Universidad de Murcia

Departamento de Ingeniería de la Información y las Comunicaciones

Algoritmos Fiables y Eficientes basados en Enrutamiento Geográfico para Redes Realistas de Sensores Inalámbricos

Tesis Doctoral

Autor:

Rafael Marin Perez

Director:

Dr. Pedro Miguel Ruiz Martínez

Murcia, Junio 2012

University of Murcia

Department of Information and Communications Engineering

Reliable and Efficient Algorithms based on Geographic Routing for Realistic Wireless Sensor Networks

Ph.D. Thesis

Authored by:

Rafael Marin Perez

Directed by:

Dr. Pedro Miguel Ruiz Martínez

Murcia, June 2012

D. Pedro Miguel Ruiz Martínez, Profesor Titular de Universidad del área de Ingeniería Telemáica en el Departamento de Ingeniería de la Información y las Comunicaciones de la Universidad de Murcia.

AUTORIZA:

La presentación de la Tesis Doctoral titulada "Algoritmos Fiables y Eficientes basados en Encaminamiento Geográfico para Redes Realistas de Sensores Inalámbricos", realizada por D. Rafael Marín Pérez, bajo su inmediata dirección y supervisión, en el Departamento de Ingeniería de la Información y las Comunicaciones, y que presenta para la obtención del grado de Doctor por la Universidad de Murcia.

En Murcia, a 20 de Junio de 2012

Dr. Pedro M. Ruiz Martínez

Agradecimientos

a Pedro por su firme dirección y constancia.
a Sanchez por sus sabios consejos.
a los compañeros de Dibulibu por el buen ambiente.
a todos los miembros del departamento por su ayuda con la red de sensores.
a mi familia y amigos por sus ánimos.
a mi amada mujer, Ana, por su alegría y apoyo.

Acknowledgments

to Pedro for his solid direction and perseverance.
to Sanchez for his wise advices.
to the workmates of Dibulibu for the good environment.
to all the fellows of the department for their help with the sensor network.
to my family and friends for their encouragements.
to my loving wife, Ana, for her happiness and support.

Resumen

Las redes de sensores inalámbricas (*WSNs, Wireless Sensor Networks*) se encuentran entre las diez tecnologías emergentes del siglo XXI. La potencia principal de dichas redes es el fácil despliegue de una gran cantidad de dispositivos autónomos que son capaces de monitorizar y controlar fenómenos naturales localizados en lugares remotos. La monitorización mediante dispositivos inalámbricos evita usar cableado en el área de despliegue y así minimizar la perturbación del fenómeno de estudio. Esta tecnología proporciona impredecibles oportunidades en el control y monitorización de todo tipo de entornos.

Las redes de sensores posibilitan un amplio abanico de aplicaciones desde la monitorización de entornos naturales hasta el control de la salud de personas pasando por sistemas de localización y rastreo. Aunque las primeras aplicaciones de las redes de sensores fueron en escenarios militares, actualmente hay una alta diversidad de aplicaciones civiles. Su principal aplicación consiste en la colección automatizada de datos en grandes aéreas de estudio para el posterior análisis y procesamiento en un centro de control. En pocos años la tecnología de las redes de sensores permitirá la implantación de sensores en personas para el control remoto de su salud.

Dichas aplicaciones requieren de dispositivos más potentes, pequeños y baratos los cuales serán disponibles gracias a los avances en las tecnologías de mecánica y micro-electrónica de circuitos. Estos dispositivos sensores se caracterizan por su diseño hardware basado en cuatro componentes principales: sensores, radio, micro-controlador y batería. Múltiples sensores capturan información relativa al entorno del lugar de despliegue. Hay muchas clases

de sensores para medir parámetros físicos y químicos tales como temperatura, humedad, precipitación, etc. Una radio inalámbrica permite la comunicación con otros dispositivos sensores para enviar y recibir los datos capturados. Un micro-controlador posibilita la gestión de dichos datos y controla la tarea de comunicación. Una batería suele ser la fuente de energía limitada que alimenta el resto de los componentes. Resaltar que el consumo de la batería es un factor crucial en los dispositivos sensores ya que determina su tiempo de vida.

Las redes de sensores distribuyen la información capturada sin la necesidad de infraestructuras de comunicación de alto coste. Los dispositivos sensores, también llamados nodos sensores, emplean su radio inalámbrica para interconectar la red y establecer comunicaciones descentralizadas. Cada nodo realiza dos roles como fuente de datos y enrutador inalámbrico. En grandes redes, un nodo fuente no es capaz de enviar los paquetes de información a un destino lejano localizado fuera de su radio de cobertura. Por tanto, los nodos intermedios colaboran para encaminar los paquetes desde una fuente a un destino siguiendo un enfoque multisalto. En la comunicación multisalto, el nodo que actualmente posee el paquete debe decidir el siguiente salto entre sus nodos vecinos localizados dentro de su área de cobertura basada en una métrica de encaminamiento. Dicha decisión de encaminamiento consiste en elegir el mejor vecino para continuar el enrutamiento del paquete hacia el destino. Para tomar dichas decisiones, los nodos necesitan algoritmos de encaminamiento que determinen la forma de descubrir los vecinos y seleccionar el mejor candidato como siguiente salto.

El desarrollo de algoritmos de encaminamiento presenta varios retos importantes en las redes de sensores. Dichas redes están formadas por miles de dispositivos equipados con recursos limitados en término de cómputo, comunicación y energía. Además la comunicación inalámbrica es la tarea de mayor consumo de energía que representa el cuello de botella en la autonomía de dichas redes. Por estas razones, los protocolos de encaminamiento deben ser escalables para un gran número de dispositivos y eficientes de acuerdo al número de mensajes transmitidos. En la literatura, multiples algoritmos de encaminamiento han sido desarrollados para establecer comunicaciones

multisalto basados en diferentes paradigmas tales como: jerarquia, calidad de servicio y basado en los datos. Sin embargo, dichos paradigmas requieren del mantenimiento de tablas de rutas o mensajes de inundación para descubrir las rutas los cuales generan una sobrecarga excesiva y no son aplicables en redes de sensores.

Para proveer comunicación multisalto en redes de sensores, ha sido propuesto el encaminamiento geográfico como la solución más eficiente y escalable. El encaminamiento geográfico emplea las posiciones de los nodos sensores para tomar las decisiones de encaminamiento. El conocimiento de dichas posiciones no es un problema en dichas redes, ya que es una información necesaria por los nodos sensores. Dato que un dato capturado (e.g. temperatura) no es útil sin la posición donde este fue tomado. En el encaminamiento geográfico, cada nodo toma las decisiones de encaminamiento basado en su posición y las posiciones de sus vecinos a un salto. Este diseño localizado del encaminamiento geográfico escala muy bien con el número de dispositivos y reduce los requisitos de cómputo, ancho de banda y energía. Para garantizar la entrega de los datos, los algoritmos geográficos combinan dos estrategias diferentes: voraz (*Greedy*) y perimetral (*Face*). En el estrategia greedy, cada nodo encamina el paquete a su vecino localizado más cerca hacia el destino reduciendo la distancia en cada salto. Cuando el paquete alcanza un nodo que no tiene vecinos más cercanos al destino que él, entonces el nodo se convierte en un máximo local que posee un área vacía. En áreas vacías, la estrategia face es utilizada para encaminar el paquete alrededor del perímetro del área vacía hasta alcanzar nodos localizados más cerca del destino que el máximo local donde el modo greedy puede ser aplicado. El proceso combinado de greedy-face-greedy es repetido hasta que el paquete encuentra el destino. Para tomar las decisiones de encaminamiento, cada paquete incorpora la posición del destino y los nodos necesitan conocer solo la información local sobre las posiciones de sus vecinos a un salto. Para descubrir la información de los vecinos, los algoritmos de encaminamiento consideran que los nodos intercambian periódicamente mensajes de control a un salto, llamados *beacons*. Sin embargo, las transmisiones periódicas de dichos beacons malgastan

los escasos recursos de los nodos sensores como la energía y el ancho de banda incluso en aquellos nodos que no participan en el encaminamiento del paquete de datos. Para evitar estas transmisiones periódicas, han sido propuestos recientemente eficientes (*beaconless*) algoritmos que proporcionan soluciones reactivas para descubrir los vecinos a un salto y seleccionar el siguiente salto.

Sin embargo, los algoritmos de encaminamiento geográficos han sido diseñados y evaluados considerando redes de sensores con comunicaciones inalámbricas perfectas. Estos protocolos asumen normalmente enlaces inalámbricos sin pérdidas e interfaces radios con rangos de alcance fijos. Experimentos recientes han demostrado que las comunicaciones inalámbricas reales sufren frecuentemente de problemas como interferencias, colisiones, etc. Dichos errores de comunicación causan severos daños en el rendimiento de los protocolos de encaminamiento en término de pérdidas de paquetes y retransmisiones. Por esta razón, considerar comunicaciones realistas es esencial para el diseño de eficientes y fiables protocolos de encaminamiento para redes de sensores.

Además, los protocolos geográficos asumen que la información de localización es perfecta. Dichos protocolos ignoran las imprecisiones producidas por los sistemas distribuidos de posicionamiento empleados en los despliegues reales. Estudios recientes han mostrado que las algoritmos geográficos existentes son ineficientes en escenarios con posiciones imprecisas. Tanto la estrategia greedy como face experimentan un gran incremento de pérdidas de paquetes, cuando los errores de localización aumentan. Concretamente, en el modo greedy la principal razón para descartar paquetes de datos es las áreas vacías, y el 90% de los casos ocurren en el área de cobertura del destino. Por eso, se hace imprescindible que los algoritmos geográficos consideren los errores de localización en las decisiones de enrutamiento.

A parte de las perfectas condiciones de las comunicaciones inalámbricas y los sistemas de localización, la mayoría de algoritmos geográficos de encaminamiento consideran que las redes de sensores son desplegadas en aéreas seguras. No obstante, el medio inalámbrico es abierto y propenso a ser atacado

por nodos maliciosos que deseen interrumpir la comunicación entre los nodos sensores. En la literatura, se han mostrado múltiples ataques que actúan como nodos legítimos participando en los procesos de encaminamiento con el objetivo de conseguir ser seleccionados como siguientes saltos. Por ejemplo, en el encaminamiento geográfico un ataque sinkhole puede explotar el esquema reactivo de los protocolos beaconless para descubrir los vecinos. Además, un ataque sybil explota las decisiones de encaminamiento basadas en posiciones y crear posiciones falsas para conseguir ser elegido como el mejor candidato para todos los paquetes encaminados dentro de su área de cobertura. Una vez, un atacante es seleccionado como el siguiente salto, este descarta el paquete de datos.

Basado en las deficiencias de los algoritmos de encaminamiento existentes, esta tesis se enfoca en proporcionar algoritmos fiables y eficientes de encaminamiento para despliegues reales de redes de sensores inalámbricos. Primero, estudiamos en detalle los efectos de las condiciones reales en el rendimiento del encaminamiento geográfico. Después, proponemos tres algoritmos de encaminamiento mejorados que son capaces de tratar con errores de comunicación, imprecisiones en las posiciones y ataques de encaminamiento. Para validar los algoritmos propuestos en escenarios reales de redes de sensores, se emplean dos experimentos diferentes mediante simulaciones y un escenario real. Las simulaciones son realizadas para demostrar la escalabilidad y la eficiencia de los protocolos en redes con un gran número de dispositivos. Los experimentos en el escenario real son hechos para comprobar la fiabilidad y la robustez de los protocolos en redes de sensores reales.

Concretamente, proponemos BOSS, un algoritmo de encaminamiento geográfico que soporta errores en las comunicaciones inalámbricas con el objetivo de garantizar la entrega de los mensajes y reducir la sobrecarga de transmisiones. El rendimiento de BOSS es comparado con dos de los más eficientes algoritmos geográficos: IGF (*Implicit Geographic Forwarding*) y BLR (*Beacon-Less Geographic Routing*) Tantos los experimentos en simulaciones como en el escenario real demuestran que BOSS mejora el rendimiento de IGF y BLR en términos de fiabilidad de entrega y eficiencia en ancho de banda. Además,

todos los resultados confirman que BOSS se adapta bien a las propiedades intrínsecas de las comunicaciones inalámbricas de las redes de sensores.

Después de tratar con los errores de comunicación, se propone una extensión a BOSS llamada EGLE el cual es una solución eficaz para el encaminamiento geográfico que considera las imprecisiones en las posiciones. EGLE es comparado contra MER (*Maximum Expectation within Transmission Range*) el cual es el algoritmos geográfico con mejor rendimiento que considera los errores de localización en las decisiones de encaminamiento. En las evaluaciones mediante simulaciones y el escenario real, EGLE mejora el rendimiento de MER tanto en fiabilidad de entrega como en eficiencia de ancho de banda. Todos los resultados muestran que EGLE proporciona un buen balance entre una pequeña sobrecarga de mensajes de control y un alto ratio de entrega superior al 90% incluso en escenarios reales con un 100% de errores de localización.

Finalmente, se presenta SBGR, un algoritmo geográfico auto-protegido que es capaz de encaminar mensajes de datos incluso en escenarios donde existen atacantes. SBGR proporciona dos mecanismos de protección simples que solo requieren que los dispositivos sensores almacenen temporalmente el estado de los mensajes encaminados en su área de cobertura con el objetivo de soportar los recursos limitados característicos de las redes de sensores. La evaluación de SBGR es realizada en comparación con SIGF (*Secure Implicit Geographic Forwarding*), el algoritmo de encaminamiento geográfico más eficiente y seguro. Tanto los experimentos en simulaciones como en el escenario real muestran que SBGR mejora el rendimiento de SIGF en términos de ratio de mensajes entregados y reducida sobrecarga de mensajes de control.

Abstract

Wireless Sensor Networks (WSNs) belong to the top-ten of emerging technologies for the 21st century. Recent advances in mechanical and microelectronic circuits enable more powerful, smaller and cheaper wireless sensor devices. One of the assets of WSNs is their easy deployment of a large number of autonomous devices being able to monitor physical and chemical phenomena from remote wide areas. This technology provides unprecedented opportunities in control and monitoring for all type of environments.

For WSNs, there is a high diversity of potential applications in both military and civil contexts. In these applications, sensor devices distribute the captured information without the need of expensive communication infrastructure. Sensor devices employ wireless radios to communicate the information from a source to a destination following a multihop forwarding approach. To establish multihop communications, sensor devices require routing algorithms adapted to specific properties of WSNs.

The development of routing algorithms presents important challenges in WSNs. These networks are formed by thousands of nodes equipped with constrained resources in terms of computing, communication and energy. Moreover, wireless communication is the energy bottleneck limiting the autonomy of WSNs. For these reasons, routing protocols must be scalable for an increasing number of nodes and efficient in the number of transmitted messages. So, existing routing algorithms discovering routes based on flooding mechanisms produce an excessive transmission overhead and are not suitable for WSNs. To solve these problems, a novel routing paradigm based on the nodes' positions

has been proposed for WSNs. Geographic routing exploits location information available for the majority of WSNs applications. So sensor nodes take routing decisions employing the positions of their 1-hop neighbors. The localized design of geographic routing scales well with the number of nodes and decreases the requirements of processing, bandwidth and energy.

However geographic routing algorithms have been designed and evaluated considering perfect radio communications. Under realistic WSNs, geographic routing suffers from wireless communication errors such as interferences, collisions, packets losses, etc. Moreover to easy the deployment, WSNs applications employ distributed localization systems which produce positions inaccuracy degrading severely the performance of geographic routing. Geographic algorithms also neglect that military applications of WSNs are deployed in unsafe areas which are prone to routing attacks exploiting the open wireless medium to interrupt communications.

Based on the deficiencies of existing geographic protocols, this thesis is focused on providing reliable routing algorithms for realistic WSN deployments. First we study in detail the effects of realistic conditions in the performance of geographic routing. Then, we propose three enhanced algorithms being able to deal with communications errors, positions inaccuracy and routing attacks. To validate all algorithms proposed for realistic WSNs, we employ two different approaches: simulation and testbed experiments. The simulated experiment is done to demonstrate the scalability and efficiency of these protocols in networks with a large number of nodes. The second analysis is performed in order to assess the reliability and robustness of the protocols in a real network.

Concretely, we propose BOSS, a geographic routing algorithm supporting wireless communications errors in order to guarantee the packet delivery and reduce the transmission overhead. The performance of BOSS is compared against two of the most efficient geographic algorithms: IGF (*Implicit Geographic Forwarding*) and BLR (*Beacon-Less Geographic Routing*). Both simulated and testbed experiments demonstrated that BOSS outperforms IGF and BLR in terms of delivery reliability, bandwidth efficiency and end-to-end performance.

Moreover, all results confirm that BOSS adapts well to the inherent problems of wireless communications in WSNs.

After dealing with communications errors, we provide a BOSS extension called EGLE which is an effective geographic routing solution supporting inaccurate positions. EGLE is compared against MER (*Maximum Expectation within Transmission Range*) which is the best-performance geographic algorithm considering location errors in routing decisions. In simulated and testbed evaluations, EGLE enhances the performance of MER in both delivery reliability and bandwidth efficiency. All results show that EGLE provides a good balance between little control overhead and high delivery ratio (above the 90% even in real networks with 100% of location errors).

Finally, we present SBGR, a self-protected geographic algorithm being able to route data packets even in scenarios where malicious nodes are present. SBGR provides two simple protection mechanisms requiring only that nodes store temporally the status of forwarded packets in their coverage areas in order to support constrained resources in WSNs. The SBGR evaluation is performed in comparison with SIGF (*Secure Implicit Geographic Forwarding*), the most efficient and secure geographic protocol. All simulated and testbed experiments show that SBGR outperforms SIGF in terms of packet delivery ratio and reduced control overhead.

Outline of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the concept of Wireless Sensor Network (WSN) and shows the importance of the research works associated with this technology. In the last years, the WSN technology has become a hot topic not only for researchers but also for the industry. This chapter describes the WSNs technology and the problems that the thesis pretends to solve. Moreover, we define the objectives and the methodology employed to develop the thesis in order to achieve suitable solutions for realistic scenarios.

In this chapter, the main goals are:

- Introducing the WSNs technology.
- Presenting the problems related with the design of routing algorithms in WSNs.
- Defining the specific objectives of this thesis.
- Explaining the methodology that has guided the development of the thesis.
- Enumerating the main contributions of the solutions proposed.
- Showing the structure and contents of the following chapters.

1.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) comprises a big set of unattended devices being able to monitor and control phenomena in remote wide areas. The easy deployment of tiny wireless devices avoids using hundreds meters of cables in the studied area minimizing also environment perturbation. WSNs enable a wide range of applications from environment monitoring to human health control passing through subjects such as tracking systems [1, 2, 3, 4]. Although military research motivated the first WSNs, nowadays there is a high diversification of civil applications. The main application of WSNs consists of the data collection in a distant target area for the posterior analysis and processing in a control center. The sensors integrated in each device make possible the measurement of environment parameters in the deployment place. In few years the WSNs technology will enable the sensors implantation in human bodies and the remote control of health symptoms.

These applications require autonomous and cheap sensor devices which will be enabled thanks to the advances in embedded technologies. These sensor devices are featured for the constrained hardware design based on four main components: sensors, radio, microcontroller and battery. First, sensors capture information associated with the environment of the deployment area. There are many kinds of sensors to measure physical and chemical parameters such as temperature, humidity, pressure, etc. Second, a wireless radio interface permits to communicate with other sensor devices for sending and receiving data. Third, a microcontroller enables managing the data captured and the communication task. Fourth, a battery is the common energy source with limited capacity supplying the rest of components. Note that the energy consumption is crucial for sensors nodes because of determining their lifetime.

