientos en la primer iteración, el error de computación permanece nulo en las siguientes iteraciones, considerando que no hay cambios mayores en los datos sensados en la red. Bajo estas condiciones, la convergencia

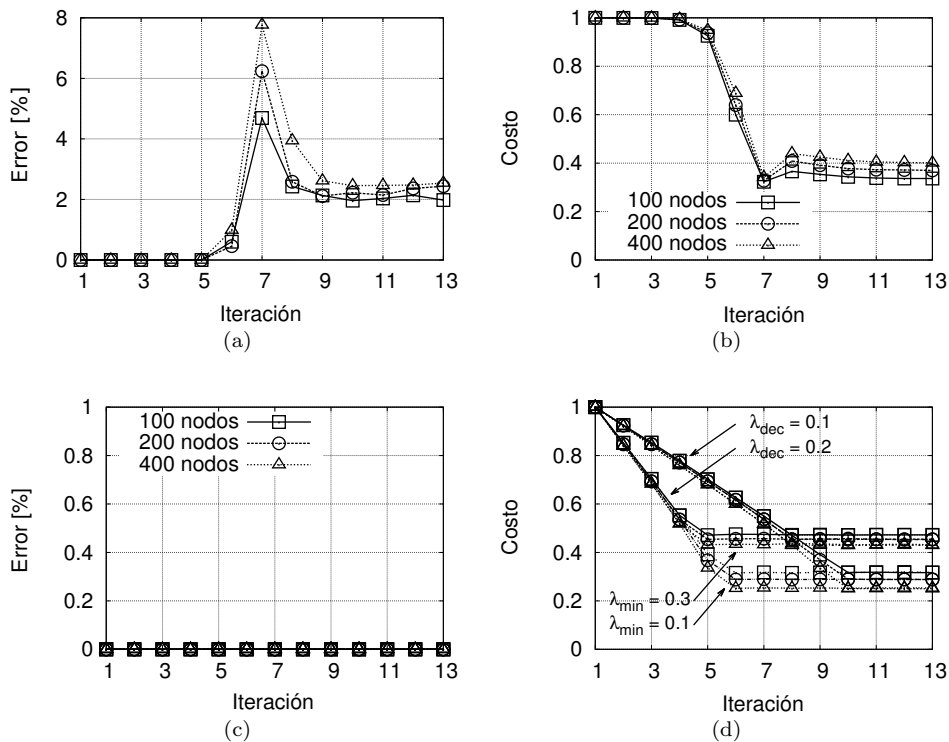


Figura 6.11: Convergencia: protocolo sin estados (a,b) y con estados (c,d).

del algoritmo puede ser analizada como si este intenta decrementar el costo de requerimiento mientras se mantiene el error nulo (es decir, recolectando los mismos datos relevantes) como se muestra en la Figura 6.11c,d. En efecto, la primer iteración tiene siempre un costo igual a 1, debido a que cada nodo retransmite el requerimiento una sola vez. Después de cada iteración, se refuerzan caminos entre el nodo sink y nodos con datos relevantes basado en niveles de feromona locales a cada nodo, y la probabilidad P de usar otros caminos es reducida, minimizando el costo de comunicación. El comportamiento de cada nodo cuando recibe un requerimiento es descrito por 4 parámetros fijos (λ_{max} , λ_{min} , λ_{inc} , and λ_{dec}), y por uno variable λ , pudiendo tener un valor entre 0 y 1. El trabajo colaborativo de toda la red define en una manera descentralizada la performance de este protocolo. En este análisis, diferentes valores de λ_{dec} , λ_{inc} , y λ_{min} son considerados, para comprender la operación de este protocolo. Cuanto mayores son estos valores, más rápida es la convergencia a una configuración de mínimo costo; sin embargo, este costo tiende a ser mayor para mayores valores de λ_{dec} y λ_{inc} . Existe un balance entre tiempo de convergencia y costo mínimo. El valor de λ_{max} es siempre configurado a 1 para evitar errores en la computación. El comportamiento de este protocolo para diferentes tamaños de red, como se muestra en la Figura 6.11c,d, es independiente del tamaño de la red.

De las simulaciones obtenidas se puede observar que el costo de requerimiento para el protocolo sin estados puede ser decrementado a más del 50% con respecto a Flooding, mientras se mantiene el error limitado a menos del 2.5% con el escenario considerado. Notese que el protocolo con estados puede minimizar el costo de comunicación respecto al del protocolo sin estados mediante la configuración de un menor valor de λ_{min} , a expensas de una menor robustez a eventos dinámicos. Ello se debe a la menor probabilidad de escapar de mínimos locales, como se discutirá más adelante. Si $\lambda_{min} \rightarrow 0$, entonces los costos de comunicación de diseminación de requerimientos y de respuesta tienden a tener

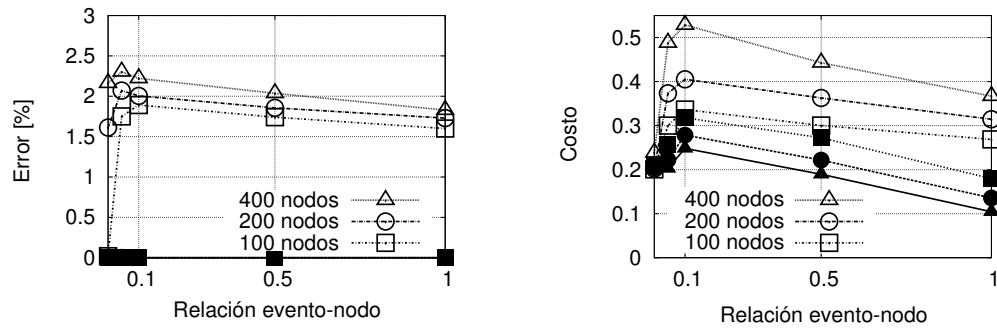


Figura 6.12: Análisis de eventos de protocolo sin estados (blanco), y con estados (negro).

el mismo valor, debido a que los mismos caminos son usados para retransmitir tanto requerimientos como respuestas.