In WSNs, devices provide automatic collection and distribution of the captured information without the need of expensive communication infrastructure. Sensor devices, so-called nodes, employ wireless radio interfaces to interconnect the network and generate a distributed communication infrastructure. Each node plays a dual role as data source and wireless router. In large networks, a source

node is not able to transmit information packets directly to a distant destination located outside its radio range. Thus nodes collaborate to route the packets from a source to a destination following a multihop forwarding approach. In multihop communication, the node currently holding the packet must select the next hop among all its neighboring nodes located inside its radio range based on a routing criterion. A routing decision consists of selecting the best neighbor being able to continue the forwarding of the information towards the destination. To take these decisions, sensor nodes need routing algorithms that determine the way to discover neighbors and select the best candidate as next hop.

1.2 Motivation

In WSNs, building efficient and scalable routing algorithms is a very difficult task because of the constrained resources and high number of nodes. In the literature, many routing protocols have been developed to establish multihop communications based on different paradigms such as data-centric, hierarchical and QoS (Quality of Service). Nevertheless these paradigms require maintaining routing tables or flooding route discovery packets which generate an undesirable overhead in dense WSNs containing hundreds or thousands nodes, as demonstrated in [5].

To provide multihop communication in dense WSNs, geographic routing (GR) has been proposed as the most scalable and efficient solution [6, 7]. GR employs nodes positions to take routing decisions. This is not an issue, because in WSNs, positions are basic information for any node. For instance, a measured data (e.g. temperature) is not useful without the position where it was measured. In GR, each node takes routing decisions based on its position and the positions of its 1-hop neighbors. Thus GR scales well to the network density and the number of nodes. And each node requires low energy and low computation capacities to forward data packets towards the destination. To guarantee the data delivery, most geographic algorithms combines greedy and face strategies. In greedy mode each node routes the packet to its neighbor located closest to the destination

reducing the distance in each hop. When the packet reaches a node that has no closer neighbors to the destination than itself, and then it becomes a local maximum and a void area appears. In void areas face routing is utilized to route around the perimeter of the void till reaching nodes located closer than the local maximum to the destination where greedy routing can continue. The greedy-face-greedy process is repeated until the packet finds the destination. To take routing decisions, each packet incorporates the destination position and nodes need to know only local information about the position of 1-hop neighbors. To discover neighborhood information, geographic algorithms consider that nodes exchange 1-hop control messages, called beacons. However periodic beacon transmissions waste resources, i.e. energy and bandwidth, even in nodes not taking part in any routing process. To avoid these periodic transmissions, recent beaconless algorithms have proposed reactive solutions for discovering 1-hop neighbors and selecting the next hop.

However most geographic protocols are designed and simulated considering wireless sensor networks with perfect communications. These protocols often assume perfect wireless links and fixed radio ranges. Recent experiments [8, 9] demonstrated that real wireless communications suffer from frequent problems such as interferences, collisions, etc. These communications errors cause severe damages in the routing protocol performance in terms of packet losses and retransmissions.

On the other hand, geographic protocols have assumed perfect location information. They neglect the common inaccuracy produced by positioning systems employed in real deployments [10]. Recent studies [11] have demonstrated that existing geographic solutions are ineffective in networks with inaccurate positions. Both greedy and face strategies experiment a huge increment of packets losses, when the location error increases [12]. Concretely, in greedy strategy the main reason of dropped packets is void areas, and 90% happens in the destination coverage area [13].

In addition, the perfect conditions of wireless communications and location systems, most geographic routing algorithms consider that WSNs are deployed

in secure areas. Nevertheless open wireless medium is prone to be attacked by malicious nodes which want to avoid the communication among sensor nodes [14, 15]. Routing attackers can act as legitimate neighbors taking part in the routing process in order to achieve being selected as the next hop. Concretely, in geographic routing a sinkhole attacker exploits beaconless schemes for discovering neighborhood. Moreover a sybil attacker exploits routing decisions based on positions and creates fake positions to pretend being the best forwarder for all packets forwarded within its radio range. Once an attacker is selected as the next forwarder, it can drop the data packet.

In conclusion, geographic routing is the most scalable and efficient solution which requires effective mechanisms to deal with communication errors, location inaccuracy and routing attacks. Those are the objectives of this thesis, and we elaborate them in the next sections.

1.3 Objectives

The main objective of the thesis is focused on the design and development of scalable and reliable routing protocols adapted to the specific properties of WSNs. To do that, we consider realistic WSNs scenarios where there are communication errors, position inaccuracies and routing attacks. Moreover sensor nodes posses constrained resources, thus the designed algorithms must be simple. Addressing the main objective requires the achievement of the following subobjectives.

1. Studying the features of real wireless communications. Studying the problem related with packet losses during the forwarding process. Adapting of the geographic routing algorithms to provide reliable routing for realistic conditions.

2. Studying the inaccuracy of positioning systems and the effects in geographic routing. Studying the main issues related to inaccurate positions of geographic routing. Developing mechanisms that mitigate the influence of location errors and provide a high packet delivery.

3. Studying the routing attacks that affects geographic routing protocols. Studying the effects of the main attacks and their behavior to interrupt the communication. Designing an effective algorithm to avoid the effects of routing attacks in order to guarantee multihop communications in unsafe environments.

To cover each subobjective, we perform an extensive analysis of the problem and the existing solutions proposed in the literature. Based on our previous analysis, we design an enhanced solution. To evaluate our solution, we compare against the most relevant existing proposals to confirm the scalability and efficiency in large networks with thousands of nodes. Finally, we validate the goodness of our solution in a small testbed with realistic sensor devices.

1.4 Methodology

In realistic deployments of WSNs, multiple factors can affect the performance of geographic routing algorithms. First, the communication is performed by wireless radio interfaces which are affected by interferences and collisions. Second, the positioning system generates inaccurate positions that degrade geographic routing decisions. Third, the open wireless medium is prone to routing attackers that interrupt the communication among sensor nodes. Considering all these variables at the same time makes very difficult the study and the development of reliable algorithms. For this reason, we followed an increasing realism methodology. We start with simple models with perfect assumptions, then we reduce the assumptions and include realistic variables that increase the realism of the model.

As most geographic routing algorithms, we consider the well-known unit disk graph (UDG) model to represent WSNs. In this model, WSNs are represented as a graph where each sensor device corresponds to as a node and the communication between two devices is represented as a link between these nodes. This model assumes that sensor devices provide wireless radio interfaces with uniform transmission ranges and omni-directional antennas for receiving signals. Thus,

nodes located inside the radio range can communicate directly without errors and its links are bidirectional.

Based on this simplified model, we increase progressively the complexity and include previous variables mentioned. First, we include wireless communication errors to obtain a more realistic model. Second, we add in the model the inaccuracy of positioning system represented as an estimated position with an associated error. Finally the last model considers the presence of routing attackers in unsafe WSNs deployments.

The development of routing algorithms requires extensive evaluations to validate their performance in realistic conditions of WSNs. In this thesis, we evaluate routing algorithms employing two different approaches: simulation and testbed networks. The first study is performed to assess the scalability and efficiency of these protocols in networks with high amount of nodes. To do that, we develop routing protocols in TinyOS [16] which is the most used operation system for developing of WSNs applications. In TinyOS, we implement the protocols using the NesC programming language which is a component-oriented variant of C language. TinyOS provides a network simulator called TOSSIM [17] enabling the emulation of communications between thousands of sensor nodes. TOSSIM models the wireless communication with a probabilistic MAC layer including collisions and interferences. However, these models do not consider the typical problems in wireless communications such as radio range variability and link asymmetry. Therefore, we compare routing protocols in a testbed scenario to validate their reliability and robustness in realistic deployments. The testbed scenario consists of 35 sensor nodes distributed within the first floor of the Computer Science building at the University of Murcia, described in Section 3.3.4. Using both simulated and testbed experiments, we evaluate the algorithms proposed in the thesis and demonstrate the improvements provided for realistic WSNs scenarios.

Note that we have made an effort to guarantee that our algorithms are well designed for the operation in realistic WSNs. Next section lists the main contributions achieved from this thesis.

1.5 Contributions

In the following we describe the main results obtained during the development of the thesis:

- **BOSS**: Beacon-less On demand Strategy Scheme. The design of BOSS provides several effective mechanisms to deal with error-prone wireless communications. BOSS is based on an empirical analysis with real wireless radios determining the strong relationship between big packets and low reception probabilities. Concretely BOSS includes an enhanced beaconless discovering scheme discarding neighbors with low-reception links which are sensitive to generate packets losses and retransmissions. Moreover BOSS adds a delay function combining greedy and face strategies and a passive acknowledgment mechanism guaranteeing the hop-to-hop delivery in order to reduce collisions and control overhead.

- **EGLE**: Effective Greedy routing protocol supporting Location Errors. EGLE considers the presence of inaccurate positions during routing decisions in greedy mode. EGLE provides three operation modes based on location errors to mitigate the packets losses for void areas. A greedy selection heuristic prevents reaching local maximums. An alternative forwarding in a limited region of the local maximum permits to exit the void areas. And, a lightweight broadcasting strategy propagates the data packet in a reduced area near the estimated position of the destination to guarantee the delivery. Finally, EGLE combines the beaconless nature of the protocol and a neighborhood discovery function to reduce the transmission overhead.

- **SBGR**: Self-Protected Beaconless Geographic Routing. SBGR defenses against routing attacks to reduce the performance degradation of geographic routing. Moreover SBGR provides two simple forwarding modes based on a fully distributed scheme and a reduced flooding mechanism. First, nodes forward data packets by competing distributively in order to prevent sinkhole attackers intercepting and dropping packets. Second, nodes

detecting a sybil attacker flood data packets in a limited area to guarantee their advances towards the destination.

The results of the thesis have been presented to the research community in international conferences and journals. Section 6.2 lists the main publications.

1.6 Organization

The thesis is divided into six chapters. Below we describe the five remaining chapters.

Chapter 2 presents the state-of-art in wireless sensor networks and discusses existing routing algorithms. We show the main properties of WSNs and the fundamental requirements that routing algorithms must cover. Based on the specific WSNs requirements, we study and compare the multihop routing paradigms proposed in the literature. Finally, we present geographic routing which is the most efficient and scalable paradigm for WSNs and the base of the algorithms developed in the thesis.

Chapter 3 studies failure factors in realistic wireless communications to design a reliable geographic routing solution. We divide the chapter into three parts. The first part shows the most efficient solutions of geographic routing and their limitations based on their assumption of perfect communication. In the second part, we make an empirical analysis of realistic wireless radios to understand better the behavior of wireless communications. Based on our analysis, we include communication errors in the simplified UDG model and design BOSS to improve the forwarding process for increasing reliability as well as reducing traffic overhead The third part compares BOSS against two of the most efficient geographic algorithms via extensive simulations and a real-testbed network. The results of both evaluations confirm that BOSS is the most scalable, efficient and reliable protocol for WSNs with a large number of nodes and realistic wireless communications.

Chapter 4 is focused on mitigating the performance degradation of geographic routing in scenarios with location errors. First, considering location errors we

classify the main error factors of geographic routing and the solutions proposed in the literature. Second, we analyze in detail the effects of location errors in two relevant geographic routing protocols and their failure conditions. Third, we include the location errors in the realistic communication model and extend BOSS to provide an effective geographic solution (EGLE) to deal with the main causes of routing failures. Fourth, we employ simulations and testbed experiments to evaluate EGLE comparing with BOSS and MER, which is the best-performance protocol considering location errors. The experiments demonstrate that EGLE outperforms these protocols in all location errors scenarios and provides a good balance among delivery reliability and communication efficiency.

Chapter 5 takes into account unsafe scenarios with the presence of malicious nodes. First, we describe the routing attacks determined in the literature and discuss the limitations of existing secure mechanisms according to the WSNs requirements. Second, we study the behavior of the main routing attacks that affect geographic algorithms. Third, considering realistic communications we model routing attackers and design a self-protected geographic algorithm (SBGR). Fourth, the SBGR evaluation is performed by simulation and testbed experiments comparing with BOSS and SIGF which provides secure mechanisms to defense against routing attackers. All results show that SBGR improves the performance of SIGF and achieves an efficient and robust communication solution for WSNs deployed in insecure environments.

Finally, the chapter 6 presents the results obtained in the thesis and the conclusions extracted. Moreover, it shows the main publications performed during the thesis development.

Chapter 2

Wireless Sensor Networks and Multihop Routing

This chapter presents the state-of-art in Wireless Sensor Networks (WSNs) and discusses the main paradigms of multihop routing. It describes the history of WSNs based on the first deployments of military sensor networks and the earlier developments in ad-hoc wireless communications. The communication architecture used in WSNs is shown considering also the hardware components of individual sensor devices and the importance of advances in embedded technologies to enable the application of WSNs in multiple disciplines. After describing the specific requirements of WSNs, we study and compare the multihop routing paradigms proposed in the literature. Finally, we present the fundamentals of geographic routing which the most efficient and scalable paradigm for WSNs and the base of the protocols proposed in the thesis.

In this chapter, the main goals are:

- Introducing the history and evolution of wireless sensor networks.
- Describing the networking architecture and the components of sensor nodes.
- Discussing about the specific requirements of wireless sensor networks for multihop routing.

- Comparing some of the most relevant routing paradigms for WSNs and discussing about their limitations.

- Explaining in detail the fundamentals of geographic routing and the most important algorithms proposed in this paradigm as well as introducing their main disadvantages.

2.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of a set of thousands or hundreds nodes equipped with batteries, sensing, computation and communication components. These components are integrated in a tiny cheap device which is able to be deployed in inaccessible and remote places. The power of WSNs lies in the capacity of deploying a large number of simple devices which capture data and collaborate to provide a global view of the area of interest. WSNs enable a wide range of applications from environment monitoring to human health control passing through subjects such as tracking systems. In fact, WSNs were identified in the top ten of emerging technologies for the 21st century [18].

Unlike traditional wired systems, the deployment cost of WSNs is drastically reduced. The easy installation of tiny wireless devices avoids using hundreds meters of cables in the studied area minimizing also environment perturbation. Nodes cooperate with each other to enable the dynamic adaptation to network topology changes (i.e. adding or removing nodes) without human intervention.

In comparison with mobile phones or personal digital assistants(PDA), wireless sensor nodes establish the communication without requiring a pre-existing infrastructure. Nodes provide wireless radio interfaces to interconnect the network and generate a distributed communication infrastructure. Each node acts as router to forward the information from a source node to a destination node in a multi-hop fashion. The flexibility of its distributed architecture supports network failures (i.e. broken nodes) without degrading the communication performance.

For WSN technology there is a high diversity of applications in both military and civil contexts. The main application of WSNs consists of data collection in a distant target area for the posterior analysis and processing in a central server. The sensors integrated in each node make possible the periodic measurement of physical and chemical parameters in the deployment place. For instance, recent research projects show the deployment of WSNs to monitor areas with difficult access and harsh environment conditions such as volcanoes and bridges [19, 20]. In few years the WSNs technology will enable the sensors implantation in human bodies and the remote control of health symptoms [21, 22].

The vision of WSNs is to deploy high amount of sensor nodes to cover a huge area for a reduced cost. Although current sensor nodes have resources constraints in terms of communication and processing capacities, each year the advances in mechanical and microelectronic technologies provide more powerful, smaller and cheaper embedded systems. Nowadays there are centimeters chips integrating 32-bit microcontroller, hundreds KB of memory and kilometers-range wireless radio.

Energy is the most constrained resource of sensor nodes which are supplied by batteries and provide wireless communications. Unlike phones and PDAs, sensor nodes using batteries must have a long lifetime depending on the phenomenon under study which may be even several years. The recent study of Koomey[23] showed that the energy efficiency of computing was doubled every 1.52 years from 1975 to 2009. However the radio interface is the most energy consumption component, and wireless communications are the energy bottleneck [24]. For this reason, the power autonomy of WSNs is directly related with the design of efficient routing protocols providing low communication overhead.

2.1.1 History of Sensor Networks

The origin of sensor networks was driven by military applications from 1950s to 1990s. During the Cold War in the early 1950s, the US Navy developed the first sensor network to detect and track Soviet submarines, called SOund SUrveillance System (SOSUS) [25]. SOSUS contained multiple arrays of acoustic sensors or

hydrophones deployed in the Atlantic and Pacific oceans. Large sensor arrays were wired by undersea cables to achieve a communication system.

In the 1980s and 1990s, the US Air Force developed networks of radar sensors to control planes. That networks combined ground-based radars and Airborne Warning and Control System(AWACS) [26] to provide the aircrafts detection. The AWACS planes were equipped with radars whose size were 30 feet in diameter and 6 feet thick. AWACS was also applied in military scenarios to gather information from battle fields in order to achieve tactical global views for strategic command and control communications.

Moreover the US Air Force used unattended ground sensors to detect movements of people and vehicles [27] in the Vietnam War. Seismic, acoustic, magnetic and infrared sensors were deployed to enable Air Delivered Seismic Intrusion Detector (ADSID) and Remotely Monitored Battlefield Sensor System (REMBASS). For instance, each ADSID node being 9 inches in diameter contained a seismometer and a long-range wireless interface to communicate directly to a plane. Despite that the ADSID nodes were large devices with an autonomy of only few weeks due to the high energy cost of direct communication, they successfully demonstrated the concept of wireless sensor networks.

2.1.2 History of Ad-hoc Networking

As sensor networks, the development of wireless ad-hoc networks was started by military applications, since centralized networks requiring base stations can not be deployed in battle fields where distributed communication is essential. In distributed networks nodes collaborate to route information toward a destination located outside their radio range using a multihop forwarding scheme.

The US Department of Defense, more concretely the Defense Advanced Research Projects Agency (DARPA), played a vital role in the initial development of wireless ad-hoc networks from 1970s to 1990s. In 1972, DARPA created Packet Radio Networks (PRnet), also called Ad Hoc Networks, where mobile nodes were able to forward information by packet broadcasting[28]. That nodes had a low computing capability, while their primary radios consumed high power. In 1980,

DARPA started the Distributed Sensor Networks (DSN) program to research on modern ad-hoc networks. Those networks were designed to prove wireless communications between large numbers of autonomous low-cost nodes. In 1983, DSN program upgraded the PRnet to Survivable Radio Networks (SURAN) which addressed the issues of earlier networks in order to achieve radios with lower energy requirements that enable powerful routing algorithms [29]. In 1994, DARPA funded the Global Mobile (GloMo) program to apply the technological advances of emergent Internet to wireless networks[30]. The objective was to provide the Internet advantages (i.e. robust routing protocols, mobile applications, heterogeneous environments and quality of service) to wireless mobile users. In 1997, US Army developed Tactical Internet (TI) to test the functionality of wireless ad-hoc networks in large scenarios [31]. The tested scenarios consisted of thousands mobile nodes (i.e vehicles and people) employing Internet routing algorithms adapted to wireless communications. In 1999, the US Marines created the Extending the Littoral Battlespace Advanced Concept Technology Demonstration (ELB ACTD) [32]. ELB ACTD validated the improvements of military operations in littoral areas generated by commercial WLAN (Wireless Local Area Networks) technologies.

2.1.3 Architecture of Wireless Sensor Networks

As shown Fig 2.1, the typical WSNs architecture consists of large number of nodes interconnected wirelessly with sink nodes to communicate sensing data of a phenomenon studied from a target area to a control center. The phenomenon is the subject studied in a remote and unattended area, where there is not any power supply and communication infrastructure. In the target area, sensor nodes are deployed to capture and transmit environment data over the wireless short range radio to one or multiple sink nodes. For large WSNs deployments, a sensor node is not able to transmit data directly to a distant sink destination. Each sensor node plays a dual role as data source and wireless router based on a multihop fashion. Sinks act as gateways which possess large range radios (i.e. satellite, GPRS, etc) to communicate directly sensing data to the central server. In the center server,

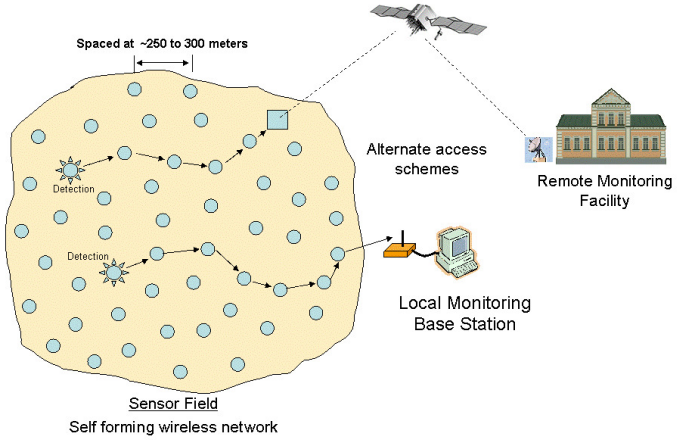


Figure 2.1: Architecture of Wireless Sensor Networks.

the gathered information is post-processed and analyzed.

2.1.4 Components of Wireless Sensor Nodes

A sensor node consists mainly of energy, sensing, processing and communication components as shown in Fig 2.2 Batteries are the most used way to provide energy supplies to sensor nodes deployed in distant and harsh environments. The lifetime of batteries must be more than the observed phenomenon duration. To recharge batteries and increase their lifetime, renewable energy sources (i.e. solar, vibration) are employed. However, these renewable sources require expensive hardware (i.e. solar panel) being prone to theft. Battery often is the unique energy source and the major weakness of sensor nodes.

Sensing module is composed of two parts: sensors hardware and an Analog to Digital Converter (ADC). Sensors measure physical and chemical parameters providing analog outputs. The ADC converts analog outputs to digital data. Processing unit contains a microcontroller with ROM program memory and RAM

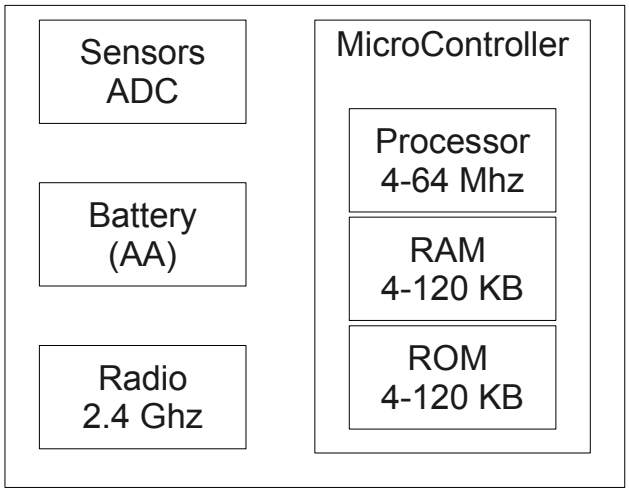


Figure 2.2: Components of Wireless Sensor Nodes.

flash memory. The microcontroller enables the aggregation and compression of sensed data in small packets to minimize the number of radio transmissions. Moreover, the microcontroller manages the communication operation. To communicate data, radio frequency antenna supports the transmission and reception over the air in a range of hundreds of meters. These three components must be provided in embedded devices with strict requirements of size and cost.

2.1.5 Advances in Embedded Technologies

The advances of embedded technologies enable the development of cheaper and smaller wireless sensor nodes. Several research approaches evolve the development of sensor nodes based on integrated circuits, thin-film structures and micro-assembly techniques for individual components. To obtain a tiny microcontroller, the System-on-Chip (SoC) technology [33] provides the possibility of integrating all computing components on a single chip. Existing commercial SoCs provide 32-bit microprocessors with hundreds KB of RAM and hundreds KB of ROM memory in few centimeters chips [34]. To measure a set of physical parameters, Micro-Electro-Mechanical Systems (MEMS)

technology [35] makes possible the integration of multiple sensors in the same CMOS circuitry [36]. For instance, a node may contain the following sensors: light, temperature, humidity, acoustic, accelerometer, magnetic, etc. To achieve low-power wireless communications, Radio Frequency (RF) technology combines transmitter and receiver circuits on a single transceiver. Actually commercial transceivers provide a data ratio of hundreds kbps for hundreds meters distances with a extremely low consumption of less than 30 mW[37]. To integrate computing, communication and sensing operations, packaging techniques enable including all these components in a reduced board. Nowadays, there are embedded devices equipping low-power microcontroller, wireless communication and sensors hardware in a portable board of few centimeters.

2.1.6 Recent Applications of Wireless Sensor Networks

Although military research motivated the first WSNs, nowadays there is a high diversification of civilian applications. Examples of civilian sensor networks range from large-scale environment monitoring to small body networks for health control. Based on previous studies [1, 2, 3, 4], we categorize sensor network applications in the following fields(see Table.2.1).

The most popular application of WSNs is environmental monitoring in large areas. The ability of WSNs to place autonomous and low-cost nodes in harsh environments without communication infrastructure provides real-time data collection directly from interesting areas. Real-time environmental data is key to forecast the upcoming phenomenon and send prompt warnings. Some application examples are soil moisture monitoring [38], solar radiation mapping [39], glacial control and climate change [40], environmental observation and forecasting in rivers [41], forest fire alarm [42] and landscape flooding alarm [43] In additional, habitat study is one of the driving applications for WSNs [44]. Unattended and lightweight nodes are ideal platforms to gather bio-physical or bio-chemical information from fauna and flora without the perturbation of studied entities such as Birds [45], Redwoods [46], Storm Petrels [45], Zebras [47], and Oysters [48].

The WSNs technology is also the cornerstone for many industrial applications.

Field	Application
Environment	Climate Monitoring Fire-Alarm System Flooding Detection
Security	Military Surveillance Intrusion Detection
Industrial	Structures Monitoring Machines Control Inventory System
Quality-of-Life	Health Monitoring Smart Home/Office

Table 2.1: Applications of Wireless Sensor Networks.

Manufacturing plants and general engineering facilities employ the WSNs technology to ensure product quality as well as efficient and safe operation. Autonomous and cheap sensor nodes replace traditional processes of maintenance and production which are manual and expensive. Such networks have already been developed widely in machine status monitoring [49] and inventory systems [50]. Status monitoring for civil structures (i.e. bridges) is a relevant studied field for research and industry. To measure the effects of wind or earthquake, traditional mechanisms provide acoustic emission, ultrasonic testing, and radar tomography. The advance of WSNs has resulted in small and easily-deployed sensor devices for many tasks related to structural status monitoring [51, 52, 53, 20]. Moreover, security applications focus the usage of WSNs on military surveillance missions and intrusion detection systems. Such missions often involve a high human risk and require a high stealthiness degree. For these reasons the capability to deploy unattended sensor nodes is vital. WSNs enable the acquisition and verification of states and positions of enemies in hostile regions [54, 55]. Moreover several researches showed the application of WSNs to Intrusion Detection Systems (IDS) [56, 57, 58], since sensor nodes

often maintain neighborhood information allowing the detection of anomalous behaviors for malicious nodes. This specific property of WSNs makes them ideal for a detection-based security scheme.

Nowadays, WSNs are also applied for the enhancement of the human life quality. For instance, several electrical devices are equipped with sensor nodes to achieve the remote home control from anywhere[59]. In additional, sensor nodes can be found in human bodies to develop symptoms monitoring and alert systems [22, 21] and in smart classrooms to evaluate children's learning environments [60]. In the future, many other applications will be possible due to the WSNs evolution.

2.1.7 Evolution of Wireless Sensor Networks

The proliferation of small and portable devices with high computing performance and the pocket size is very common such as phones and PDAs. Nowadays the manufacture of centimeters embedded device is possible using Commercial Off-The-Shelf (COTS) [61] components. Considering the Moore's Law [62], the future advances in mechanic and electronic technologies will increase complex electronic functions in millimeters size with minimum weight. Below we describe the recent evolution of wireless sensor nodes with lower power consumption and higher computing capacities.

In 1996 UCLA (University of California, Los Angeles) and Rockwell Science Center manufactured the first Low Power Wireless Integrated Micro-sensors (LWIMs) [63]. LWIMs employed low cost CMOS components to demonstrate the integration of a power source, a processor, a radio interface and multiple sensors in a small device. Its radio interface operated at a single frequency within the 902-928 MHz ISM band obtaining a low data rate (10 kbps) with a short range of less than 30 meters. An enhanced version of LWIMs were developed two years later, named Wireless Integrated Network Sensors (WINS) [64]. WINS presented a novel design for low cost and low power radio communication. The wireless radio supported a 100 kbps transmission rate with an adjustable consumption between 1 to 100 mW. Each WINS node contained a powerful 32-bit processor

from Intel with 4 MB flash memory and 1 MB SRAM. The processor of WINS provided high computing capabilities, but required excessive power consumption above 200 mW in active mode and 0.8 mW in sleeping mode.

In 1999 University of California at Berkeley released the Smart Dust project [65] to develop smaller and cheaper nodes with less power consumption. The objective was to produce fully functional sensor nodes with millimeter size, also called Motes. The fist mote, WeC, included a radio transceiver (RFM TR1000) supporting a 10 kbps bit rate with 36 mW transmitting power and 9 mW receiving power. WeC possessed an 8-bit 4-MHz microcontroller of Atmel company (AVR AT90S2313) with 512 Bytes RAM and 128 Bytes flash ROM. That microcontroller needed a low power consumption of 15 mW in active state and 45 μW in sleeping state. Moreover, WeC provided a sensors board to measure temperature and light. After the WeC, UC Berkeley designed Rene and Dot motes [66] manufactured by Crossbow company in 1999 and 2000, respectively.Although Rene motes had the same WeC design, they integrated more memory and a 51 pin connector to expand the sensors board. While Dot motes contained the ATMEGA163 microcontroller providing four times more RAM than the Rene processor.

In 2001-2003 UC Berkeley and Crossbow developed the Mica motes family to improve the Rene and Dot performance in terms of memory and radio [67]. The Mica family consisted of Mica, Mica2, Mica2Dot and MicaZ motes. Particularly the first version of Mica mote used an 8-bit 4-MHz microcontroller with 4 KB RAM and 128 KB ROM (ATmegal03L). Mica also had the RFM TR1000 radio enabling four times more bandwidth than WeC with a similar energy consumption. In 2002 Mica, Mica2 and Mica2Dot were manufactured with an enhanced microprocessor (ATmegal28L) decreasing the energy consumption with 33 mW and 75 μW in active and sleeping mode, respectively. Those motes incorporated a radio chip (Chipcon CC1000) which operated in the 400 MHz and 900 MHz bands and used FSK frequency modulation. In 2003 the latest member of the family, MicaZ, was produced with a 250 kbps radio module (Chipcon CC2420) supporting the 802.15.4 protocol and data encryption.

In 2004 UC Berkeley designed Telos [68] as an ultra low power consumption mote. Telos incorporated a faster 8 MHz processor needing only 3 mW in active mode and 15 μW in sleep mode. Its radio module provided 250 kbps for the 2.4 GHz band and 802.15.4 compliant. The radio was printed in the circuit board to reduce its cost. In additional, the board contained an USB port to facilitate the debug and re-programming processes. The board also included humidity, temperature, and light sensors to monitor the environment. Following the Telos design, in 2005 the Moteiv company (now Sentilla) developed the Tmote-Sky [69]. The main components of Tmote-Sky were a low-power 16-bit microcontroller (Texas Instrument MSP430F1611) with 10 KB of RAM and 48 KB of ROM memory and a low-power radio chip (Chipcon CC2420). Tmote-sky provided an ultra-low power architecture that incorporated on-board sensors (i.e. light, temperature and humidity) and decreased cost and size.

Recently Jennic company presented one of the last proposals for high performance sensor node on a single low-cost chip, named JN5148 [34]. The chip integrates a powerful 32-bit RISC microcontroller with large memory 128 KB ROM and 128 KB RAM. Its radio transceiver supports a 802.15.4 compliant and a 667 kbps data rate at 2.4 GHz band with a ultra-low consumption below 18 mW. The JN5148 chip represents the future generation of wireless sensor nodes that will be manufactured for cents and deployed in millions.

Beside hardware technologies, several research developed power efficient communication protocols for the emerging field of wireless ad-hoc networks (WANs). In 1997, the IEEE 802.11 Working Group introduced the first communication standard for Wireless Local Area Networks (WLANs), called 802.11 protocol [70]. The 802.11 protocol provided a high data rate and a Carrier Sense Multiple Access (CSMA) mechanism for Medium Access Control (MAC) layer. The design was oriented for wireless networks composed of laptops and PDAs with high power and computing capacities. In 2001, the IEEE Working Group presented the 802.15.4 upgraded version [71] for wireless communications with short range and low data rate requirements. The 802.15.4 MAC protocol was specifically designed for low power devices of WSNs.

2.1.8 Requirements of Wireless Sensor Networks

This section summarizes the main WSNs requirements affecting multihop routing protocols. These requirements often are interrelated, even sometimes the performance of one requirement is opposite to another one. For instance, the increment of security produces the reduction of the energy efficiency. In WSNs, the design of routing protocols is influenced by the following requirements:

- Energy is the most critical requirement. Sensor nodes are placed in distant positions where electric energy is unavailable. Thus, nodes using batteries must operate autonomously during months or even years. The battery lifetime is determined by the power consumption of three main components (i.e. processing, sensing and communication). These components are disabled during idle states in order to minimize the power consumption. However, wireless communication consumes the majority of the energy, in particular the transmission is the most energy consuming task [72]. To maximize the WSNs lifetime, efficient routing protocols must minimize the number of transmissions.

- Scalability is a specific property of WSNs where thousands or hundreds of nodes are deployed to sense a target area. For the limited sensing coverage of nodes, the density of neighbor inside the same radio range is from tens to hundreds [73]. This factor requires distributed protocols where nodes take routing decisions using only local neighborhood information.

- Topology is a relevant factor for WSNs even in applications where nodes are stationary. In static WSNs, the topology may change by adding and removing nodes. Moreover topology changes happen because of failures in nodes due to physical damage or power lack. Other important factor affecting the network topology is the fluctuation of link qualities between nodes with wireless radios. These topology changes break multihop paths and damages routing protocols reducing their performances in terms of latency, efficiency and reliability [74].

- Connectivity is of great importance in dense WSNs. The connectivity is defined as the capacity of establishing communication between any two individual nodes. To achieve connectivity among all nodes, WSNs require localized communication protocols avoiding the overhead of routes maintenance techniques and flooding discovery mechanisms [5].

- Security is a key aspect in some WSNs applications such as Intrusion Detection Systems (IDS) where the delivery of warning packets is essential. An attacker tries to avoid the proper operation of IDS by interfering warning packets. To guarantee multihop communications in presence of attackers, many cryptographic algorithms have been proposed [75, 76]. However cryptographic algorithms need high resources in terms of computation, energy and bandwidth. In addition, cryptography is insufficient against attackers being able to obtain private keys and compromise multihop protocols [14].

Considering these WSNs requirements, several routing techniques have been proposed in the literature [77, 74]. An overview of multihop routing proposals for WSNs is presented in the next section.

2.2 Multihop Routing in WSNs

This section summarizes the state-of-the-art of the main routing paradigms for WSNs. According to several studies [74, 77, 78, 79] we classify routing protocols into four kinds: data-centric (or attribute-based), hierarchical, QoS (Quality of Service) and geographic (or location-based). In data-centric routing all nodes play the same role and provide the same operations, and the routing process is based on query messages. In hierarchical routing there are two types of nodes: cluster-head and normal nodes. Cluster-head nodes aggregate data from normal nodes to decrease data traffic transmitted toward the sink. In QoS routing sensor nodes forward packets in order to guarantee network parameters such as energy efficiency, effective sample rate and bounded delay. In geographic routing

Paradigms	Protocols
Data-centric	SPIN Directed-Diffusion Rumor-Routing COUGAR ACQUIRE
Hierarchical	LEACH PEGASIS TEEN and APTEEN SOP
QoS	SPEED SAR
Geographic	GFG

Table 2.2: Routing Paradigms for Wireless Sensor Networks.

nodes exploit their position information to forward packets toward the destination. Nodes take routing decisions according to the positions of their neighbors and the position of the destination. For each paradigm, we present several examples of routing protocols below (see Table 2.2).

2.2.1 Data-centric routing

Data-centric routing is based on the sending of query messages from a single sink to request information of multiple sensor nodes. In dense WSNs, the sink has severe difficulty to select sensor nodes. Since that the use of global identifiers is impossible because of the large number of deployed nodes. Unlike traditional address-based routing where routes are created using network addresses of nodes, in data-centric routing query messages contain attributes to specify requested data. So, the sink sends a query message to sensor nodes placed in a concrete region and waits for their data answers. Moreover nodes aggregate data during the forwarding

to decrease the information redundancy.

Two earlier data-centric proposals, SPIN [80] and directed diffusion [81], considered data negotiation among nodes to avoid redundancy and save energy. The negotiation concept motivated several protocols such as Rumor-routing [82], COUGAR [83] and ACQUIRE [84]. These protocols and their main ideas are presented below.

SPIN (Sensor Protocols for Information via Negotiation)

Heinzelman et al. [80] proposed SPIN which is a family of adaptive protocols for data-centric routing. The main idea is to distribute data from sources in the whole network. SPIN considers that sensor nodes located close to each other may obtain similar data which can be aggregated. The protocol identifies data by assigning high-level names, called meta-data. Metadata are determined by each application to increase the flexibility. Nodes employ metadata to negotiate the data transmission with their neighbors in order to eliminate redundancy.

The SPIN communication follows 3 phases using three types of messages: ADV, REQ and DATA. First, each node obtaining new data advertises the specific metadata in an ADV message. Neighbors interested in the incoming metadata send a REQ message to request the DATA message with the data. The communication process is repeated till the data is fully diffused in the network.

The aggregation process of SPIN decreases the network traffic and power consumption. Moreover SPIN behaves well with topological changes owning to nodes require only 1-hop neighborhood information. Nevertheless, the advertisement scheme does not ensure data delivery when neighbors are not interested.

Directed Diffusion

Intanagonwiwat et al. [81] proposed directed diffusion, a data-centric protocol for WSNs. Directed diffusion enables data distribution by a naming scheme of attribute-value pairs. This scheme permits the data aggregation from various sources in order to eliminate redundancy.

Directed diffusion uses 3 phases: interests requesting, gradients building and data dissemination. Using attribute-value pairs the sink floods a query message with a data interest in the whole network. During the interest propagation each node stores a gradient which indicates the previous query sender. The propagation process constructs various paths between the sink and sources. Finally, the sink chooses the best path which is strengthened by the initial interest retransmission. Sources employ the chosen paths to transmit data toward the sink. Intermediate nodes aggregate data from various sources in order to decrease the transmission overhead and power consumption. In addition directed diffusion provides a repairing technique for broken paths.

Unlike SPIN, direct diffusion is an on-demand protocol without global network information. Direct diffusion provides an aggregation tree to communicate data from sources to the sink. The usage of the best paths is ideal to support applications needing high-rate data flows. Nevertheless, the reactive query technique is inefficient in scenarios where there is periodic data delivery (i.e. environmental monitoring) or the route is utilized only once (i.e. alert systems).

Rumor-Routing

Braginsky et al. presented a variant of directed diffusion, called Rumor routing [82]. The objective is to decrease the high overhead of queries flooding. Rumor routing is designed to request few interest events. In rumor routing, events are distributed in the network utilizing long-lifetime packets, called agents. Each sensor node observing an event stores it in a table and produces an agent. Agents traverse the network and propagate data of local events to far away nodes which store the events in their tables. Employing their events table, sensor nodes answer to particular event queries of the sink node.

Unlike directed diffusion, rumor routing requires no flooding to distribute events minimizing the energy consumption and communication overhead. Results through simulation showed that rumor routing enhances significantly power efficiency and supports nodes failures. Nevertheless if the application needs many events, the maintenance cost of agents and event tables becomes impracticable.

COUGAR

Yao et al. introduced an alternative data-centric protocol called COUGAR [83] which considers the network as a distributed database system. COUGAR utilizes declarative queries to abstract the queries computation from the network layer. The abstraction is provided through a new query layer between the application and network layers. Based on database systems, COUGAR proposes an architecture where nodes choose a leader to aggregate and transmit the information to the sink. The sink is in charge of producing a query plan that incorporates the information of data flows. Also, this plan explains how to choose a leader for a specific query. The proposed architecture considers the data aggregation in all nodes to decrease the power consumption and the traffic toward the leaders.