Además, el protocolo sin estados tiende a introducir más errores que el protocolo con estados. Sin embargo, el segundo requiere de buena sincronización de los parámetros de configuración debido a su naturaleza distribuida, para evitar inconsistencias. En el protocolo sin estados, esta condición no es crítica, debido a que el nodo sink configura y envía la política de enfriamiento dentro del mensaje de requerimiento.

6.6.2. Análisis de Eventos

En este caso se evalúa el performance de los protocolos sin estado y con estado para diferentes campos de datos, es decir, desde el punto de vista de la distribución de los datos en la red. En este sentido, se define la métrica relación evento-sensor (esr), la cual es la relación entre la cantidad de fuentes de información o eventos y la de nodos sensores presentes en la red. En este sentido, si se tienen n nodos sensores y e eventos, el factor esr puede ser definido como $esr = e/n$. Normalmente, los eventos y los nodos no están en las mismas posiciones, es decir, se asume una distribución aleatoria independiente entre ambos.

Para simplificar el análisis, se considera que todos los eventos tienen las mismas características, es decir, la misma amplitud y coeficiente de difusión. Se consideran valores de esr que van desde 0.01 a 1, y configuraciones de $T_0(1) = 10000$ para el protocolo sin estado y $[1 \ 0.2 \ 0.1 \ 0.1]$ para el protocolo con estados. Los resultados obtenidos por simulación son mostrados en la Figura 6.12. Se puede observar que el protocolo sin estados tiene un error de computación muy bajo, cuyo valor mínimo se da cuando $esr \rightarrow 0$, es decir, para el caso de un evento único o de pocos eventos en la red. Por otro lado, un $esr \approx 0,1$ introduce el máximo error (en caso del protocolo sin estados), con el máximo costo de comunicación para ambos protocolos. Como consecuencia, se define un escenario con 10 eventos como caso de estudio a lo largo de todo este trabajo. Podemos observar también, que el costo de comunicación se reduce rápidamente a medida que $esr \rightarrow 1$ debido al proceso de filtrado en la red.

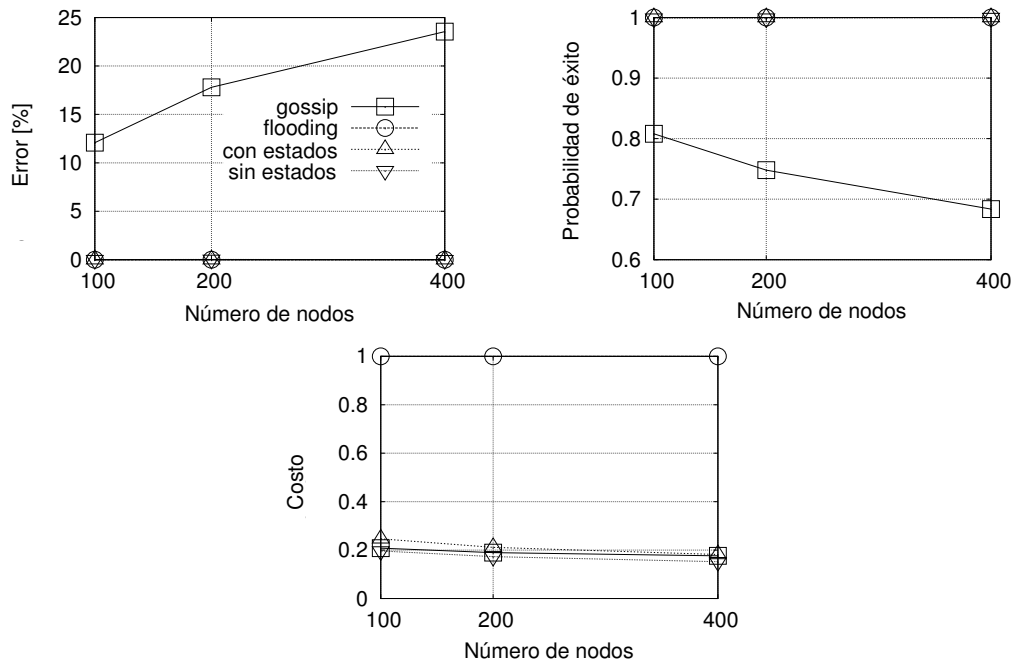


Figura 6.13: Detección de un evento para diferentes protocolos.

6.6.3. Detección de Evento Único

En este caso, se analiza la capacidad de los protocolos para detectar un evento aleatoriamente localizado en la red. Tengase en cuenta que este análisis es un caso particular del anterior en el que $esr \rightarrow 0$. Sin embargo, el objetivo en este caso es comparar la performance de ambos protocolos respecto a dos protocolos tradicionales tales como Flooding y Gossip. Gossip es configurado con una *probabilidad gossip* = 0,25 con el fin de tener el mismo costo de requerimiento que los protocolos propuestos. La performance de estos protocolos es mostrada en la Figura 6.13. Se puede observar que Flooding tiene la capacidad de encontrar siempre aquellos nodos con argumentos relevantes para computar una función dada, pero este incurre en un elevado costo de comunicación, debido a que cada nodo tiene que retransmitir el mensaje de requerimiento (es decir, costo de requerimiento = 1). Además, las performances de los protocolos sin y con estado se encuentran entre las obtenidas para los mecanismos de Flooding y Gossip. Por otro lado, los protocolos propuestos incurren en errores mucho más bajos que Gossip, y por otro lado, pueden reducir el costo de comunicación en gran medida, en este caso en particular, una reducción del 80 %.