Nevertheless, the independent layer for queries computation possess some disadvantages. First, including query layer on nodes might add an extra overhead in terms of routes storage. Second, the data aggregation in all nodes needs time synchronization. Third, leader nodes must be kept to avoid communication failures.

ACQUIRE

In [84], Sadagopan et al. proposed a data-centric technique for querying sensor networks called ACtive QUery forwarding In sensoR nEtworks (ACQUIRE). Like COUGAR, ACQUIRE sees the network as a distributed database where a query message contains multiple sub-queries. In ACQUIRE, the sink node disseminates a query in the network. During the dissemination each sensor node tries to answer utilizing its pre-cached information. When the pre-cached information is not updated, the node transmits the query to its d-hops neighbors. Once d-hop neighbors resolve fully the query, they answer to the sink. This process permits simple and complex queries.

Unlike data-centric approaches employing flooding, ACQUIRE provides a power efficient querying mechanism by adjusting the variable d. Note that when d is equal to network diameter, the algorithm behaves similar to flooding.

Otherwise if d is too small, the query travels only few hops. Authors presented a mathematical modeling which finds an optimal value of $d = 4$ in well-distributed networks. Nevertheless, results of the mathematical model are not validated through simulations.

2.2.2 Hierarchical routing

Hierarchical or cluster-based routing is a well-known technique to provide scalability and power efficiency in WSNs. The idea is to aggregate data from various nodes which belong to a specific cluster in order to decrease the traffic toward the sink. Hierarchical routing consists of two stages: first-stage is employed for forming cluster heads (CHs) and second-stage is utilized for routing. Cluster formation is based on the power and proximity of nodes to the observed region. In cluster-based architectures high-power nodes act as CHs to aggregate and transmit the information from low-power nodes sensing the target phenomenon.

For WSNs one of the first hierarchical routing protocols is LEACH [85]. Based on LEACH, several algorithms have been developed such as PEGASIS [86] and TEEN [87]. Moreover some alternative hierarchical protocols have been proposed (i.e. SOP [88]). Below, we describe these hierarchical routing protocols.

LEACH

Heinzelman et al. [85] proposed Low Energy Adaptive Clustering Hierarchy (LEACH). LEACH forms clusters of sensor nodes according to the received signal strength and employ local cluster heads (CHs) as relays toward the sink. It employs a combined TDMA/CDMA scheme to decrease the number of intra-cluster and inter-cluster collisions. In LEACH, a distributed formation scheme chooses arbitrarily a few nodes as CHs and rotates the CH role among all nodes to balance the power consumption and maximize the network lifetime. In LEACH each CH aggregates data received from sensor nodes belonging to the specific cluster and transmits a compressed packet to the sink.

The operation of LEACH is divided into two stages: CHs organization and data transmission. The duration of data transmission is longer than the duration of CHs organization in order to reduce the control overhead.

During the organization each node elects a random number, r, between 0 and 1. A node (n) acts as CH for the actual round if r is less than the following threshold ($T(n)$):

$$T(n) = \frac{p}{1 - p(r \ mod \ (1/p))} \ \ if \ n \in G. \tag{2.1}$$

where p is the CHs percentage over the total number of nodes, r is the actual round, and G is the group of non-cluster head nodes in the last $(1/p)$ rounds. Authors estimated that the optimal value of p is 0.05.

The CHs flood advertisement packets to the remaining sensor nodes. According to the signal strength of the advertisements received, each sensor node chooses the best CH and transmits a joining packet to the chosen CH. Once a CH received all joining packets, this CH creates a TDMA scheme considering the number of joining nodes and informs them about their transmission time slot. During the transmission stage each node senses and sends data to its chosen CH node. Each CH aggregates the received data and transmits the compressed information to the sink. To decrease collisions and interferences each CH communicates employing different CDMA codes. Periodically, the network runs again the organization process to choose new CHs.

LEACH is fully distributed and needs no global network knowledge. When some CH nodes die, LEACH enables a dynamic cluster formation increasing the network lifetime. Nevertheless LEACH requires that sensor nodes has high computing capacity to support two MAC layers In additional the dynamic formation requires extra overhead (i.e. CH changes, advertisement packets, etc.) increasing the power consumption.

PEGASIS

Lindsey et al. [86] presented an improved version of the LEACH protocol named Power-efficient GAthering in Sensor Information Systems (PEGASIS). PEGASIS

objectives are increasing the network lifetime and decreasing the bandwidth consumption. Unlike LEACH which forms multiple clusters, PEGASIS creates near optimal chains where nodes require only local communication with their 1-hop neighbors. PEGASIS constructs the chain in a greedy fashion choosing in each hop the neighbor closest to the sink. To choose the closest neighbor, each node utilizes the 1-hop neighbors distances estimated according to their signal strength. Once the chain construction finishes, each node using its closest neighbor transmits data which are aggregated in each hop till reaching the sink. When the sink receives data from all nodes, the round finishes and the chain-creation process is repeated. This process decreases the necessary power for transmitting data to the sink and distributes the energy consumption among all nodes.

Simulation results showed that PEGASIS outperforms LEACH about 100–300% for different network topologies and sizes. So, PEGASIS avoids the computing overhead of dynamic cluster formation and decreases the number of transmissions by optimizing the data aggregation. However PEGASIS introduces excessive packet delay from distant nodes on the chain. Also, nodes require energy neighborhood information to take routing decisions increasing the control traffic.

TEEN and APTEEN

Manjeshwar et al. proposed two hierarchical routing protocols called Threshold-sensitive Energy Efficient sensor Network (TEEN) [87] and Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network (APTEEN) [89]. TEEN and APTEEN are based on a hierarchical architecture where nodes closer to the sink form clusters. Both protocols were designed for time-critical applications where the network operates reactively to the changes of sensed parameters (i.e. temperature, humidity, etc.).

In TEEN, sensor nodes measure periodically parameters of the environment, but data transmission is done only for relevant information. After the clusters formation, each cluster head (CH) diffuses two thresholds for each measured

parameter to its members. First, the hard threshold represents a minimum interesting value of the sensed parameter. Second, the soft threshold indicates a small value change of the sensed parameter. The data transmission is only generated when the sensed value is bigger than the hard threshold and the value change is equal to or greater than the soft threshold.

The main advantages of TEEN are the suitability for time critical applications and the reduction in power consumption. As data transmission consumes more power than data sensing, the energy consumption in TEEN is less than proactive protocols such as LEACH. Moreover in the formation of each clusters, the CH distributes the thresholds, and the user can adjust both hard and soft thresholds in order to optimize the trade-off between power efficiency and data accuracy. Smaller value of the soft threshold produces more accurate information, but more power consumption.

On the other hand, APTEEN is an extension of TEEN and provides both proactive and reactive techniques for periodical collection and time-critical events, respectively. In APTEEN, each CH broadcasts the sensed parameters, their thresholds, the TDMA schedule and a count time to its members. The count time represents the maximum period in which a node can not send data, after that time data transmission is forced. Each CH also performs data aggregation in order to reduce consumption. APTEEN offers a high flexibility by supporting three different query types: historical for analyzing past data values, one-time for taking a snapshot view of the network and persistent for monitoring an event during a specific period. However the main drawback of APTEEN is the extra complexity required to develop the threshold functions and the count time.

Simulated results of TEEN and APTEEN have demonstrated that both algorithms outperform LEACH. Moreover the APTEEN performance is somewhere between TEEN and LEACH in terms of network lifetime and energy dissipation. In most of the tested scenarios, TEEN obtains the best results because it decreases the transmission overhead. The main disadvantages of both TEEN and APTEEN are the overhead and complexity related with the method of developing threshold-based functions, the way of dealing with parameter-based naming of

queries and the technique of forming clusters at multiple levels.

SOP

Subramanian et al. [88] presented a Self-Organizing Protocol (SOP) and a taxonomy of WSNs applications. According to such taxonomy, authors make an architecture supporting heterogeneous sensor nodes that can be mobile or stationary. In SOP nodes monitor the environment and transmit their data to a pre-configured set of nodes acting as routers. These routers forming the communication backbone are stationary and forward monitoring data to sink nodes. Each sensor node is connected to a router to participate in the network. Authors proposed a routing architecture which needs node identification. In this architecture each node is identifiable by means of the address of its router. The routing architecture follows a hierarchical model where clusters of sensor nodes are created.

To support fault tolerance SOP utilizes for packet broadcasting the Local Markov Loops (LML) algorithm which performs a random walk on spanning trees of a graph. The LML algorithm employs routers to keep all sensor nodes connected. In this algorithm nodes can be identified individually in the network. Thus SOP is suitable for applications where any node may be the communication destination. Due to the reduced number of routers, the proposed architecture achieves power efficiency.

As authors' results show, SOP requires a small cost for keeping routing tables and forming the hierarchical architecture. Owning to LML broadcasting trees, SOP decreases the energy consumption for broadcasting which is less than the energy required by the SPIN protocol [80]. In addition LML broadcasting trees enable fault tolerance (i.e. died nodes). However the main disadvantage is the proactive organization of LML which produces additional overhead. Another problem is associated to the hierarchy formation when there are many disconnected points in the network. In these cases the increment of reorganization overhead reduces significantly the communication efficiency.

2.2.3 Quality-of-Service routing

QoS-based routing protocols balance their performances between traffic quality and power efficiency. In these protocols, the path formation between sensor nodes and the sink is addressed as a network flow problem. Concretely, nodes delivering data to the sink must satisfy particular metrics such as delay, energy, bandwidth, etc. Various examples of QoS routing protocols are described below.

SAR

Sohrabi et al. [90] introduced one of the first QoS-based routing protocols for WSNs, called Sequential Assignment Routing (SAR). SAR is a table-driven multi-path solution with a local path restoration technique to prevent failures of single routes. To form multiple paths from source nodes to the sink, SAR constructs trees rooted at 1-hop neighbors of the sink. The path formation results in a multi-route tree composed by all nodes. Each node chooses one of these paths to transmit data according to three factors: the priority level of each message, energy resources of neighbors and QoS on each route. The aim of SAR is minimizing the average weighted factors and optimizing the network lifetime. The weighted factors are estimated as the product of a weight coefficient related with the priority level of the packet and the additive QoS factors.

SAR provides a route re-computation technique for supporting any topological changes (i.e. nodes failures). The sink triggers periodically the path re-computation, and nodes employ a localized handshake scheme to recover from network failures. The localized handshake recovery keeps the consistency of routing tables between downstream and upstream nodes on each path. Each node detecting any local failure performs automatically a localized path restoration.

As authors' results demonstrated, SAR provides tolerance to failures and easy recovery. Nevertheless, the algorithm needs an additional cost of routing table maintenance which is infeasible in dense networks.

SPEED

He et al. presented a sophisticated QoS routing protocol called SPEED [91] which supports soft real-time end-to-end guarantees in WSNs. In SPEED, nodes maintain neighborhood information and employ geographic routing to find the paths toward the sink. SPEED strives to ensure a pre-configured data rate in the network. To guarantee the data rate they divide the packet delay by the distance to the sink. The scheme prevents congestion in networks carrying high traffic.

SPEED contains a routing algorithm called Stateless Geographic Non-Deterministic forwarding (SNFG) which consists of four extra mechanisms. First, a beacon exchange mechanism provides neighbors' information such as positions. Second, nodes estimate the 1-hop neighbors delay by the elapsed time from data transmissions till the reception of ACK packets. Employing 1-hop delays, each node chooses the neighbor that satisfies the desired rate. Third, a failure detection mechanism checks the forwarding ratio of each neighbor considering a miss when the wished rate is not achieved. If the forwarding ratio is less than a random number between 0 and 1, the packet is dropped. Fourth, SPEED utilizes a local re-routing technique to prevent voids when a node can not find a next-hop neighbor. This technique avoids congestion by transmitting packets back to the sources, when they look for new paths.

Simulation results showed that SPEED outperforms two well-known ad-hoc routing protocols (dynamic source routing (DSR) [92] and ad-hoc on-demand vector routing (AODV) [93]) in terms of miss ratio and end-to-end delay. SPEED consumes less power because of its efficient design in aspects such as control packet overhead and uniform traffic distribution. The good balance is possible through the SNGF routing module which distributes packets in a reduced forwarding region. Nevertheless, the SNGF routing module does not consider any energy metric. Moreover DSR and AODV were not designed for WSNs, thus SPEED must be compared with efficient routing protocols to understand its realistic performance.

2.2.4 Geographic routing

In WSNs, the location of an event is a crucial information for the majority of applications. In most applications, nodes are manually deployed ensuring that routes exist to forward data toward the sinks [94]. To obtain positions, different localization systems exist such as Global Positioning System (GPS), infrastructure-based localization techniques, and ad-hoc localization algorithms [95, 96, 97, 98, 99, 100, 101]. Once location information is available, the operation of communication protocols is simplified improving the energy efficiency considerably.

Geographic routing employs location information of sensor nodes to take forwarding decisions. The objective of geographic routing is to reduce the distance towards the destination in each hop. So, nodes forward the packet through their closer 1-hop neighbor toward the destination. To take forwarding decisions, nodes need only local neighborhood information. Unlike previous algorithms, geographic routing requires neither routing tables nor flooding discovery activities. In fact, geographic routing prevents the extra cost of routing information maintenance which involves high energy consumption and frequent updates in mobile or dynamic networks. Therefore geographic routing is attractive for WSNs with frequent topology changes. Given that geographic routing is central for the work in this thesis, we analyze in detail these protocols in Section 2.3.

2.2.5 Routing Paradigms Comparison

This section provides a comparison of the four routing paradigms designed for WSNs: data-centric, hierarchical, QoS and geographic. For these four routing paradigms, we discuss their advantages and disadvantages summarized in Table 2.3.

Data-centric routing decreases significantly the communication overhead in networks where sensed data is of interesting to an unique sink. In these scenarios, data-centric algorithms construct power efficient paths between sensor nodes and

the sink. The path formation depends on attribute-value pairs often determined in the application layer. However, the limitation of pre-configured attribute-value pairs avoids sophisticated queries. These attribute-based protocols are inefficient in applications requiring connectivity to multiples sinks. Moreover, the path formation requires flooding mechanisms that are not scalable and robust in dense WSNs with dynamic topologies.

On the other hand, hierarchical routing is proposed as an efficient and scalable technique to route monitoring data of nodes groups to the sink. In these protocols each cluster-head node aggregates monitoring data of its region to decrease the traffic toward the sink. The additional overhead of clusters formation is traded for the energy saving in the data transmission phase in dense networks. Nevertheless this does not pay off in small networks with multiple sinks. Hierarchical protocols are inefficient in dynamic topologies (i.e. most of WSNs) where clusters updates are frequently producing excessive overhead.

Unlike data-centric and hierarchical paradigms proposed for power efficiency, QoS routing is focused on ensuring minimum performances in terms of delay and bandwidth. QoS techniques keeps multiple paths between sources nodes and sinks to ensure the network connectivity even when there are communication failures. The main disadvantage is the additional cost of keeping routing tables which comprises resources constraints, i.e. memory and energy, in dense WSNs.

Geographic routing exploits location information of sensor nodes to take forwarding decisions. Assuming the knowledge of location information, geographic routing provides the most efficient and scalable scheme to forward packets in comparison to previous paradigms. Unlike previous paradigms geographic routing needs neither maintaining routing tables nor flooding discovery activities. This paradigm decreases memory, traffic, computation and energy consumption because routing decisions are performed based on the positions of 1-hop neighbors and the destination. Geographic routing needs local neighborhood information supporting fast responses to dynamic topology changes. Taking localized routing decisions also offers a scalable solution in dense WSNs. Geographic routing using efficient destination discovery

Paradigms	Advantages	Disadvantages
Data-centric	High Energy-Efficiency	Low Connectivity Low Scalability Low Topology-Robustness
Hierarchical	High Energy-Efficiency High Scalability	Low Connectivity Low Topology-Robustness
QoS	High Topology-Robustness High Connectivity	Low Energy-Efficiency Low Scalability
Geographic	High Energy-Efficiency High Scalability High Topology-Robustness High Connectivity	

Table 2.3: Advantages and Disadvantages of Routing Paradigms.

mechanisms [102] enables full connectivity among all nodes.

2.3 Background on Geographic Routing

Based on previous research studies [103, 104, 105], we present an extensive overview of Geographic Routing (GR) and its two main forwarding strategies: greedy and face. In GR, forwarding decisions are based on the location of the destination and the positions of neighboring nodes in each hop. In greedy strategy, the current forwarder holding the packet chooses a closer neighbor as next hop reducing the distance toward the destination. Nevertheless, a greedy strategy does not ensure the packet delivery in sparse networks with void areas. A void area appears, when the current forwarder has no neighbors closer to the destination than itself and becomes a local maximum. To resolve a local maximum, face strategy enables that the packet advances around the perimeter of the void area. Face forwarding ensures the packet delivery in WSNs considering particular assumptions which we will explain in Subsection 2.3.3. Combinations of greedy

and face strategies provide the most efficient and effective GR solutions.

2.3.1 Network Model and Assumptions

As most of routing solutions, GR considers that WSNs follow an Unit Disk Graph model (UDG) [106]. Each link between two nodes u and v is represented as an edge $e = (u, v)$, if the distance $|uv|$ is lower than the radio range r. The UDG model assumes that all nodes have the same radio range, and all links are bidirectional. Moreover GR requires the following specific assumptions:

- Each node determines its geographic location using an existing positioning technique. As location information is key in many WSNs applications (i.e. environment monitoring), many positioning mechanisms have been proposed. To obtain relative positions, neighbors exchange packets and estimate the distance between them based on the received signal strength. [99, 100, 101]. Otherwise, nodes calculate their global coordinates using a low-power GPS (Global Positioning System) receiver [107].

- Each node knows the position of its 1-hop neighbors. Nodes can obtain this information by periodically broadcasting packets including their positions. To decrease the overhead of periodic transmissions, several on-demand mechanisms have been proposed to request neighbors' position only at the forwarding time [108, 109, 110].

- The source node knows the position of the destination. To map node identifiers to geographic locations, several location service mechanisms have been introduced for WSNs [102]. Location service mechanisms are classified according to their dissemination strategy into the following categories: flood, quorum, home-zone and movement. Flooding-based dissemination is the fastest way to disseminate location information in the entire network. To decrease the high transmission overhead of flooding, quorum-based and home-zone-based techniques store location information in one or more nodes whose positions are well-known. For mobile

networks, movement-based dissemination exchanges location information only between neighbors and exploits the mobility of nodes to distribute this information in the whole network. Moreover in some applications, nodes know the pre-configured destination position (i.e. a fixed sink node gathers all sensed data).

2.3.2 Greedy Forwarding

The first greedy forwarding algorithms were introduced in the 1980s for grid networks [111, 112, 113]. These strategies were designed to guarantee the packet delivery in uniformly dense networks without void areas. The main idea of these strategies is reducing the distance to the destination in each hop. The current forwarder chooses the next hop among neighbors closer to the destination than itself maximizing a local forwarding criterion. In the next subsections, we describe existing greedy strategies according to the criterion they use.

Greedy Routing Scheme (GRS)

Finn et al. proposed Greedy Routing Scheme(GRS) [112] based on the criterion of the advance which is equal to minimize the distance toward the destination in each hop. To maximize the advance toward the destination, the current forwarder chooses the neighbor located closest to the destination. The aim of GRS is providing the largest advance per hop in order to follow the shortest path. In GRS, backward forwarding is not allowable to avoid routing cycles. However, this scheme may produce that the packet follows a deviated path for the straight line from the source to the destination. Lateral deviations are common in networks with low density.

Most Forward within Radius (MFR)

Takagi and Kleinrock presented the Most Forward within Radius (MFR) routing strategy [111]. MFR introduces the notion of progress as the projection of the neighbor position on the line drawn from the current forwarder to the destination.

In MFR, the next hop selection provides the maximum progress in the destination direction. MFR seeks a double objective: minimizing the path length and optimizing the number of radio transmissions. The disadvantage of MFR is that a packet may move away from the destination even though there are nodes located on a more direct trajectory or physically closer.

Compass Routing (CR)

Kranakis et al. proposed Compass Routing (CR) [113] including the concept of angular distance. CR chooses the neighbor minimizing the angle distance based on the forwarder-destination line. In CR, the packet follows the most straight direction from the source to the destination. The main advantage of CR is that the packet is able to go around a void area in certain situations. However, this technique based on the direction is prone to routing cycles. We illustrate these greedy strategies in Fig 2.3.

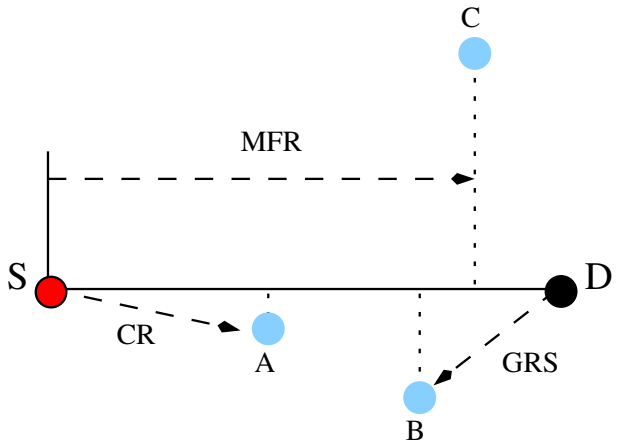


Figure 2.3: Greedy Routing Scheme(GRS), Compass Routing(CR) and Most Forward within Radius (MFR).

Other greedy strategies

Motivated by the reduction of traffic congestion, Bose presented two variants of CR and GRS, called Random Compass [114] and Random Progress Method [115]. Random Compass provides an arbitrary selection between the two closest-angle neighbors located on either side (clockwise and anticlockwise) of the forwarder-destination line. While Random Progress Method has an arbitrary selection between the two neighbors minimizing the destination distance. Both random strategies balance the communication overhead in the source-destination path among all intermediate nodes.

On the other hand, two energy-aware techniques were proposed to decrease the energy consumption by minimizing the adjustable transmission power. Nearest with Forwarding Progress (NFP) [116] and Nearest Closer (NC) [117] choose the neighbor which is closer to the destination using distance or progress criterion respectively and generates the minimum distance from the current forwarder.

Recently, Stojmenovic [118] presents a novel criterion for greedy forwarding based on the concept of cost over progress. The cost measure depends on the energy used, while progress measures the difference in distances to the destination. So, the current forwarder select the neighbor which minimizes the ratio of cost/progress.

In addition, Sanchez et al. propose Locally Optimal Source Routing (LOSR) [119] to reduce the overall energy consumption in greedy forwarding. The main idea of LOSR is to follow the shortest path in terms of energy between the current forwarder and the one originally selected as next-hop by the greedy routing scheme. To do that, the authors apply Dijkstra's algorithm [120] to the subgraph made of the neighbors of the current node. Then it uses a Source Routing Header (SRH) to annotate the message with the list of nodes that must be traversed. This makes the message follow the best energy-efficient path (eventually going through nodes which do not provide advance but reduce energy consumption) according to the local knowledge of the current node.

However, greedy strategies fail in void areas where there are no closer nodes decreasing the destination distance. To recover from a void area, a sophisticated

face strategy must be applied. Several face approaches have been proposed in order to guarantee the packet delivery.

2.3.3 Face Routing in Planar Graphs

Based on geographic positions, face routing is the most used strategy to guarantee the packet delivery. The key concept is the traversal of adjacent faces in a planar graph (see Fig 2.4). A planar graph represents the network topology removing all crossing edges. Using the planar graph, the packet is routed along the edges of the faces. To do that, face forwarding employs the right-hand rule (clockwise rotation) according to the source-destination line. The successful application of this strategy requires the previous planarization of the network graph using some distributed algorithms described below.

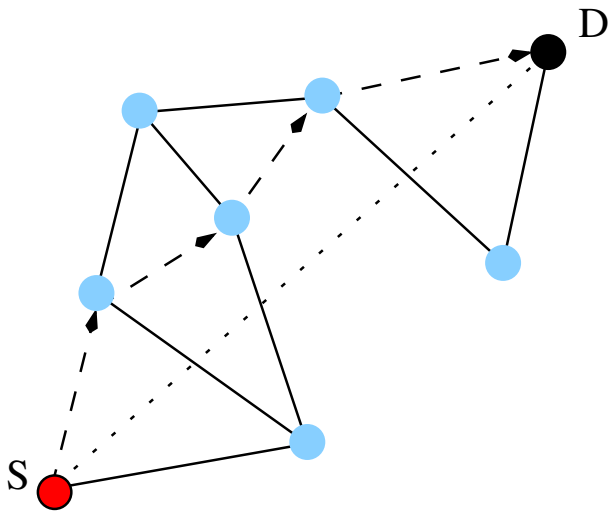


Figure 2.4: Face routing in a planar graph.

Planarization Algorithms

The planarization of the network graph is needed for ensuring the packet delivery, because crossing edges cause cycles in face routing (see Fig 2.5). Planar graph

algorithms assume that the network topology is an Unit Disk Graph (UDG). Based on UDG conditions, each node utilizing a distributed algorithm is able to construct a planar subgraph of its local topology by eliminating crossing links among its 1-hop neighbors.

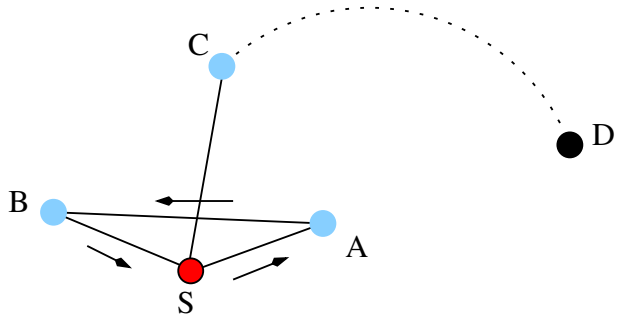


Figure 2.5: Crossing links causing a face routing failure.

For local planarization, several distributed algorithms were proposed depending on the geometric criterion employed such as Gabriel Graph (GG) [121] or Relative Neighborhood Graph (RNG) [122] (see Fig 2.6). In GG, a node u keeps the link to a neighbor v if no nodes exist within the circle whose distance is the segment $\overline{uv}$. In RNG, a node u keeps the link to neighbor v if the distance to v is lower than or equal to the distance from both u and v to every other neighbor, otherwise the link is removed. The GG or RNG criteria enable distributed algorithms to obtain planar subgraphs with only 1-hop neighborhood information. However, GG and RNG algorithms eliminate many crossing links increasing the path length in face routing. This makes it less efficient than greedy routing.

To improve the efficiency of face routing, the planar subgraph construction must approximate the original network as close as possible. This condition is defined as a spanning ratio whose objective is to minimize the path length between any two nodes in the subgraph. To address this issue, a planarization approach has been proposed based on the Delaunay triangulation [123]. Delaunay triangulation obtaining a constant spanning ratio ensures a constant overhead for any route

in the subgraph. The Delaunay triangulation of a group of nodes consists of all triangles whose circumcircle is empty. Unlike GG and RNG, the Delaunay triangulation cannot work only with local information to construct the planar subgraph. To achieve local constructions, several variations of this scheme have been developed in [124, 125, 126].

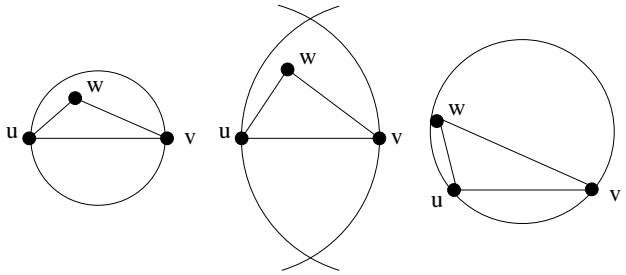


Figure 2.6: GG(left), RNG(middle) and Delaunay triangulation(right).

The main problem of planarization algorithms is the assumption of an idealized UDG model to represent realistic WSNs. WSNs are sensitive to inaccurate positions [101] and irregular radio ranges [9]. In real networks, a link between two nodes u and v may not exist even when their estimated distance $|uv|$ is lower than the theoretical radio range. Planarization algorithms fail to eliminate crossing links and to construct inaccurate planar graph, as described in [127, 128].

For realistic WSNs, Kim et al. proposed an alternative solution named Cross Link Detection Protocol (CLDP). CLDP eliminates crossing links and guarantees planar subgraphs, but increases the control overhead. To detect crossing links, nodes employ the right-hand rule to probe packets with their checked links. All links are probed several times till reaching a convergence situation where they are marked as routable or non-routable. However CLDP needs multihop exchanges to identify crossing links, thus the probing overhead grows as the network density increases. For this reason, the authors proposed an on-demand variant [129] that starts the probing process, when the face protocol must choose a link.

Face Routing

Kranakis et al. and Bose et al. introduced the earlier face routing strategies called Compass II [113] and Face-2 [130], respectively. Both strategies employ the right-hand rule to traverse a faces sequence of a planar graph till finding the destination. Compass Routing II constructs a planar subgraph based on the Delaunay triangulation criterion. Using this subgraph, the packet traverses fully each face until reaching the closest edge that intersects the source-destination line. In the intersection, the packet is forwarded to the end-node of this edge where the face is changed to continue the traversal. This process is repeated till the packet reaches the destination. Unlike Compass Routing II, Face-2 constructs a planar subgraph employing the Gabriel graph criterion. To reduce the traversal process, Face-2 changes to the next face at the first edge that intersects the source-destination line.

An issue of previous face algorithms is the choice of the optimal direction for a face traversal to avoid long detours. Using the right-hand rule, a face traversal always is performed in the right direction even when the left direction is a better choice. For instance, a right face traversal generates a long detour when the packet reaches a border of the network and goes away the destination. However, the optimal direction is impossible to determine by a local algorithm. To address this issue, Kuhn et al. proposed a face routing variant called Adaptive Face Routing (AFR) [131]. AFR employs an ellipse region around the source-destination line to bound the face traversal. When a packet finds the ellipse, the packet changes to the opposite direction. This scheme alleviates the issue of uninformed decisions preventing long detours in network borders.

2.3.4 Combined Greedy-Face Routing

Several combined approaches of greedy and face forwarding have been proposed to provide efficient and robust geographic solutions such as Greedy-Face-Greedy (GFG) [130] and Greedy Perimeter State Routing(GPSR) [132]. GFG combines greedy and face strategies based on Compass routing and Face-2, respectively. In

GFG, greedy strategy is applied until reaching the destination or a local maximum. When a local maximum is reached, the local maximum position and the first edge are stored in the packet which is forwarded in face mode. The face mode employs the Gabriel-graph criterion to obtain planar subgraphs. According to planar subgraph, Face-2 algorithm is utilized till finding a next hop closer to the destination than the local maximum where the greedy mode is resumed. On the other hand, GPSR provides greedy and face forwarding through Greedy Routing Scheme(GRS) and Face-2, respectively. GPSR constructs planar subgraps by the RNG algorithm. Like GFG, packets are first forwarded in greedy mode, when a local maximum is reached face strategy is used until greedy forwarding can be resumed.

Fig 2.7 shows an example of combining greedy and face routing. Where the source node S having a data packet addressed to the destination D. The data packet advances in greedy mode through the nodes $G1$ and $G2$ until reaching the local maximum M which has not closer neighbors toward the destination D. The local maximum M applying face mode routes the packet to the node $F4$ which has a closer neighbor $G5$ than M. So the packet can continue in greedy mode using the nodes $G5$ and $G6$ until reaching the destination D.

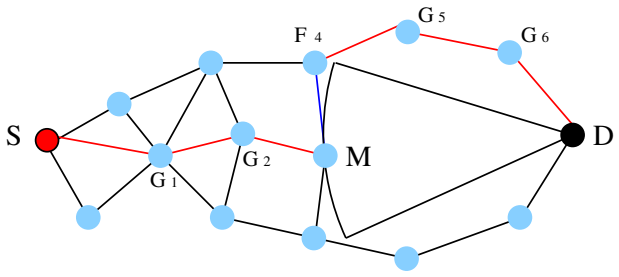


Figure 2.7: The data packet from the source node S is routed toward the destination node D using greedy and face mode.

Kuhn et al. [133] presented another geographic routing algorithm combining greedy and face strategies, called Greedy Other Adaptive Face Routing (GOAFR). GOAFR employs the GRS scheme in greedy mode, the current node chooses

the neighbor closest to the destination as the next hop. However, if the packet reaches a local maximum where there is no closer neighbors then GOAFR applies a variant of the AFR algorithm, called Other Face Routing (OFR). Unlike AFR, OFR utilizes a limited ellipse region whose size is increased progressively. The initial size of the ellipse is pre-established. If there is no path to the destination within the initial ellipse, its size is doubled and the local maximum restarts face routing again. The main aim is to find the closest node inside the limited area to the destination in order to obtain worst-case optimality for face routing. When OFR finished, it gives back the closest node to the local maximum. This face variant enhances significantly the performance of the GOAFR protocol in networks with low density. GOAFR provides a more efficient solution than AFR. However GOAFR still suffers from two main drawbacks. The full boundary may not be searched, and extra bandwidth is consumed by failed searches.

2.3.5 Beaconless Geographic Routing

Previous geographic protocols assume that nodes know their 1-hop neighborhood information by exchanging short control messages, called beacons. Each node broadcasts periodic beacons with its identifier and position. Beacons are not forwarded, thus only 1-hop neighbors can receive them. However, the beaconing mechanism reduces the efficiency of geographic protocols. For instance, periodic beacons can interfere with regular data transmission. Concretely in the nodes not taking part in any routing process, the bandwidth and power consumption represents a total waste of resources. To overcome such issues various beacon-less routing protocols have been proposed for WSNs.

Beacon-less routing algorithms employ a reactive scheme to discover 1-hop neighbors and select the next hop. In particular the current forwarder broadcasts the packet to discover its (unknown) neighbors. Neighbors receiving the packet participate as next-hop candidates. They wait a delay time according to one routing metric (i.e. the destination distance). So, the candidate closest to the destination has the shortest timeout. When the timer expires, the respective candidate transmits first the packet and becomes the next hop. There are two main

variants for this reactive scheme. In the first one, the next hop selection is based on candidates timers, thus the first transmission cancels the rest. In the second one, the current forwarder after receiving candidates transmissions chooses explicitly the next hop. The two most representative beacon-less algorithms are Beacon-Less Routing (BLR) [108] and Implicit Geographic Forwarding (IGF) [110], described in the chapter 3.

2.3.6 Advantages and Disadvantages

Geographic routing (GR) is considered as one of the most efficient and scalable solutions for WSNs [6, 7]. In GR, nodes require low energy and low computation capacities to take routing decisions. Routing decisions are based on the positions of the destination, the current forwarder and its 1-hop neighbors. Each node needs a minimum state to store only 1-hop neighbors positions. Thus, GR performance is not affected by the number of nodes and the neighbor density. For this localized design, geographic protocols provide a fully-distributed way to route packets. A node knowing its 1-hop neighborhood information is able to decide independently the next hop.

Combining greedy and face strategies, geographic algorithms are efficient and robust solutions for WSNs. Greedy strategy employs the most efficient paths from the source to the destination in uniformly-dense networks. While face strategy guarantees the packet delivery even in sparse networks with void areas. Moreover, these algorithms support networks failures and topological changes. The reason is that both beacon-based and beacon-less schemes permit detecting topological changes by proactive or reactive neighborhood discovery, respectively.

However, in realistic WSNs geographic routing protocols suffer from three relevant problems based on the assumptions of perfect wireless communications, accurate location information and safe deployment areas. First, geographic protocols are designed and simulated considering the perfect unit disk graph model to represent wireless sensor networks. However there are huge differences between a simulated link and a real one as demonstrated recent studies [9, 8]. Under realistic wireless networks, the performance of geographic protocols is

severely damaged by frequent communication problems such as interferences, collisions, packet losses, etc. Second, geographic algorithms neglect the common location inaccuracies produced by positioning systems [10]. Recent studies [10, 13] have shown that geographic algorithms are ineffective in networks with inaccurate positions. Both greedy and face strategies experiment a huge increment of packet losses as the location error increases [12]. In addition to the perfect conditions of wireless communications and location systems, most geographic routing algorithms assume that WSNs are deployed in secure areas. Nevertheless open wireless medium is prone to be attacked by malicious nodes which want to avoid the communication among nodes.

According to the problems of existing geographic protocols, this thesis is focused on providing reliable routing algorithms for realistic WSNs deployments. Chapter 3 analyzes in detail the effects of realistic wireless communications in the performance of geographic routing. There we propose a reliable geographic routing protocol, called BOSS which is designed to deal with losses, collisions and interferences common in wireless communications. Chapter 4 studies the main causes of greedy routing failures for location errors. Based on our study, we provide an effective geographic routing algorithm to guarantee the packet delivery in uniformly-dense networks even with high location errors. Finally, in Chapter 5 we analyze the behavior of routing attacks and develop a self-protected greedy protocol to defend from malicious nodes.

Chapter 3

Geographic Routing with Realistic Wireless Communications

As shown in the previous chapter, most geographic routing protocols are designed and evaluated considering the perfect Unit Disk Graph (UDG) model. Geographic protocols often assume perfect wireless links and fixed radio ranges. Recent experiments [8, 9] demonstrated that real wireless communications are prone to problems such as interferences and collisions. These communication problems severely degrade the performance of routing algorithms in terms of packet losses and retransmissions. Considering realistic radio propagation is essential to design efficient and reliable routing protocols in WSNs.

For realistic communications in WSNs, we propose the Beacon-less On Demand Strategy Scheme (BOSS). The design of BOSS provides several effective mechanisms to deal with error-prone wireless communications. Concretely, we have made a practical study to determine the impact of the packet size on its delivery probability. Our analysis conclude that the bigger the packet is, the lower the delivery probability is, when the distance between the sender and receiver is close to the radio range. For this reason, BOSS includes a beaconless neighborhood discovering scheme based on the idea of sending first the message including the data payload. This scheme discards neighbors with error-prone links which are likely to generate packet losses and retransmissions.

Several experiments have been performed to evaluate the performance of BOSS against the most important beaconless protocols. First, we simulate the protocols in networks with thousands of nodes to assess the scalability and efficiency. Second, we compare the protocols in a testbed scenario consisting of 35 wireless devices in order to validate the reliability with realistic communications.

In this chapter, the main goals are:

- Describing in detail the operations of existing beaconless geographic routing protocols and determining their most important issues.
- Performing an empirical analysis of realistic wireless radios to better understand their behaviors.
- Designing a reliable beaconless geographic protocol considering the realistic behavior of wireless radios.
- Incorporating sophisticated mechanisms to decrease the collisions and traffic during the neighborhood discovery and next-hop selection phases.
- Evaluating the proposal algorithm compared with two relevant beacon-less algorithms by extensive simulations and a real-testbed network.

3.1 Related Work: Beaconless Geographic Routing

As we commented in the previous chapter, beacon-less geographic algorithms employ a combination of the two main routing strategies: *greedy* and *face*. In greedy mode, the packet advances in the destination direction through neighbor closer in each hop. There are different schemes which differ on the metric to select the next hop (i.e. the one closest to the destination). The routing task may eventually reach a node which has no neighbors closer to the destination than itself, called *local maximum*. One of the face variants is used to traverse the void area until reaching a node closer to the destination than the local maximum where the greedy mode can continue.

In addition, beacon-less routing protocols are based on four different mechanisms:

Initial broadcast to all neighbors. The node currently holding the data packet initiates the process of selecting its next hop by broadcasting a message. Some protocols use special control messages for this purpose while others resort to broadcasting the data packet itself.