6.6.4. Robustez ante Fallas de Nodos

Una vez que los protocolos sin y con estados han llegado a su convergencia, luego de varias iteraciones, se vuelve crítico para ellos poder seguir trabajando eficientemente bajo condiciones de falla. De acuerdo a esto, se considera el caso de nodos con cierta probabilidad de falla. La Figura 6.14 muestra la performance de ambos protocolos para diferentes probabilidades de falla de nodos, a medida que el tamaño de la red es incrementado. Para simplificar este análisis, se supone que la distribución de los datos sensados es estática, es decir, que los valores sensados no cambian en el tiempo.

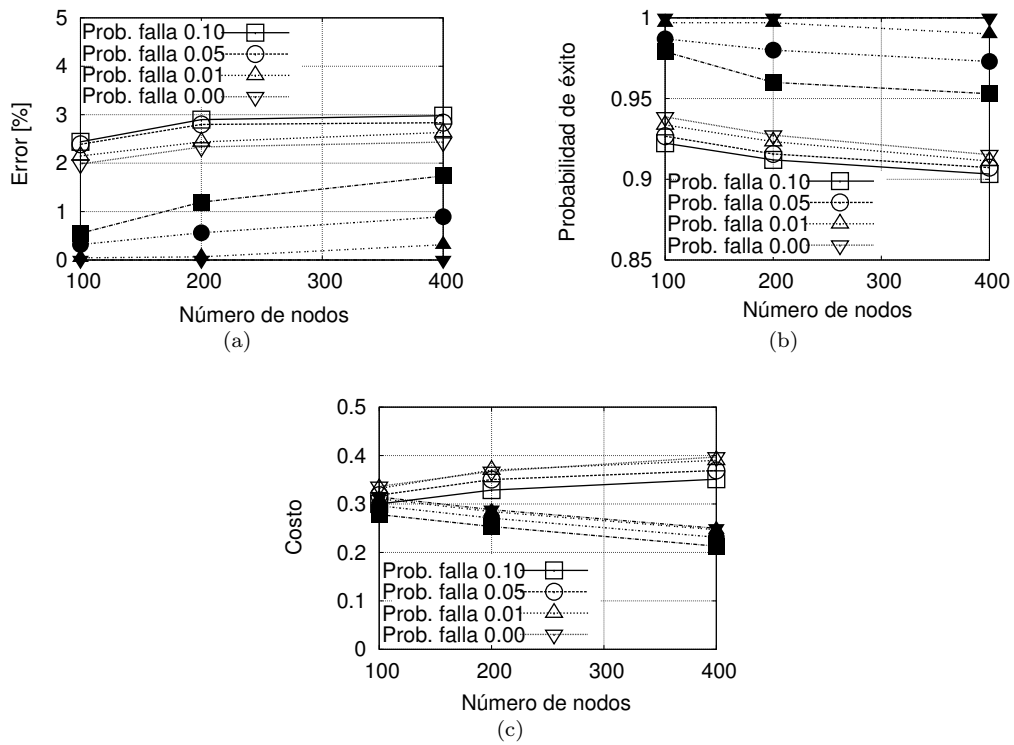


Figura 6.14: Robustez ante falla de nodos de los protocolos sin estados (blanco), y con estados (negro).

Como es de esperar, el protocolo con estados, a pesar de tener un error bajo de computación en el rango de los tamaños de red analizados, es más sensible o susceptible a fallas en nodos respecto al protocolo sin estados. La razón de esto es que el protocolo con estados tiende a mantener caminos simples (es decir, caminos marcados con feromona) para el enrutamiento de los requerimientos entre el nodo sink y cada uno de los nodos sensores que proveen argumentos relevantes para computar la función en el sink. En este sentido, la falla de uno de los nodos retransmisores o intermediarios ubicados sobre estos caminos es crítica, debido a que esto puede afectar la computación de la función. Como se puede observar, esta conducta del protocolo con estados para distributivamente conformar caminos puede ser relajada, debido a que el ancho de los caminos es una función del valor λ_{min} . El término ancho se refiere a la característica de diseminación de requerimientos en la red por múltiples caminos. En este sentido, con un valor bajo de λ_{min} (por ejemplo, $\lambda_{min} \rightarrow 0$), se tiende a formar un camino simple entre el nodo sink y un nodo que es seleccionado. A medida que λ_{min} se incrementa, cada requerimiento diseminado arriba a un nodo a través de mayor cantidad de caminos (es decir, por múltiples caminos). Este efecto es descrito en la Figura 6.15 para un mejor entendimiento. Por otro lado, en el caso del protocolo sin estados, el requerimiento arriba a un nodo a través de múltiples caminos. De este modo, una falla de un nodo de retransmisión, es decir, un nodo en el camino entre el sink y un potencial nodo con información relevante, no es tan significativo (Figura 6.15a). En cambio, en el caso del protocolo con estados, si es usado un valor bajo de λ_{min} , el protocolo tiende a formar caminos angostos luego de la convergencia, los cuales están más afectados por fallas en los nodos como se describe en la Figura 6.15b. Para mayores valores de λ_{min} , el protocolo con estados tiende a tener similar comportamiento que el caso del protocolo sin estados desde el punto de vista de los mensajes de requerimiento que arriban a un nodo, a expensas de un mayor costo de comunicación. Una descripción más detallada de esta característica es dada en la Sección 6.7. Como se explicó anteriormente, en