Definition of contention timers and forwarding area. Contention timers determine when neighbors answer the initial broadcast. In general, contention timers are defined so that nodes located closer to the destination answer first. After overhearing the first response, nodes cancel their timers to reduce contention. Finally, some protocols limit which neighbors answer the initial broadcast to those located in the so-called forwarding areas. The goal of the forwarding area is to guarantee that all the responses are received by all the candidates. This prevents forwarding inconsistencies across possible next hops.

Selection of next hop. In some protocols the next hop is selected by the sender based on the answers received by neighbors. In other cases, neighbors self-elect themselves in a distributed way and resend the data packet. Some protocols incorporate active acknowledgment using special control messages. However, passive acknowledgment is also used so that when the sender overhears the forwarding of the data packet it interprets that as an ACK from the next hop.

Face operation. When no neighbor provides advance towards the destination,

face routing needs to be applied. Traditional face routing requires the sender to know all its neighbor in order to construct a planar subgraph. In that situation beaconless protocols provide special conditions in their contention timers to make all neighbors report their positions. There are more efficient beaconless proposals that allow face routing by knowing only a relevant subset of neighbors (i.e. Kalosha et al. [134]). However, in this thesis we focus our contributions in the greedy operation. Hence, face details are not further analyzed because we will rely on any of the existing beaconless face routing solutions to deal with void areas.

Below, we explain the operation of the main beacon-less routing protocols in the literature with special emphasis on how they address each of the mechanisms highlighted before.

3.1.1 Implicit Geographic Forwarding (IGF)

Implicit Geographic Forwarding (IGF [110]) is one of the first beacon-less geographic routing protocols proposed in the literature. IGF combines MAC and network layers. The selection of the next hop is carried out at the MAC layer, and the actual delivery is done at the network layer. In IGF the node currently holding the packet broadcasts a Request to Send (RTS) frame and waits for the first Clear to Send (CTS) response. Each neighbor receiving the RTS frame evaluates its own suitability as next hop. The neighbor providing the largest advance towards the destination is preferred and should answer first. Finally, at the Network layer, the forwarding node transmits the data message and the selected neighbor confirms the reception by answering with an Acknowledgment message (ACK).

IGF includes two optimizations to reduce the number of responses and collisions. The first mechanism avoids simultaneous responses from neighbors based on timers. The second scheme cancels unnecessary responses when other neighbors' responses are overheard.

Upon receiving a RTS message, each neighbor sets a timer to wait before answering with a CTS message. The timer value depends on the reduction in distance towards the destination provided by the node plus a random component.

Thus, neighbors located closer to the destination answer first. Besides, neighbors overhearing an earlier CTS from another neighbor cancel their own timers.

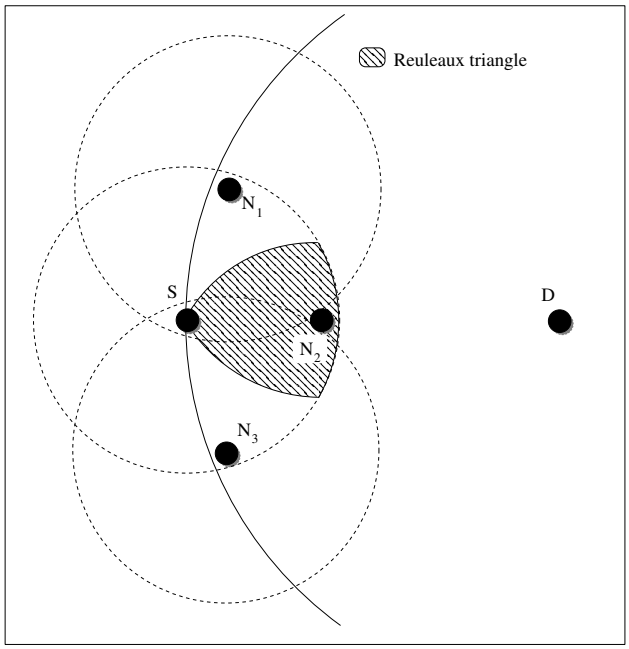


Figure 3.1: The forwarding area must be defined so that all nodes inside it can hear one another. The Reuleaux Triangle fulfills the condition of mutual possible reception for nodes located within it. Node S holding a message intended for D has three neighbors (N_1, N_2, N_3). Only N_2 is located inside the forwarding area defined by the Reuleaux Triangle. As it can be seen, transmissions from N_1 can not be overheard neither by N_2 nor by N_3.

IGF defines a forwarding area so that all nodes within that area are separated by a distance lower than the theoretical radio range. That is, in theory, all nodes inside it can hear one another (see Fig 3.1). Only those nodes located inside the forwarding area can take part in the selection process. This is defined that way to allow neighbors to overhear other neighbors' answers. However, in practice, radio propagation can make nodes within the forwarding area not to overhear some answers. Also, as a side effect, the use of a forwarding area may neglect

some neighbors providing a higher advance because of being outside that area.

3.1.2 Geographic Random Forwarding (GeRaF)

Geographic Random Forwarding (GeRaF [135]) is also a MAC/Network beaconless routing protocol. GeRaF's main contribution is a collision avoidance MAC scheme. In GeRaF next-hop candidates are those nodes whose positions are closer to the destination than the node currently holding the message. As Fig 3.2 depicts, that area is logically divided into N_p regions $A_1 \dots A_{N_p}$ such that all points in A_i are closer to the destination than all points in A_j for $j > i,\ i = 1 \dots N_p - 1$. The collision avoidance scheme assumes that nodes can have two radios. One is used for the traditional RTS/CTS handshake and the other is used just to transmit busy tones indicating that the first radio is being used to transmit control packets.

The contention scheme works as follows. The current forwarder transmits a RTS frame and starts waiting for responses during a period, called CTS slot. All nodes in the first region answer with a CTS frame and keep listening for a data packet from the current forwarder. When the forwarder successfully received the first CTS message it issues a data packet containing the payload and a header indicating the identifier of the neighbor selected as next hop. If the forwarder does not correctly receive a CTS frame within the CTS slot then, the data packet issued indicates a collision and all the nodes in the same region decide whether to send another CTS or not with probability 0.5. If no node answers during the CTS slot, the forwarder indicates in the message that nodes in the next region must answer because there are no neighbors in the first one. In the worst case this process is repeated N_p times, one for each region.

Besides, when a node has no neighbors providing advance towards the destination, GeRaF's authors suggest to use the Face-2 scheme [130]. As shown in Section 2.3.3 this scheme requires a local planarization of the neighbors. The planar subgraph construction needs the positions of all neighbors. To collect this information the current forwarder rebroadcasts the RTS message allowing all neighbors to answer even those not providing advance towards the destination.

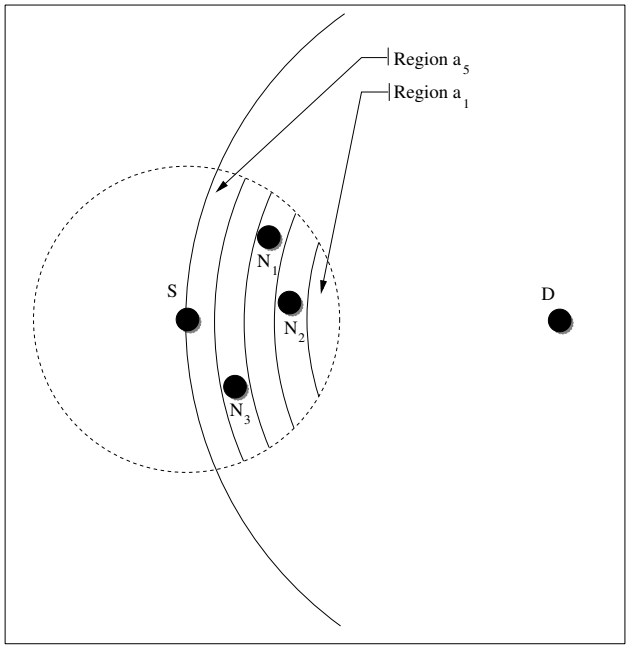


Figure 3.2: The area is divided in logical regions where neighbors are closer to the destination than the forwarding node.

3.1.3 Beacon-Less Routing (BLR)

Beacon-Less Routing (BLR [108]) relies on a distributed contention process (see Fig 3.3) as the only way of determining the next hop. BLR selects a next forwarder in a distributed manner among all its neighboring nodes without having information about their positions or about their existence. Data packets are broadcasted, and the protocol takes care that just one of the receiving nodes forwards the packet. This is accomplished by computing a Dynamic Forwarding Delay (DFD) at each neighbor depending on its position relative to the current forwarder and the destination.

Among all neighbors providing advance, the one in the best position forwards the data packet first. The remaining neighbors cancel their scheduled transmissions, when they overhear the data packet. To ensure that all nodes detect the forwarding, only nodes within a certain forwarding area take part in the contention to forward the packet. Furthermore, passive acknowledgments are used. That is, by detecting the transmission of the packet, the previous forwarder concludes that it was successfully received by its next hop.

Additionally, BLR includes a face strategy to deal with local maxima. The current forwarder broadcasts a short request, and all neighbors reply with a packet indicating their positions. If there is a neighbor located closer to the destination than the current forwarder, the neighbor is chosen as the next hop. Otherwise the actual forwarder extracts a planar subgraph (e.g. Gabriel Graph) for its neighborhood and forwards the packet according to the right-hand rule [6].

3.1.4 Contention-Based Forwarding (CBF)

In Contention-Based Forwarding (CBF [109]) there are two phases: contention process and suppression phase. In the contention process the current forwarder broadcasts the data packet and waits for its neighbors to determine themselves which one will be the next relay in a distributed contention process. During the contention process candidate neighbors compete for becoming the next relay by setting timers related to their actual positions. The neighbor providing the most

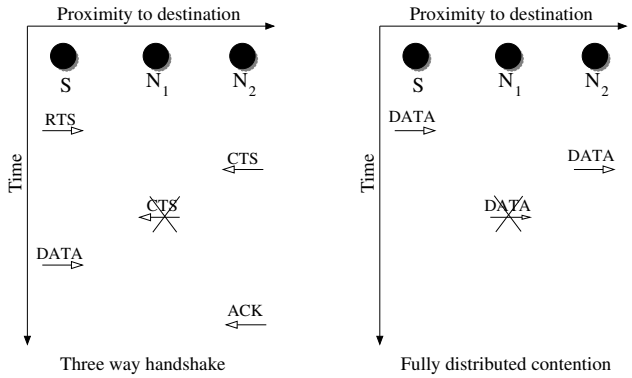


Figure 3.3: The contention timer is used to minimize the number of transmissions in the three way handshake and fully distributed contention. In that case, the first transmission of N_2 cancels the one of N_1.

advance towards the destination waits for the shortest time before forwarding the data packet. The remaining candidates cancel their timers when they hear the transmission from the winning neighbor.

The second phase is the suppression of redundant messages. The suppression phase is used to reduce the chance of accidentally selecting more than one node as the next hop as well as to reduce the overhead of duplicated packets. Three different suppression schemes are proposed. The basic scheme consists of canceling timers after hearing a transmission from another neighbor. The area based scheme defines a forwarding area as in IGF. Authors of CBF propose three different areas: Sector, Reuleaux triangle and Circle. Their results show that Reuleaux triangle is the forwarding area with better performance than Sector and Circle in terms of packet duplications and average advance in each hop. Finally, a third suppression mechanism is defined, called active suppression. The active suppression is equal to the RTS/CTS approach proposed in IGF that allows the forwarding node to determine which neighbor is selected as the next hop among the neighbors whose CTS frames were received. The active scheme selects explicitly an unique next hop preventing packet duplications. Multiple nodes may send a CTS control packet, but only one is selected because the forwarding node

acts as a central authority. Obviously, this requires the additional overhead of RTS/CTS control packets. Fig 3.3 shows the differences between the second and third schemes in terms of number of messages.

3.1.5 Motivation and Problem Statement

Most beacon-less routing protocols have been designed assuming perfect wireless communications. These protocols use the Unit Disk Graph (UDG) model [106] to represent wireless networks. This model assumes that nodes provide wireless interfaces with uniform transmission ranges and omni-directional antennas for receiving signals. In UDG, the network is represented as a graph $G = (V, E)$. Each vertex $v \in V$ indicates a node with two coordinates in a bidirectional plane ($v = (x_v, y_v)\ \forall\ v \in V$). Each $e \in E$ determines a direct communication link between two nodes (v_i, v_j) whose Euclidean distance is lower than the theoretical radio range R.

$$e = (v_i, v_j) \in E \Longleftrightarrow |\overline{v_i, v_j}| \leqslant R\ \forall\, v_i, v_j \in V \tag{3.1}$$

However, recent experiments [9, 8] have demonstrated that these assumptions are not satisfied in realistic networks, and the performance evaluations of routing protocols are very questionable. The reason is that the UDG graph is quite different to the behavior of error-prone wireless links including collisions, interferences and radio range variations. And these problems cause severe damages in the performance of routing protocols in terms of packet losses and retransmissions. Thus, the behavior of routing algorithms is unsatisfactory in realistic networks. Considering realistic communications we describe the most important issues influencing beaconless routing protocols.

- Unreliability. The collisions or interferences can cause packet losses during the forwarding process. In some protocols the lack of a retransmission mechanism makes them to achieve a low delivery ratio in realistic networks.
- Generation of duplicate messages. The usage of a forwarding area does not guarantee that all nodes within the area are able to overhead each other

	Beaconless Geographic Protocols				
Problem	IGF	$GeRaF$	CBF	BLR	
$Unreliability$	Low	$High$	$High$	$High$	
$Duplicates$	Low	Low	Low	$High$	
$Contention$	Med	Low	Med	Med	

Table 3.1: Common problems affecting beacon-less algorithms when considering real links

in error-prone wireless medium. Thus, two or more neighbors may not overhear each other even if their distance is lower than R. Then they consider themselves as the next forwarders and transmit duplicated packets.

- High Contention. It is necessary to minimize the probability of two neighbors answering at the same time. Thus, the design of reliable timer assignment functions is crucial.

Table 3.1 summarizes the level of influence of the previous issues in beaconless geographic protocols. Regarding *unreliability*, IGF is less influenced due to its active acknowledgment mechanism. The remaining protocols not using retransmission schemes are highly affected by this problem. According to the creation of *duplicates*, protocols with centralized selection decisions (IGF, GeRaF, CBF) are not affected by this problem. However, for BLR duplicates are a very serious problem because the distributed selection may fail in realistic wireless communications. We shall see clearly this effect in the experiments presented in the next section. Finally regarding *contention*, the protocols based on forwarding areas (IGF, CBF and BLR) have a moderate contention because that area limits which neighbors can answer. In the case of GeRaF based on forwarding subareas the contention is low because the division into subareas reduces the contention to those nodes within the same subarea, which is smaller than the whole forwarding area.

These arguments support the need of beaconless routing protocols being able to deal with the error-prone nature of wireless communications. These protocols

must contain efficient mechanisms to provide reliable communication minimizing contention and duplicated packets. Given these results of our analysis, in the next section we describe our proposed solution called BOSS.

3.2 BOSS: Beacon-less On-demand Strategy for Sensor Networks

In this chapter we proposed a reliable beacon-less protocol called BOSS (Beacon-less On-demand Strategy for Sensor networks). BOSS was designed to consider packet losses and duplicates which are common in realistic wireless networks. To avoid duplicates, BOSS employs a three way handshake to forward packets in a similar way to the RTS/CTS scheme used in IEEE 802.11 [70]. BOSS provides a retransmission mechanism based on both active and passive acknowledgment to improve reliability without increasing the control overhead. A contention timer function is included to decrease collisions and the number of answers during the neighborhood discovery, called Discrete Dynamic Forwarding Delay (DDFD). DDFD divides the neighborhood area into various sub-areas according to the advance towards the destination. Thus, the neighbors located in a high-advance sub-area answer before the remaining neighbors placed in low-advance sub-areas. Moreover, DDFD also prevents collisions among neighbors in the same sub-area.

The major contribution of BOSS is the use of a DATA message including the data payload to discover neighbors. The reason is that bigger messages are often more error-prone than short ones. For this reason, short RTS and CTS messages can traverse a link that a big DATA message cannot. By sending first the DATA message, BOSS performs the next-hop selection only among those neighbors that successfully received the data payload before. This design is justified by our results of a experimental analysis described in the next section that show the strong relationship between the size of the message and the error probability.

3.2.1 Analysis of Wireless Communications in Sensor Networks

BOSS is based on the assumption that the packet size has a direct relationship with the error probability. Concretely, bigger packets have less probability of being received than smaller ones. If that is the case, discovering the neighborhood using one small control packet may cause routing protocols to select a next-hop which is not able to receive the bigger data packet. The goal is to validate such assumption. Therefore, we run a set of real experiments in order to obtain the relation between Packet Size (PS) and the Packet Reception Ratio (PRR).

In the analysis, we employ a well-known sensor device called Tmote-sky[69]. The Tmote-sky integrates the following elements: a MSP430 microcontroller (with 10kB RAM memory and 48kB Flash memory), a CC2420 radio chip [37] (based in standard IEEE 802.15.4), a wireless antenna (that provides a radio range of up to 50 meters indoor and 125 meters outdoor according to the manufacture specification) and optionally sensors of humidity, temperature, and light to monitor the environment.

The experimental analysis was performed in an outdoor area of 100x100 meters. Two Tmote-sky nodes (sender and receiver) were placed at 0,5 meter above the ground and connected via USB to laptops. In each experiment the sender node transmits at maximum power (0 dBm) 50 sequences of 100 packets for each packet size. The experiments consider 8 different payload sizes (10, 25, 40, 55, 70, 85, 100 and 115 bytes) and varying the distance between the sender and receiver (from 5m to 120m). For each test the drawn results are the average of 50 sequences in order to achieve a sufficient small 95% confidence interval.

The receiver reports to its connected laptop the following measured parameters:

- PRR. The Packet Reception Ratio computed in the laptop is defined as the ratio between the number of packets received and the total number of packets sent.

- RSSI. The Radio Signal Strength Indicator is a 8-bits value given by the

CC2420 chip that indicates the received signal strength in dBm.

- LQI. The Link Quality Indicator can be viewed as the chip error ratio. It is calculated over 8-bits following the Start Frame Delimiter (SFD). The LQI values are usually between 110 and 50 and correspond to maximum and minimum quality frames respectively.
- PS. The Packet Size is the sum of the payload size and the sizes of headers in the MAC and link layers.

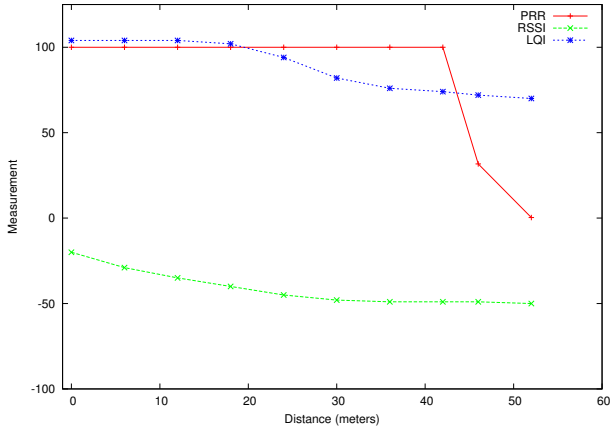


Figure 3.4: Relation of the measured parameters RSSI, LQI and PRR at varying the distance between sender and receiver.

Fig 3.4 presents the values of RSSI, LQI and PRR at varying the sender-receiver distance. As we can see, the RSSI and LQI decrease progressively when the distance increases. However, the PRR obtains almost 100% in distances shorter than 44 meters and decreases significantly in distances from 44 to 52 meters. For distances longer than 52 meters, the PRR is always zero.

As the PRR has greater variability in distances between 44 and 51 meters, Fig 3.5 shows this region with more resolution and varying the PS. Each curve represents the PRR obtained with each PS used. In this region, the PRR variations are not directly related to the distance between sender and receiver. Note that

this region is very important in greedy routing because nodes located closer to the radio range provide more advance toward the destination and are often selected as next-hops.

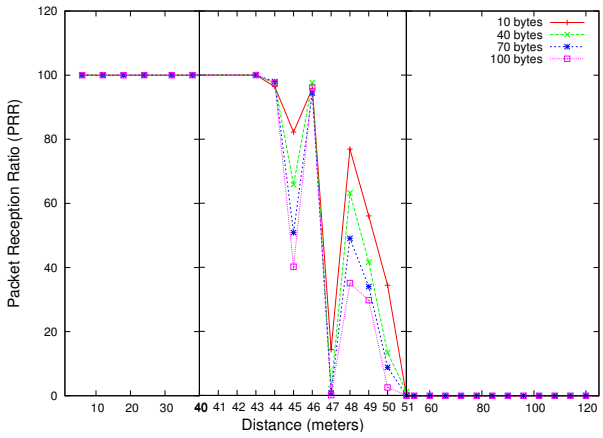


Figure 3.5: PRR values at varying the distance between 44 and 51 meters for different packet sizes.

To better understand the results, Fig 3.6 shows the relation between the PRR and PS in distances between 44 and 51 meters. Each curve represents the PRR obtained at each distance increasing the PS from 10 to 100 bytes. For each distance, the increment of PS decreases the PRR value. As we anticipated, the results demonstrate that bigger packets have less probability of being received than smaller ones.

In addition, the experiment shows that there is not a direct relation between the distance and PRR. As we can see, the results in longer distances (48 m and 46 m) can be better than in shorter distances (47 m and 45 m). This coincides with recent empirical studies [8, 136, 137] that show the irregularities of wireless communications in WSNs. The results also demonstrated that sensor nodes placed farther than 51m are not able to communicate directly. Although the maximum theoretical range is 125m, placing the sensors near the floor causes too much reflections. In some other tests done with sensors placed at 2m above the floor,

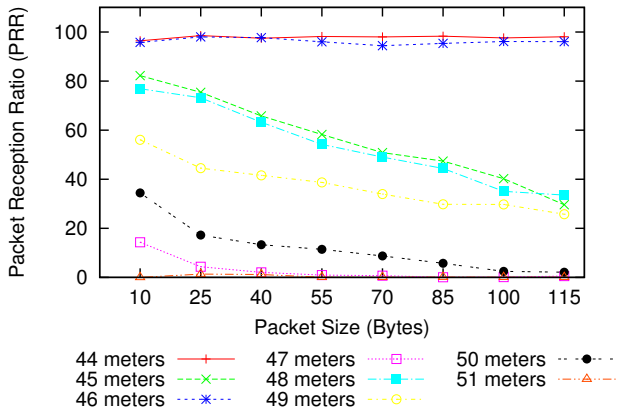


Figure 3.6: PRR values at varying the packet size in distances between 44 and 51 meters.

the range grows up to 150m.

Based on these conclusions, we design an enhanced forwarding handshake scheme to discover neighbors and select the next hop. Instead of using a control message, the forwarding node broadcasts a DATA message including the data payload to discover its neighbors. Since the bigger DATA message is more error-prone than shorter control messages. Thus, only neighbors receiving the DATA message participate in the selection phase. This scheme permits to select neighbors providing much advance and high reception probability. The forwarding operation of our BOSS protocol is presented below.

3.2.2 Data Forwarding of BOSS: Greedy and Face Mode

BOSS provides a three-way handshake scheme based on the following messages: *DATA*, *RESPONSE* and *SELECTION*. These messages include a bit in their headers to indicate the actual routing mode used (**G**reedy or **F**ace). This bit is called the Routing Mode bit (RM). The RM bit is set to **G** mode by default. Moreover given the forwarding node (i.e. the node currently holding the message), we define two relative areas around it: Positive Advance Area (PAA) and Negative

Advance Area (NAA). PAA comprises those neighbors located within the radio range which are closer to the destination than the forwarder, while the NAA contains the remaining neighbors located within the radio range but not providing advance to the destination. Now we describe the detailed operation of the protocol for greedy and face modes.

In greedy mode, the forwarding node broadcasts a *DATA* message and waits for responses during a predefined maximum time of T_{Max} seconds. The *DATA* message contains the original data packet, the positions of the forwarding node and final destination. Each neighbor receiving the *DATA* message stores it and determines the relative area where the neighbor is located (PAA or NAA). Instead of answering immediately, each neighbor starts a timer whose value depends on its position. When the timer fires, the neighbor broadcasts a *RESPONSE* message. The *RESPONSE* message contains the neighbor position and its identifier.

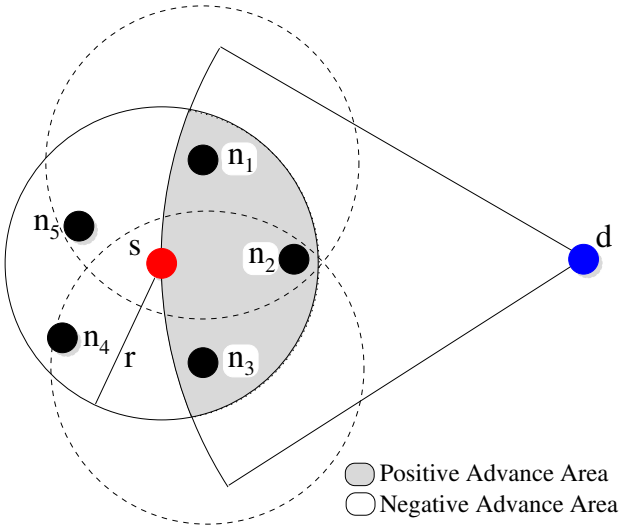


Figure 3.7: Node s currently holding the packet toward d and its neighbors. Nodes n_1, n_2 and n_3 are in the Positive Advance Area. Nodes n_4 and n_5 are in the Negative Advance Area. Nodes n_1 and n_3 cannot hear each other replies.

To reduce answers, PAA neighbors receiving a *RESPONSE* message from

another PAA neighbor cancel their timers and delete their stored *DATA* messages. *RESPONSE* messages from NAA neighbors do not cancel any timer. Note that it is possible that some NAA neighbors do not receive the *RESPONSE* message from a PAA neighbor located outside their radio coverages. Fig 3.7 shows an example of this situation where neighbors n_1 and n_3 cannot hear the *RESPONSE* messages from each other.

The forwarding node receiving a *RESPONSE* from a PAA neighbor stops its timer and broadcasts a *SELECTION* message. The *SELECTION* message contains the neighbor identifier of the first *RESPONSE* received. More than one *RESPONSE* message from PAA neighbors might arrive to the forwarding node but only the first one is used. Each neighbor receiving the *SELECTION* message cancels immediately its timer and deletes the stored message except for the selected one whose identifier is included in the message. The selected neighbor becomes the new forwarding node and starts again the handshake scheme by broadcasting a new *DATA* message. The *SELECTION* message also allows nodes to cancel their timers if they did not overheard the *RESPONSE* from the selected PAA node.

Fig 3.8 shows an example of the BOSS forwarding handshake. The forwarding node s broadcasts the *DATA* message which is received by neighbors n_1 and n_2. But neighbor n_3 cannot hear the *DATA* message due to its weak wireless link to the s. Then the closest neighbor n_2, whose timer expires first, broadcasts the *RESPONSE* message which cancels the transmission of n_1. The relay s indicates n_2 as the next hop using a *SELECTION* message.

Some nodes may have no neighbors providing advance toward the destination. In that case a so-called void area is found, and the routing process cannot continue in greedy mode. In geographic routing protocols, there are different face strategies to surround these void areas, but they need the position of 1-hop neighbors in order to build locally a planar subgraph. As we have already commented, in BOSS the *RESPONSE* messages from NAA neighbors do not cancel any timer. Thus when the forwarding node has no PAA neighbors, then it becomes a local maximum. In that case, the forwarding node waits during a predefined maximum

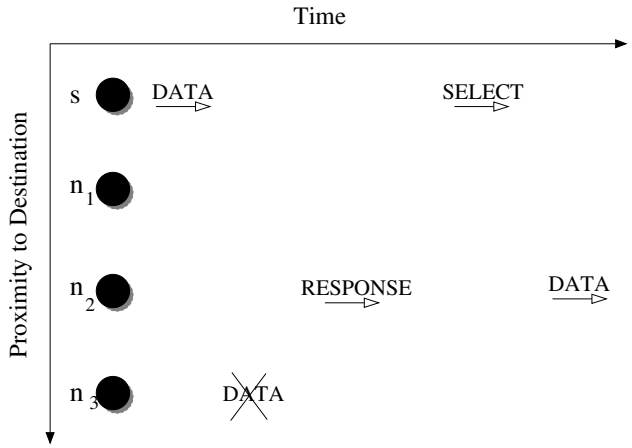


Figure 3.8: The node s holding a data packet addressed to d has three greedy neighbors n_1, n_2 and n_3 ordered by its proximity to the destination. The wireless link between s and n_3 is weak due to its long distance near to the radio range.

time of T_{Max} in order to store all the *RESPONSE* messages transmitted from NAA neighbors. When the T_{Max} timer expires, the forwarding node builds the planar Gabriel Graph [121] using all NAA neighbors' positions and selects the next hop employing the Face-2 mechanism, described in Section 2.3.3.

In face routing, the *SELECTION* message includes some extra information. Concretely the identifier and position of the forwarding node, the identifier of the next hop selected and the current face information defined by Face-2. The face information consists of the position of the local maximum where face routing started (L_p), the first edge (E_0) traversed on the current face and the point (L_f). L_f indicates the cross point between the $\overline{L_pD}$ line and the current face, being D the position of the destination node. Additionally, the RM bit is changed to **F** representing face routing.

When data packets are being routed in face routing, the behavior of neighbors is slightly different. First, the forwarding node includes in the *DATA* message the L_p point. A neighbor receiving the *DATA* message checks if its position is closer to the destination than the L_p point. If that is the case, the routing process must

resume to greedy mode. Thus, the RM bit of the *RESPONSE* message is set to **G**. Only in those cases, the *RESPONSE* message will be sent, but after the timer expires as in greedy routing. The forwarding node may stop its timer if it receives a *RESPONSE* message including a $RM \equiv$**G**. In that case, the forwarder changes to greedy mode and sends the appropriate *SELECTION* message with a $RM \equiv$**G**. If there is no neighbor closer to the destination than L_p then the forwarding node will wait up to T_{Max} seconds. The forwarder applies the Face-2 mechanism for determining the next hop and selects it by broadcasting the corresponding *SELECTION* message including also a $RM \equiv$**F**.

3.2.3 Detailed Operation in Realistic Wireless Networks

In realistic wireless network, collisions, interferences and packets losses damage severely the routing protocols. Thus, we consider the error-prone wireless communications in the design of BOSS protocol. The BOSS design includes various simple mechanisms to improve reliability and contention.

Data Discovering, Passive Acknowledgement and Retransmission Scheme

The design of BOSS is based on the direct relation between the packet size and packet reception ratio (PRR), shown in our wireless communication analysis 3.2.1. The main idea behind BOSS is to include the data payload in the first *DATA* message to discover neighborhood. So, the current forwarder broadcasts a *DATA* message and waits for the *RESPONSE* message of its neighbors. The forwarder uses a short *SELECTION* message to indicate the neighbor becoming the next hop. In BOSS only neighbors receiving the *DATA* message participate in the forwarding process. Unlike BOSS, the traditional RTS/CTS mechanism employs a small control message for discovering neighbors and may select a neighbor whose reception probability of a bigger data packet may be very low. Therefore, the neighborhood discovery of BOSS prevents message losses and retransmissions during the selection phase.

In BOSS, the *DATA* message is stored temporally by neighbors during the

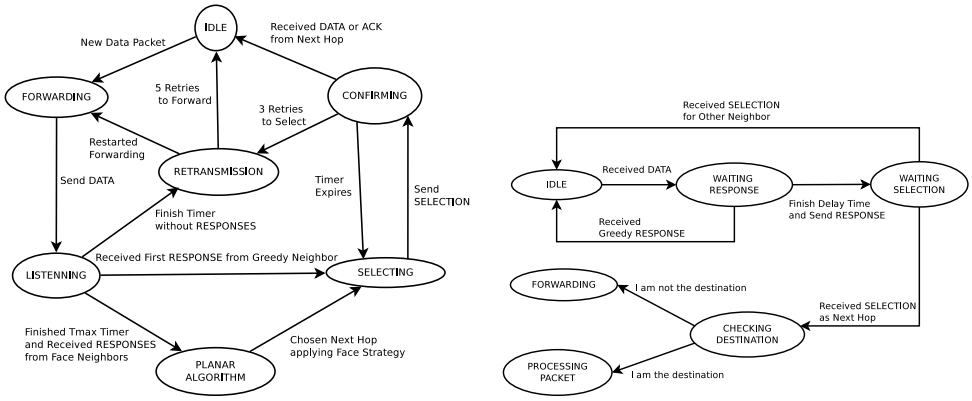
forwarding process. To implement that, a neighbor receiving a *DATA* message stores it and waits some time to answer. Obviously, the neighbor receiving a *RESPONSE* message from other PAA neighbor deletes the *DATA* message. The same occurs after receiving a *SELECTION* message from the forwarder to another selected neighbor. But, as we are dealing with error-prone wireless networks, both messages might be lost. In that case, the neighbor sends its own *RESPONSE* message when its waiting timer expires. The forwarding node receiving that late response must ignore it, but the neighbor is waiting for a *SELECTION* message. So, the neighbor will never receive the *SELECTION* message and will delete the *DATA* message after a maximum time.

Moreover, as a *SELECTION* message may be lost, the forwarder needs the confirmation of its reception. To do that, we use two different techniques: a passive acknowledgement (PACK) and an active one (ACK). The use of *ACK* introduces a new message in the forwarding process incrementing the protocol overhead in a 33%. Thus, BOSS employs the *DATA* message of the next forwarder as *PACK* to confirm the reception of the previous *SELECTION* message. The *ACK* is also needed when the *SELECTION* message arrives to the destination. When a forwarding node does not receive a *PACK* or an *ACK*, it resends the *SELECTION* message up to a maximum of 3 times. That means that, neighbors selected as next forwarders must keep their *DATA* messages during at least the 3 possible rounds of re-selections.

Finally, when the third re-selection fails the whole handshake scheme is repeated. The forwarder re-transmits the *DATA* message and the neighbors start again the contention timers. The handshake repetition can be tried up to 5 times. After that, the packet is dropped. However, the experimental results of this chapter show that in BOSS the *DATA* retransmissions are rarely used. The reason is that the neighborhood discovery of BOSS limits participating neighbors only to receive the big *DATA* message. The *RESPONSE* and *SELECTION* messages are significantly smaller than the *DATA* one, thus their reception probability is higher. For this reason, the probability of needing retransmissions is very low.

In the BOSS handshake scheme, a sensor node can play two possible roles as

forwarder or neighbor. The behavior of these two roles are shown in Fig 3.9 using two state diagrams.



(a) The behavior of a forwarder (b) The behavior of a neighbor

Figure 3.9: State diagrams of the BOSS protocol.

Discrete Dynamic Forwarding Delay (DDFD)

In the handshake scheme, all the neighbors wait for a period of time before answering the forwarding node. The waiting time is related to the neighbor position. This behavior has three important goals: avoiding collisions, reducing responses and determining the forwarding strategy. Note that if all the neighbors answer immediately, the probability of collisions increases exponentially because the *DATA* message arrives almost at the same time to all of them. On the other hand, in BOSS the forwarding node selects as next hop the neighbor which replies first. Therefore, the forwarding strategy is clearly controlled by the assignment of waiting time. Additionally, the first answer cancels the remaining responses in order to reduce the bandwidth consumption. The key is to design a function for determining the waiting times to guarantee that the most promising neighbors answer first.

As some other beaconless protocols, in BOSS a neighbor determines its waiting time based on the concept of the advance in order to measure its goodness as next forwarder. In our case, we define the advance as follows:

$$A(j,d,i) = dist(i,d) - dist(j,d) \tag{3.2}$$

where j, i and d are the neighboring, forwarding and the destination nodes respectively, and $dist(a,b)$ represents the Euclidean distance between the positions of the nodes a and b. Obviously, the maximum advance possible is equal to the radio range R, and the minimum one is $-R$.

Our Discrete Dynamic Forwarding Delay (DDFD) function assigns smaller delay times to the neighbors providing the maximum advance toward the destination. To do that, instead of using directly the advance value, we divide the neighborhood into sets of neighbors providing a similar advance (see Fig 3.10). Concretely, we define the Number of Sub Areas (NSA) in which the whole coverage area is uniformly divided. Then, considering that the maximum difference in advance between two neighbors is $2 * R$, each neighbor determines in which Common Sub Area (CSA) it is placed using the following equation:

$$CSA = \left\lfloor NSA \times \frac{R - A(j,d,i)}{2 * R} \right\rfloor \tag{3.3}$$

Here, the value of CSA falls between 0 and $NSA - 1$ corresponding 0 to the area placed closest to the destination and $NSA-1$ to the farthest one. Given the CSA, each neighbor computes its waiting time according to the next equation:

$$T = \left(CSA \times \frac{T_{Max}}{NSA}\right) + random\left(\frac{T_{Max}}{NSA}\right) \tag{3.4}$$

Where, T_{Max} is a constant representing the maximum delay time that a forwarding node will wait for answers from its neighbors, and $random(x)$ is a function obtaining a random value between 0 and x. By its construction, the function assigns half the total T_{Max} delay to the PAA neighbors and half to the NAA neighbors. That allows the forwarding node to determine whether there are PAA neighbors or not because a PAA neighbor will always answer before $\frac{T_{Max}}{2}$ seconds. Additionally, the neighbors in the same CSA can wait different amount of times thanks to the random function. Neighbors from consecutive $CSAs$ will never wait the same amount of time because the base time is determined by the CSA index.

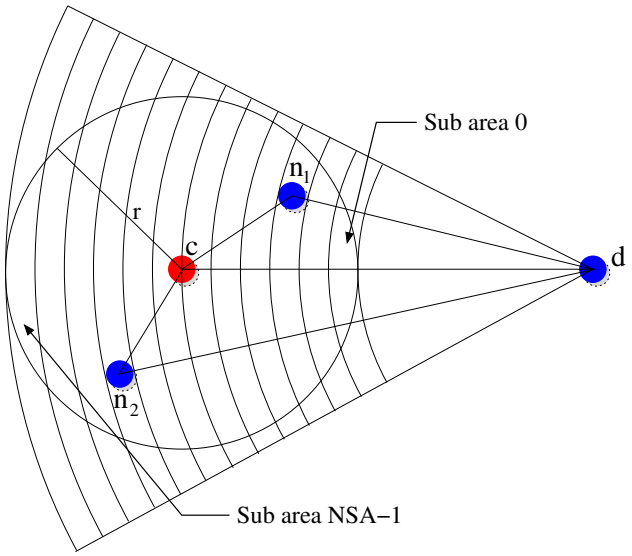


Figure 3.10: Division in areas for the DDFD.

Unlike other beaconless proposals, our DDFD function combines an uniformly distributed value depending on the advance with a random value generated. So that the total delay does not mixes responses from neighbors in different sub-areas. Thus, the function reduces the number of responses and the probability of generating simultaneous responses from neighbors which are in the same sub-area. Unlike DDFD, other proposed functions calculate the waiting time only depending on the advance value. In those cases, the forwarding node could have several neighbors providing a similar advance, and therefore they reply simultaneously causing collisions.

Finally, our DDFD function combining both routing modes (greedy and face) reduces the transmission overhead and the hop-to-hop delay. When greedy routing fails, the current forwarder does not need to re-start a new handshake process (DATA, RESPONSES). In those cases, no PAA neighbors will answer during the first $\frac{T_{Max}}{2}$ seconds, but after that, all NAA neighbors will reply during the second half of T_{Max}. The equivalent situation occurs when the data packet is routed in face mode. If there is a neighbor closer to the destination than the position of

the local maximum where face routing started, then that neighbor answers before any of the NAA neighbors. Then the *SELECTION* message transmitted by the forwarding node will cancel all the responses from those NAA neighbors. Unlike existing protocols, in face mode our DDFD function reduces the control overhead avoiding another discovering message and its corresponding delay.