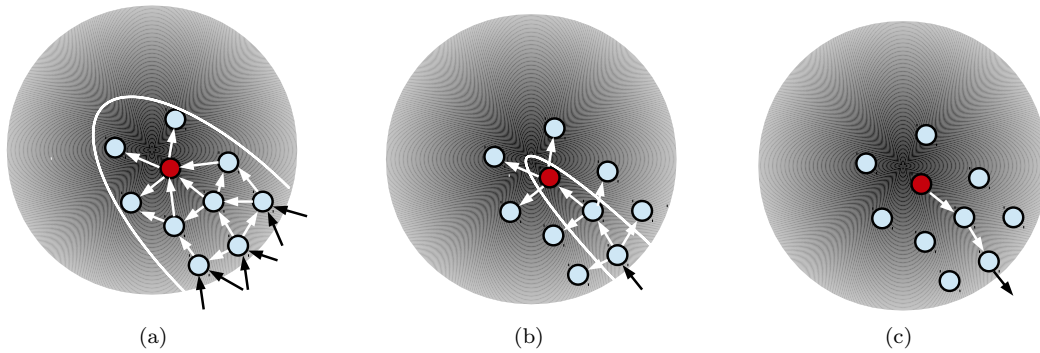


Figura 6.15: Diseminación de requerimientos: protocolo sin estados (a), y con estados (b), y retransmisión de respuesta (c) luego de la convergencia.

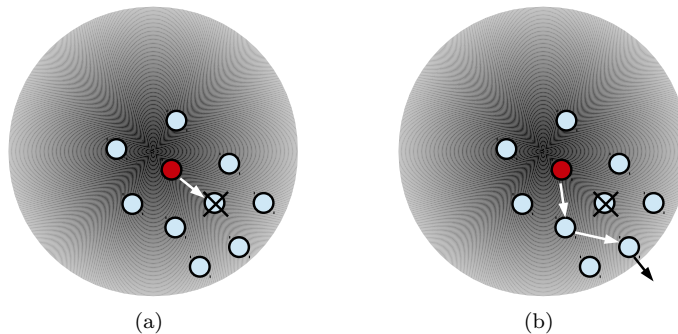


Figura 6.16: Mecanismo de auto-curado para hacer frente a fallas en nodos.

ambos protocolos se reportan los argumentos de los nodos que han quedado seleccionados utilizando el camino inverso de aquel camino de primer arribo, como se muestra en la Figura 6.15c.

A fin de incrementar la robustez de los protocolos propuestos ante falla de nodos o pérdida de paquetes, se propone un *mecanismo simple de auto-curado (self-healing)*. El objetivo principal de este mecanismo es evitar la pérdida de datos relevantes para la aplicación en el proceso de reporte de datos. En este sentido, cada nodo que envía sus datos hacia el nodo sink puede, mediante el uso de aprendizaje de refuerzo basado en escucha activa, detectar si el paquete de datos ha sido retransmitido por su nodo vecino o no. En caso de que no detecte la retransmisión, el nodo que envía el dato lo vuelve a retransmitir. Este mecanismo es seguido por cada nodo retransmisor en el camino de respuesta. En este sentido, cada nodo mantiene una tabla con información de nodos vecinos, la cual es actualizada en cada iteración de computación. Esta tabla es ordenada en orden temporal de arribo del requerimiento, en el cual se almacenan los *id* de los nodos vecinos inmediatos (a un salto de distancia) a través de los cuales se recibe el requerimiento. Tengase en cuenta que el protocolo sin estados puede también beneficiarse de esta mejora, no solo para el caso del protocolo con estados configurado con valores bajos de λ_{min} . En la Figura 6.16a se describe el caso en el cual, el nodo vecino que debe retransmitir el dato de respuesta hacia el nodo sink ha fallado. Ante esta situación, el nodo que transmitió el dato, al no detectar la retransmisión de su vecino en el transcurso de un instante de tiempo, puede seleccionar, de su tabla local, a otro nodo vecino mas próximo al sink para retransmitir el dato (Figura 6.16b).

Característica \ Protocolo	Sin Estados (PASA)	Con Estados (PhINP)
Tipo de control	centralizado	distribuido
Parámetro de control	T	λ
Localización del parameter	dentro del requerimiento	local a cada nodo
Comportamiento como Flooding	$T \gg$	$\lambda_{min}=1$ or $\lambda_{dec}=0$
Velocidad de convergencia	logarítmica, lineal D	$\lambda_{inc}, \lambda_{dec}, \lambda_{min}$
Capacidad de readaptación	no	si
Velocidad de readaptación	-	$\lambda_{inc}/\lambda_{dec}$

Tabla 6.3: Comparación de los protocolos propuestos.

6.7. Comparación de los Protocolos

Basado en los resultados obtenidos de los análisis realizados, se puede ver que cada protocolo propuesto tiene su propio campo de aplicación mas conveniente. Es decir, cada protocolo es más eficiente un para un escenario dado. Este análisis considera que ambos protocolos ya han convergido a un estado de baja energía, adaptándose a los datos presentes en la red. En este sentido, el protocolo con estados es más apropiado que el protocolo sin estados en escenarios con eventos dinámicos. Sin embargo, el protocolo sin estados es más robusto a fallas en nodos, debido a que el mismo mensaje de requerimiento puede arribar a un nodo a través de múltiples caminos más probablemente. En este sentido, en el protocolo con estados existe un trade-off entre el costo de comunicación y la robustez ante falla de nodos. Si se configuran los nodos con un valor bajo de λ_{min} , este protocolo tiende a un estado de baja energía, con un costo mínimo de comunicación, a expensas de una baja robustez a falla de nodos, y viceversa.

Una diferencia sustancial que existe entre ambos protocolos es la probabilidad de exploración de los nodos sensores (es decir, la capacidad de retransmitir el requerimiento), relacionado con la regla de filtrado de nodos vecinos. En el protocolo con estados, esta probabilidad es limitado al valor de λ_{min} , el cual es independiente de la posición del nodo en la red. Sin embargo, en el protocolo sin estados, esta probabilidad es dependiente de la distancia entre este nodo y el sink, tal que, cuando más lejos esté el nodo, menor es la probabilidad de retransmitir el requerimiento (es decir, menor capacidad de exploración). Esto es debido a la degradación de temperatura T que sufre el mensaje de requerimiento cuando este es retransmitido a través de la red, usando un control centralizado. En este sentido, la temperatura configurada en el mensaje de requerimiento por el sink es la mayor que puede tener durante la iteración. Esto es consecuencia de un control centralizado. Esta última característica hace mas confiable al protocolo con estados para WSNs grandes.

6.8. Comentarios de Cierre

En este capítulo se describió el análisis de los resultados obtenidos de los protocolos de filtrado sin estados y con estados para diferentes tipos de escenarios y condiciones. Además, se los comparó respecto a protocolos tradicionales, y se realizó una comparación entre ellos para determinar cual es el escenario más adecuado para cada uno de ellos.

Conclusiones

EL computo de funciones estadísticas de datos obtenidos a partir de medidas realizadas a magnitudes físicas (por ej. calor, luz, etc), ejecutado en forma distribuida dentro de una red inalámbrica de sensores, surge como una novedosa y eficiente solución. Sin embargo, los nodos que conforman la red tienen capacidades limitadas de energía, computación, y memoria, lo que limita su aplicabilidad. El manejo eficiente de estos recursos es un desafío para la adopción masiva de esta tecnología.

Debido a que la comunicación inalámbrica es el principal factor de consumo energético, y que a diferencia de las redes tradicionales, los nodos pueden procesar los datos transportados dentro de los paquetes cuando estos son retransmitidos, con un uso reducido de energía, se busca desarrollar protocolos de procesamiento de datos que permitan minimizar la comunicación.

Haciendo uso de esta característica se han propuesto en los últimos años paradigmas de enrutamiento de paquetes basados en datos, y se han desarrollado una gran variedad de estrategias de procesamiento dentro de la red, como por ejemplo agregación y fusión de datos, los que permiten reducir la cantidad de paquetes transmitidos. Sin embargo, en el caso de la computación de ciertas funciones, como por ejemplo, funciones simétricas de tipo umbral (por ej. máximo, mínimo, rango, isocontorno, etc), en las que solamente un pequeño conjunto de nodos aporta argumentos relevantes, reportar los datos de todos los nodos y aplicar técnicas de agregación o fusión es ineficiente. Una alternativa es considerar a este problema como de búsqueda de información, el cual es desafiante si no se conoce a priori la localización de estos nodos. Debido a la dinámica de estas redes y de los parámetros a medir, no solo se requiere de una etapa inicial de descubrimiento, sino también de una continua búsqueda de nuevos datos, por lo que se lo ejecuta en forma iterativa.

Se muestran indicios muy fuertes en esta tesis, mediante análisis y simulación, que la hipótesis de aplicar protocolos de procesamiento basados en búsqueda mediante gradientes de información, con mecanismos de filtrado y aprendizaje, para la computación de funciones de umbral, es más eficiente que utilizar protocolos de agregación o fusión de datos.

Como trabajos futuros se pretende implementar estos protocolos desarrollados en redes de sensores reales, hacer público el código de implementación de los mismos para que puedan ser utilizados y ampliados en sus funcionalidades, y generar documentación de como deben ser configuradas cada uno

de los parámetros de configuración.

Bibliografía

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, pp. 393-422, 2002.
- [2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey," *Computer Networks*, vol. 52, no. 12, pp. 2292-2330, 2008.
- [3] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, "Energy Conservation in Wireless Sensor Networks: A Survey," *Ad Hoc Networks*, vol. 7, pp. 537-568, 2009.
- [4] K. Han, J. Luo, Y. Liu, and A. Vasilakos, "Algorithm Design for Data Communications in Duty-Cycled Wireless Sensor Networks: A Survey," *IEEE Communications Magazine*, vol. 51, Iss. 7, pp. 107-113, 2013.
- [5] R. Rashid, and G. Robertson, "Accent, A Communication Oriented Network Operating System Kernel," *In Proc. 8th Symposium on Operating Systems Principles*, pp. 64-75, 1981.
- [6] M. Frodigh, P. Johansson, and P. Larsson, "Wireless Ad Hoc Networking—The Art of Networking without a Network," *Ericsson Review*, no. 4, pp. 248-263, 2000.
- [7] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A Survey of Active Network Research," *IEEE Communnications Magazine*, vol. 35, pp. 80-86, 1997.
- [8] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *ACM Mobicom '99*, pp. 263-270, 1999.
- [9] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, "Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks," *In Proceedings of Fourth IEEE Workshop on Mobile Computing Systems and Applications*, pp. 49-58, 2002.
- [10] G. J. Pottie, and W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51-58, May 2000.
- [11] "10 Emerging Technologies That Will Change The World," *MIT Technology Review*, Feb. 2003.
- [12] C. G. Bell, R. Cady, H. McFarland, B. Delagi, J. O'Laughlin, R. Noonan, and W. Wulf, "A New Architecture for Mini-Computers – The DEC PDP-11," *Sprint Joint Computer Conference*, pp. 657-675, 1970.

- [13] P. K. Kikiras, D. K. Drakoulis, D. A. Dres, and G. I. Stamoulis, "Wireless Sensor Networks: Business Models and Market Issues," *6th Conf. on Telecommunication Techno-Economics - CTTE 2007*, pp.1-5, 2007.
- [14] J. Zheng, and M. J. Lee, "Will IEEE 802.15.4 Make Ubiquitous Networking a Reality?: A Discussion on a Potential Low Power, Low Bit Rate Standard," *IEEE Communications Magazine*, vol. 42, no. 6, pp. 140-146, 2004.
- [15] A. Wheeler, "Commercial Applications of Wireless Sensor Networks Using ZigBee," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 70-77, 2007.
- [16] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless Sensor Networks: A Survey on the State of the Art and the 802.15.4 and ZigBee Standards," *Elsevier Comput. Commun.*, 30, 7, 2007.
- [17] IEEE 802.15.4, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)", 2003.
- [18] ZigBee Specification (2007), ZigBee Alliance, 2007.
- [19] Plan estratégico nacional de Ciencia, Tecnología e Innovación "Bicentenario" (2006-2010), Ministerio de Ciencia, Tecnología e Innovación Productiva.
- [20] J. M. Rabaey, M. J. Ammer, J. L. Da Silva, D. Patel, and S. Roundy, "PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking," *Computer*, vol. 33, no. 7, pp. 42-48, 2000.
- [21] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: a Survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115-121, 2006.
- [22] J. N. Al-Karaki, and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6-28, 2004.
- [23] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks," *In Proceedings of the 7th ACM Int. Symposium on Mobile Ad Hoc Networking and Computing*, 2006.
- [24] D. Jung, T. Teixeira, A. Barton-Sweeney, and A. Savvides, "Model-Based Design Exploration of Wireless Sensor Node Lifetimes", *Lecture Notes in Computer Science*, Springer, vol. 4373, 2007.
- [25] D. Niyato, E. Hossain, and A. Fallahi, "Sleep and Wakeup Strategies in Solar-Powered Wireless Sensor/Mesh Networks: Performance Analysis and Optimization," *IEEE Transactions on Mobile Computing*, vol. 6, no. 2, pp. 221-236, 2007.
- [26] V. Raghunathan, S. Ganeriwal, and M. Srivastava, "Emerging Techniques for long lived Wireless Sensor Networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 108-114, 2006.
- [27] D. Niyato, E. Hossain, N. M. Rashid, and V. K. Bhargava, "Wireless Sensor Networks with Energy Harvesting Technologies: a game-theoretic Approach to Optimal Energy Management," *IEEE Wireless Communications*, vol. 14, no. 4, pp. 90-96, 2007.
- [28] W. Qu, K. Li, M. Kitsuregawa, and T. Nanya, "An Efficient Method for Improving Data Collection Precision in Lifetime-adaptive Wireless Sensor Networks," *IEEE Int. Conference on Communications - ICC '07*, pp. 3161-3166, 2007.

- [29] X. Tang, J. Xu, "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," *Proc. 25th IEEE Int. Conference on Computer Communications - INFOCOM 2006*, pp. 1-12, 2006.
- [30] R. Rajagopalan, and P. Varshney, "Data Aggregation Techniques in Sensor Networks: A Survey," *IEEE Communication Surveys & Tutorials*, 4th Quarter 2006, vol. 8, no. 4, pp. 48-63, 2006.
- [31] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, pp. 70-87, 2007.
- [32] J. Wong, and M. Potkonjak, "Search in Sensor Networks: Challenges, Techniques, and Applications," *In IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. IV, pp. 3752-3755, 2002.
- [33] Z. Cheng, and W. Heinzelman, "Flooding Strategy for Target Discovery in Wireless Networks," *Wireless Networks*, vol. 11, pp. 607-618, 2005.
- [34] K. Rachuri, and C. Siva Ram Murthy, "Energy Efficient and Scalable Search in Dense Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 58, no. 6, pp. 178-185, 2009.
- [35] J. Faruque, and A. Helmy, "RUGGED: Routing on Fingerprint Gradients in Sensor Networks," *IEEE Int. Conference on Pervasive Services (ICPS 2004)*, pp. 179-188, 2004.
- [36] J. Faruque, and A. Helmy, "PBS: A Virtual Grid Architecture for Information Gradient-based Active Querying in Sensor Networks," *Int. Symposium on Collaborative Technologies and Systems. CTS 2008*, pp. 9-18, 2008.
- [37] J. Liu, F. Zhao, and D. Petrovic, "Information-Directed Routing in Ad Hoc Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, 2005.
- [38] F. Dressler, and O. Akan, "A Survey on Bio-Inspired Networking," *Computer Networks*, Elsevier, vol. 54 (6), pp. 881-900, 2010.
- [39] G. Riva, and J. Finochietto, "A Probabilistic Query Routing Scheme for Wireless Sensor Networks," *Argentine Symposium on Artificial Intelligence, ASAI 2011*, JAIIO, 2011
- [40] J. P. Leal Licudis, J. C. Abdala, G. Riva, and J. Finochietto, "Flexible Prototyping for Ad Hoc Wireless Sensor Network Protocols", *Argentine Symposium on Technology, AST 2011*, JAIIO, 2011.
- [41] G. Riva, and J. Finochietto, "A Parallel and Adaptative Query Routing Scheme for Wireless Sensor Networks," *IEEE International Conference on Communications (ICC 2012)*, pp. 243-247, 2012.
- [42] G. Riva, and J. Finochietto, "Pheromone-based In-Network Processing for Wireless Sensor Network Monitoring Systems," *IEEE International Conference on Communications (ICC 2012)*, pp. 6560-6564, 2012.
- [43] G. Riva, and J. Finochietto, "Pheromone-based In-Network Processing for Wireless Sensor Network Monitoring Systems," *Journal of Network Protocols and Algorithms (NPA)*, Macrothink Institute, vol. 4, no. 4, pp. 156-173, 2012.

- [44] G. Riva, J. Finochietto, and G. Leguizamón, "Bio-Inspired In-Network Filtering for Wireless Sensor Monitoring Systems," *IEEE Congress on Evolutionary Computation (CEC)*, pp. 3379-3386, 2013.
- [45] G. Riva, and J. Finochietto, "In-Network Filtering Schemes for Type-Threshold Function Computation in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks (IJDSN)*, Hindawi, Special Issue: Bio-Inspired Mechanisms in Wireless Sensor Networks, 2014.
- [46] <http://db.csail.mit.edu/labdata/labdata.html>
- [47] <http://lcav.epfl.ch/page-86035-en.html>
- [48] G. Piccinini, and A. Scarantino, "Computation vs. Information Processing: Why their difference matters to cognitive science," *Studies in History and Philosophy of Science* 41, pp. 237-246, 2010.
- [49] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," *Int. Workshop on Distributed Event-Based Systems - DEBS '02*, 2002.
- [50] S. Madden, M. Franklin, and J. Hellerstein, "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks," *5th Annual Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.
- [51] A. Giridhar, and P. R. Kumar, "In-Network Information Processing in Wireless Sensor Networks," *Wireless Sensor Networks: Signal Processing and Communication Perspectives*, Wiley, 2007.
- [52] A. Giridhar, P. R. Kumar, "Computing and Communicating Functions over Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 755-764, 2005.
- [53] A. Giridhar, P. R. Kumar, "Toward a Theory of In-Network Computation in Wireless Sensor Networks," *IEEE Communications Magazine*, vol. 44, iss. 4, pp. 98-107, 2006.
- [54] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building Efficient Wireless Sensor Networks with Low-Level Naming," *Proceedings of the eighteenth ACM symposium on Operating systems principles, SOSP'01*, pp. 146-159, 2001.
- [55] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *MobiCom*, pp. 56-67, 2000.
- [56] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling Data-Centric Routing in Wireless Sensor Networks," *Int. Workshop on Distributed Event-Based Systems -DEBS '02*, 2002.
- [57] M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks," *Int. Journal of High Performance Computing Applications*, vol. 16, no 3, pp. 293-313, 2002.
- [58] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Gradient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," *Wireless Networks*, 11, pp. 285-298, 2005.
- [59] R. Baumann, S. Heimlicher, V. Lenders, and M. May, "HEAT: Scalable Routing in Wireless Mesh Networks using Temperature Fields," *IEEE Int. Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2007*, pp. 1-9, 2007.

- [60] C. Buchhorsts, and D. Dujovne, "Natural Gradient Routing: Sink Convergence using Data as Guide," *IEEE Latin-America Conference on Communications, (LATINCOM)*, pp. 1-6, 2013.
- [61] K. Akkaya, and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Networks*, 3 (2005), pp. 325-349, 2005.
- [62] D. Braginsky, and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," *ACM Int. Workshop on Wireless Sensor Networks and Applications*, pp. 22-31, USA, 2002.
- [63] F. Xue, and P. R. Kumar, "The Number of Neighbors needed for Connectivity of Wireless Networks," *Wireless Networks*, vol. 10, no 2, pp. 169-181, 2004.
- [64] R. V. Kulkarni, A. Förster, and G. K. Venayagamoorthy, "Computational Intelligence in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 13, No 1, pp. 68-96, 2011.
- [65] E. Talbi, "Metaheuristics: From Design to Implementation," Wiley, ISBN: 978-0-470-27858-1, 2009.
- [66] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science, New Series*, vol. 220, No. 4598, pp. 671-680, 1983.
- [67] A. Zimmerman, and J. Lynch, "A Parallel Simulated Annealing Architecture for Model Updating in Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 9, no. 11, pp. 1503-1510, 2009.
- [68] R. Sutton, and A. G. Barto, "Reinforcement Learning: An Introduction," MIT Press, 1998.
- [69] K. Alvin Yau, P. Komisarczuk, and P. Teal, "Reinforcement Learning for Context Awareness and Intelligence in Wireless Sensor Networks: Review, New Features and Open Issues," *Journal of Network and Computer Applications*, Elsevier, 35 (2012), pp. 253-267, 2012.
- [70] B. Park, S. Park, E. Lee, S. Noh, and S. Kim, "Large-scale Phenomena Monitoring Scheme in Wireless Sensor Networks," *In Proc. 71st IEEE Vehicular Technology Conference - VTC 2010-Spring*, pp. 1-5, 2010.
- [71] M. Li, and Y. Liu, "Iso-Map: Energy-Efficient Contour Mapping in Wireless Sensor Networks," *In IEEE Transactions on Knowledge and Data Engineering*, vol. 22, No. 5, pp. 699-710, 2010.
- [72] J. Sun, "Multi-Threshold Based Data Gathering Algorithms for Wireless Sensor Networks," *Journal of Networks*, vol. 4, no. 1, pp. 30-41, 2009.
- [73] J. Heidemann, F. Silva, Y. Yu, D. Estrin, and P. Haldar, "Diffusion Filters as a Flexible Architecture for Event-Notification in Wireless Sensor Networks," *In ISI-TR-556, USC/ISI*, pp. 1-7, 2002.
- [74] D. J. T. Sumpter, "The Principles of Collective Animal Behaviour," *Philosophical Trans. of the Royal Society B* 361: 5-22, 2005.
- [75] OMNeT++ simulation library. Available at: <http://www.omnetpp.org>
- [76] J. Gehrke, and S. Madden, "Query Processing in Sensor Networks," *IEEE Pervasive Computing*, vol. 3, no 1, pp. 46-55, 2004.

- [77] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A Taxonomy of Wireless Micro-Sensor Network Models," *Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28-36, 2002.
- [78] J. Ahn, S. Kapadia, S. Patten, A. Sridharan, M. Zuniga, and J. Jun, "Empirical Evaluation of Querying Mechanisms for Unstructured Wireless Sensor Networks," *ACM SIGCOMM Computer Comm. Review*, vol. 38, No 3, pp. 19-26, 2008.
- [79] F. Kuhn, T. Locher, and R. Wattenhofer, "Tight Bounds for Distributed Selection," *7th IEEE Int. Conference on Peer-to-Peer Computing (P2P)*, 2007.
- [80] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors," *Proc. of the First IEEE Workshop on Embedded Networked Sensors*, 2004.
- [81] M. Saleem, G. Di Caro, and M. Farooq, "Swarm Intelligence-based Routing Protocol for Wireless Sensor Networks: Survey and Future Directions," *Information Sciences*, pp. 4597-4624, 2011.
- [82] T. Camilo, C. Carreto, J. Silva, and F. Boavida, "An Energy-Efficient Ant-Based Routing Algorithm for Wireless Sensor Networks," *In Proc. ANTS, Lecture Notes in Computer Science (LNCS)*, Springer, vol. 4150, pp. 49-59, 2006.
- [83] S. Okdem, and D. Karaboga, "Routing in Wireless Sensor Networks Using an Ant Colony Optimization (ACO) Router Chip," *Sensors 2009*, 9, 909-921, 2009.
- [84] F. Ducatelle, G. Di Caro, and L. Gambardella, "Using Ant Agents to Combine Reactive and Proactive Strategies for Routing in Mobile Ad Hoc Networks," *Int. Journal of Computational Intelligence and Applications (IJCIA2005)*, 2005.
- [85] M Brandl, T. Posnicek, KH. Kellner, A. Kos, C. Mayerhofer, and C. Fabian, "A Simple Pheromone based On-Demand Routing Protocol for Wireless Sensor Networks," *Proceed. Fourth Int. Conf. Syst. Netw. Commun. (ICSNC2009)*, pp. 254-260, 2009.
- [86] X. Hui, Z. Zhi-gang, and N. Feng, "A Novel Routing Protocol in Wireless Sensor Networks based on Ant Colony Optimization," *Int. Journal of Intelligent Information Technology Application*, 3(1), pp. 1-5, 2010.
- [87] E. Michlmayr, A. Pany, and G. Kappel, "Using Taxonomies for Content-based Routing with Ants," *Int. World Wide Web Conference (WWW 2006)*, 2006.
- [88] T. Stutzle, and H. Hoos, "Max-Min Ant System," *Future Generation Computing Systems*, vol. 16, pp. 889-914, 2000.
- [89] M. Amiri, "Evaluation of Lifetime Bounds of Wireless Sensor Networks", *CoRR*, vol. Abs/1011.2103, 2010.
- [90] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast Gossip Algorithms for Consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748-2761, 2009.
- [91] M. Franceschelli, A. Giua, and C. Seatzu, "Distributed Averaging in Sensor Networks Based on Broadcast Gossip Algorithms," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 808-817, 2011.
- [92] A. Meliou, C. Guestrin, and J. Hellerstein, "Approximating Sensor Network Queries Using In-Network Summaries," *Proc. of IPSN*, pp. 229-240, 2009.

-
- [93] Y. Xu, T. Fu, W. Lee, and J. Winter, "Processing K Nearest Neighbor Queries in Location-Aware Sensor Networks," *Signal Processing*, 87 (12), pp. 2861-2881, 2007.
- [94] Y. Xu, W. Lee, J. Xu, and G. Mitchell, "Processing Window Queries in Wireless Sensor Networks," *Proceedings of the 22nd Inter. Conference on Data Engineering (ICDE'06)*, pp. 70, 2006.
- [95] H. Huang, J. Hartman, and T. Hurst, "Efficient and Robust Query Processing for Mobile Wireless Sensor Networks," *International Journal of Sensor Networks*, vol. 2, no. 1/2, pp. 99-107, 2006.
- [96] J. Zhang, X. Zhu, and H. Peng, "Bi-Filtered Forwarding: A Quasi-Optimal Routing Algorithm for Query Delivery in Wireless Sensor Networks," *International Journal on Smart Sensing and Intelligent Systems*, pp. 993-1011, 2013.
- [97] A. M. Zungeru, L. Ang, and K. P. Seng, "Classical and Swarm Intelligence based Routing Protocols for Wireless Sensor Networks: A Survey and Comparison," *Elsevier, Journal of Network and Computer Applications (JNCA)*, 35, pp. 1508-1536, 2012.
- [98] L. Vieira, U. Lee, and M. Gerla, "Phero-Trail: Bio-inspired Routing in Underwater Sensor Networks," *IEEE Journal on Selected Areas in Communications, JSAC*, vol. 28, no. 4, pp. 553-563, 2010.