3.3 Simulation and Real-Testbed Results

This section provides a comparison between BOSS and two relevant beacon-less algorithms: Beacon-Less Routing (BLR) [108] and Implicit Geographic Forwarding (IGF) [110]. BLR is the best-know proposal of fully distributed forwarding, while IGF is the best-performance solution of three way handshake scheme. To evaluate these protocols, we use two types of experiments: simulated and real-testbed networks. Simulation is used to validate the scalability and efficiency of these protocols in networks with a large number of nodes. Testbed experiments are performed in order to assess the reliability and robustness of the protocols in real deployments.

As existing geographic routing protocols, we assume that nodes know their own accurate positions by means of any positioning system [98, 138], a source node can determine the position of a destination by a location service [5, 139, 140].

To develop the protocols, we use the TinyOS [16] operation system which is one of the most used systems to implement WSNs applications and protocols. The protocols are implemented in the NesC programming language, a component-oriented variant of C language.

3.3.1 Performance Metrics

For the evaluation of the protocols, we consider the following performance metrics:

- Duplicated Packets. This metric accounts for the number of data packets

received by the destination for each one sent by the source.

- Total Face Transmissions. This metric indicates the total number of messages transmitted during the forwarding process in face mode. It determines the overhead generated by the recovery strategies of the beaconless protocols.

- Total Transmissions. This metric accounts for the total number of packets transmitted during the routing process of a data packet from the source to the destination. It includes also the messages not received and the transmissions made for duplicated packets due to communication errors.

- Packets per Hop. This metric calculates the mean number of messages transmitted during the routing process in each hop. It includes also the duplicated messages propagated in the path from the source to the destination.

- Packet Delivery Ratio. This metric shows the reliability of the protocols. It determines the percentage of packets that reach the destination node. This is an important performance metric in scenarios with realistic wireless conditions.

- End-to-end Delay. This metric accounts for the total time required from the source starts the forwarding process until the packet reaches the destination.

- Hop Count. This metric estimates the mean number of point-to-point links in a path. The number of hops is the average number of intermediate nodes between the source and the destination.

3.3.2 Simulation Experiments

To evaluate the protocols, we run them on the TinyOS simulator, called TOSSIM [17]. TOSSIM emulates communications of wireless sensor nodes. In addition we model the network with a realistic MAC layer including collisions and

interferences. That is, a message sent out by a node might not be received by other nodes in its radio range. To do that, we use the results of our empirical experiments commented in Section 3.2.1. Based on this results we build a function to compute the reception probability of a packet depending on its size and the sender-receiver distance.

For the configuration of BOSS, we use the following values: $T_{Max} = 600ms$ and $NSA = 10$. In IGF there are two different timers set to $300ms$ for greedy and face modes, respectively. The equivalent occurs in the case of BLR, it requires two timers of $300ms$. To make a fair comparison, the three protocols are configured to behave in the same way when packets losses occur. The *SELECTION/ACK* process can be repeated up to 3 times and the maximum number of retransmissions is set to 5. BLR can only do that in face mode because in greedy mode there is a fully distributed forwarding.

The simulated scenario is a 500x500m^2 area in which a varying number of nodes (from 150 to 700 nodes) are deployed. This results in networks with different densities (mean number of neighbors/node). We have considered 8 different mean densities to represent a wide spectrum of networks such as sparse and dense. On the other hand, the source and destination nodes are always placed in (0,0) and (500,500) coordinates, respectively. Thus, using a radio range of $R = 50$, the theoretical minimum number of hops is $\frac{500\sqrt{2}}{50} \simeq 14$. Although, the assumption of fixed radio range is not realistic, that is useful to determine the deviation from the best path of each tested protocol. For each scenario the results are the average over a total number of 200 simulations in order to achieve a sufficiently small 95% confidence interval.

3.3.3 Analysis of Simulation Results

First, we study the overhead of duplicated packets in realistic conditions in wireless communications. Fig 3.11 shows the mean number of duplicated packets arriving to the destination for each one sent by the source. As we can see, in BLR the number of duplicates increases exponentially with the network density. Because the forwarding area of BLR is not able to ensure that all neighbors inside

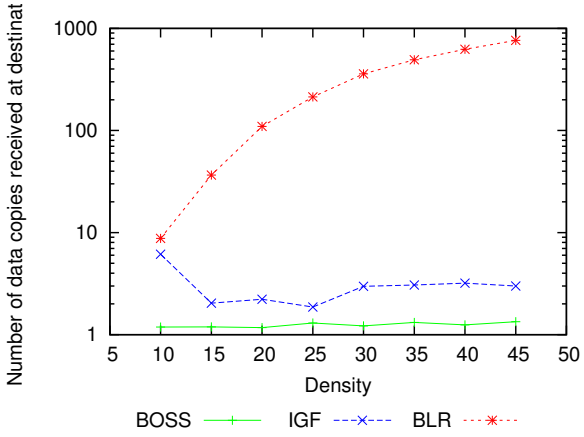


Figure 3.11: Duplicated Packets

this area overhear each other due to collisions or interferences. IGF and BOSS employ a three-way handshake scheme where the forwarder indicates explicitly the selected neighbor as next hop. Thus, IGF only has a mean of 3 duplicated packets, while BOSS does not produce almost any duplicate. IGF generates more duplicates than BOSS because of the robustness of our neighborhood discovery scheme. Unlike BOSS, IGF does not consider the quality of the links during the discovery process, neighbors with lossy links are likely to be chosen. Thus, the *ACK* messages can be lost making that the forwarder selects a different neighbor as next hop although the first selected neighbor is already forwarding the message. However BOSS allows that only neighbors receiving the *DATA* message take part in the *SELECTION/ACK* process to avoid packets losses.

Here, we analyze the efficiency of beaconless protocols in face mode. Face routing is applied when the forwarder has no neighbors closer to the destination than itself. Fig 3.12 shows the number of transmissions made during face routing. The scenarios with lower density force the three protocols to utilize face mode. In those sparse networks, the three protocols require a higher number of messages. In sparse scenarios, the neighborhood discovery scheme of BOSS consists of sending the data packet in the first place to allow discarding the neighbors with

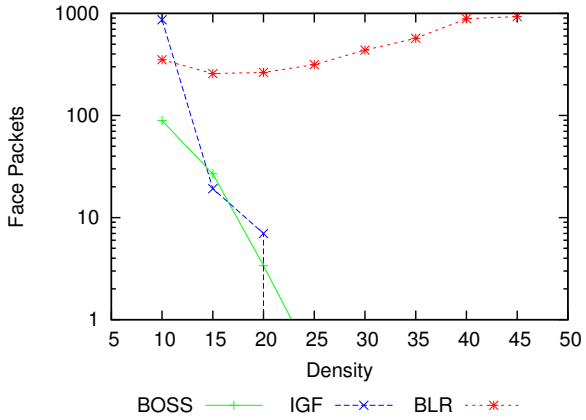


Figure 3.12: Total Face Transmissions

lossy links. In IGF, the probability of choosing a lossy link is higher as the number of packets routed in face mode confirms. But unlike IGF and BOSS, BLR continues using face mode when the density increases. The reason is that the increment of the density increases the probability of simultaneous transmissions from neighbors participating in the forwarding process. Those collisions prevent that the forwarder receives the transmissions from its neighbors and the remaining neighbors cancel their timers. Thus, the forwarder changes to face mode although, in parallel, new branches of duplicated packets are created.

Now we measure the transmission overhead required by the protocols to route data packets. Fig 3.13 shows the mean number of messages transmitted by each protocol at increasing densities. The global performance of BLR is very bad in comparison with the other two algorithms. The results prove the inefficacy of BLR operation where neighbors compete distributively to forward the packet, and duplicates are very frequent. IGF and BOSS transmit more messages in the scenarios with lower densities due to the extra overhead of the face mode. Both algorithms reach their normal behavior when the density is above 20, and the routing process is performed mostly in greedy mode. BOSS transmits less messages than IGF in all the networks tested. In most of them, BOSS only needs

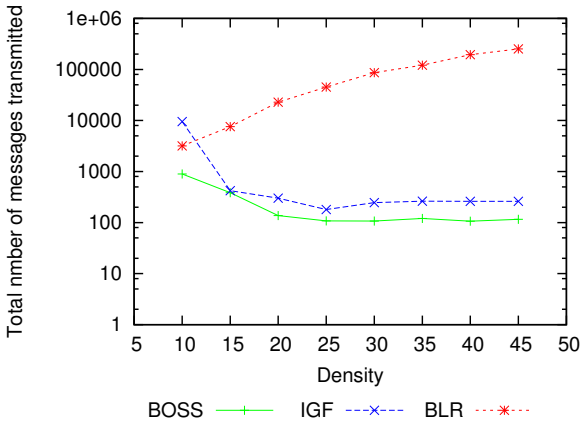


Figure 3.13: Total Transmissions

100 messages that is less than half the transmissions required by IGF. The reason is that the neighborhood discovery technique of BOSS is more reliable than the typical RTS/CTS scheme. Unlike IGF, the discovery technique of BOSS discards neighbors with weak links in order to prevent retransmissions in the selection phase.

Here, we examine more closely the two protocols with better performance: IGF and BOSS. To work in realistic networks, authors of IGF consider the problems caused by lossy links, and they define an active acknowledgement method to confirm the successful data delivery per hop. The next forwarder sends an *ACK* message to confirm the reception of the last selection transmitted from the previous forwarder. Therefore, the mean number of messages per hop is at least 4. BOSS employs the *DATA* message transmission as a passive confirmation to the previous forwarder. The passive acknowledgement reduces one message in the forwarding process per hop. In BOSS, the *ACK* message is only transmitted when the previous forwarder does not receive the *DATA* message and retries the selection phase. That situation rarely happen due to the effective discovering way to avoid error-prone neighbors. Fig 3.14 shows the mean number of packets per hop. We can see that the mean number of packets per hop of IGF is 7 and 5

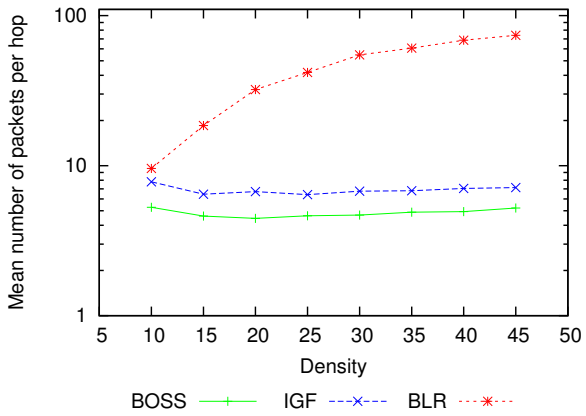


Figure 3.14: Packets per Hop

for BOSS. In theory IGF has only one more message per hop than BOSS, but in practice the number of responses and retries is being higher in IGF than in BOSS. That means, the delay function included in IGF has a worst performance than our DDFD function.

To analyze the reliability of the protocols, Fig 3.15 shows their packet delivery ratio achieved at varying mean densities. The three algorithms have a high delivery ratio because using 10 forwarding retries is enough in most of the situations. Nevertheless, BLR has the higher ratio due to the unacceptable number of duplicated packets. On the other hand, IGF does not generate excessive duplicates, but its delivery ratio is lower than the one of BOSS. The reason is that IGF chooses lossy links during the discovery of neighbors. Finally, BOSS almost has perfect delivery for the networks with more than 20 neighbors per node. That is, when the packets are routed mostly in greedy mode. At the same time, as we have already seen, BOSS transmits half of messages than IGF and thousands of times less messages than BLR.

The performance of the protocols in terms of the end-to-end delay is shown in Fig 3.16. The end-to-end time is calculated according to the first data packet arriving to the destination. The results demonstrate that the density is

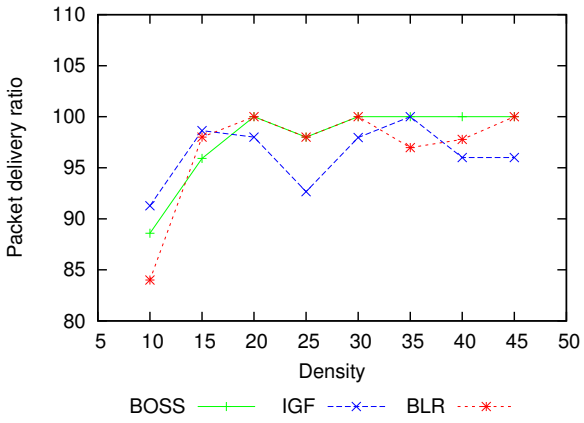


Figure 3.15: Packet Delivery Ratio

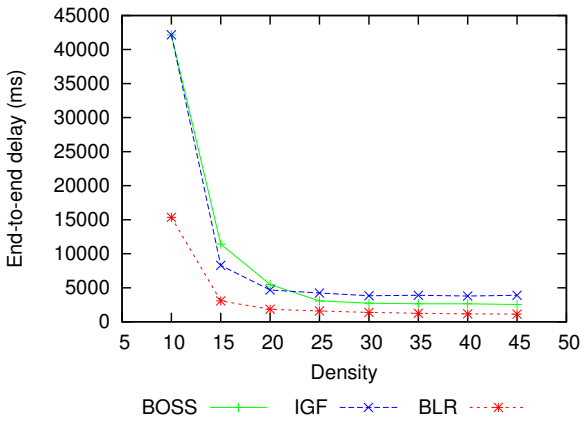


Figure 3.16: End-to-end Delay

strongly correlated with the end-to-end forwarding time. The decrement of the network density increases the end-to-end delay. Obviously, this is due to the effect of face routing. The figure shows that BLR achieves the shorter end-to-end delay than IGF and BOSS. Because, in BLR neighbors compete distributively and forward directly the data packet when their timers expire. However as we abovementioned, this distributed scheme generates a huge amount of duplicates. On the other hand, BOSS outperforms IGF for two reasons. The greedy-face combination of the DDFD delay function avoids restarting the handshake process (DATA/RESPONSES) when greedy routing fails. And the neighborhood discovery scheme discarding unreachable neighbors prevents retransmissions in the selection phase.

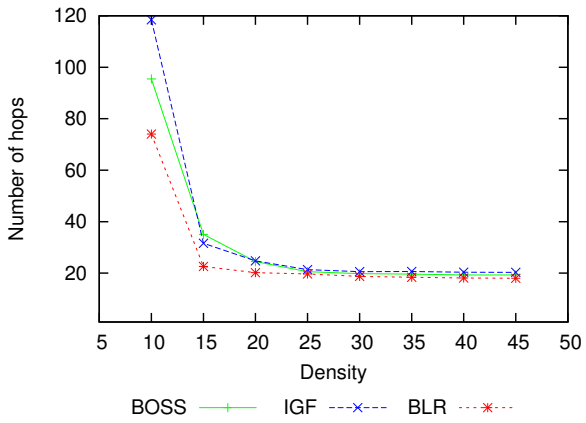


Figure 3.17: Hop Count

Fig 3.17 shows the mean number of hops to successfully reach the destination. We measure the hop count of the first data packet arriving the destination and ignore the duplicates arriving later. As we can see, the three algorithms achieve a similar number of hops near to 20 which is closer to the theoretical limit of 14. Because they use the same greedy strategy based on the advance criterion to minimize the destination distance in each hop using the closest neighbors. In dense networks, the greedy strategy provides near the optimal solution for the

three protocols.

3.3.4 Real-TestBed Network

Existing studies [9] have shown that most routing protocols have been evaluated through simulators modeling wireless communications by simplified conditions of realistic WSNs. These models do not consider the typical problems in wireless communications such as radio range variability and link asymmetry. Here, BOSS is compared with IGF and BLR in a testbed indoor scenario in order to assess their reliability and robustness in realistic deployments. Moreover, we analyze the performance of the three protocols through simulation of the same scenario. The goal is to compare the experimental and simulated results for studying the effects of the realistic wireless communications in each algorithm.

The experimental scenario consists of 35 sensor nodes distributed within the first floor of the Computer Science building at the University of Murcia as shown in Fig 3.18. The sensor nodes deployed are TmoteSky [69] motes from the company Sentilla. The deployment covers an approximate area of $75x40m$ with a mean density of 8 neighbors. We use different colors to indicate the quality of links between nodes representing the packet reception ratio in wireless communications.

To determine the quality of links, we utilize a periodic beacon scheme in every mote and calculate the mean number of packets received during ten hours. In addition the results show that the mean radio range of nodes was 45 meters. This is particularly interesting because the results confirm that the links quality is not directly related to the distance. For instance, some links with high quality over 80% have twice the distance than other links with less quality than 30%. Obviously, the main cause for these variations are the effect of obstacles, reflexions, refractions, etc. All these effects severely degrade the performance of geographic routing protocols.

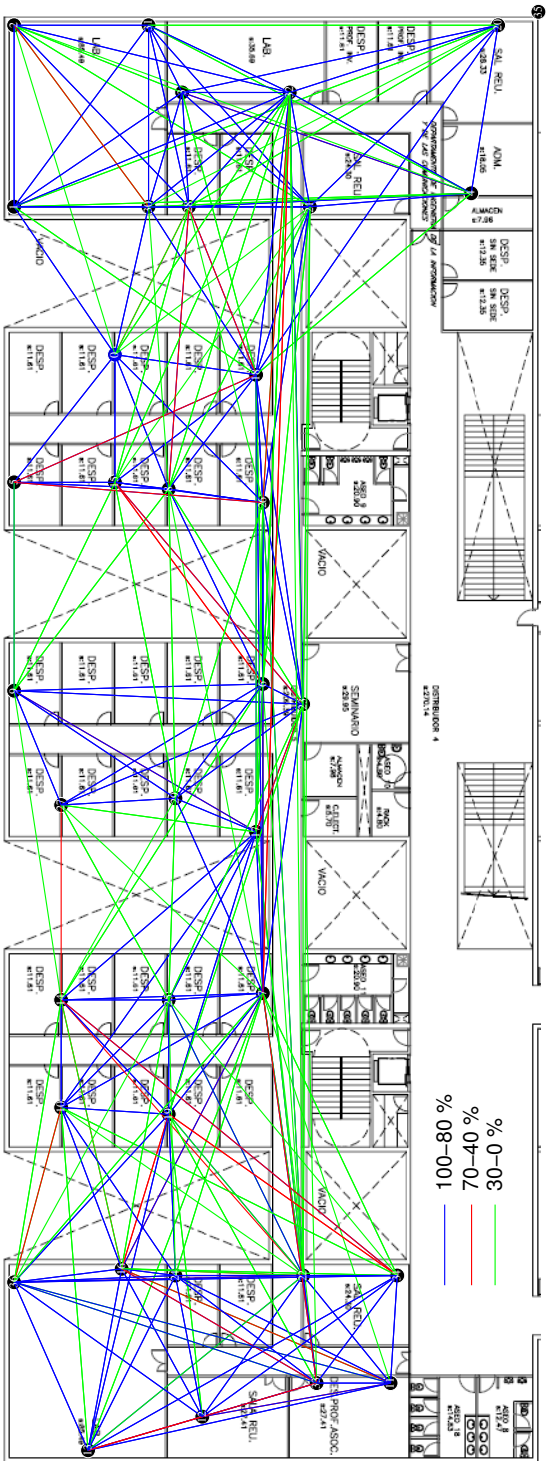


Figure 3.18: Deployment in first floor of Computer Science building

We develop an event log system to monitor the operation of the routing protocols during its running in the testbed, shown in Fig 3.19. Every TmoteSky is connected through a USB port to a NSLU2 device [141] which works as a bridge between the node and a log server. Each NSLU2 gateway has an Ethernet port through which the connected node logs all its wireless traffic to a central server via TCP/IP. In each experiment each node generates a log entry for each transmission and reception, including all relevant information such as the time, source, destination, size and type of the packet. With this information we obtain cumulative distribution functions (CDF) for the different performance metrics. As the previous section, we measure the performance metrics: end-to-end delay, total number of messages, number of messages per hop, total hop count and packet delivery ratio. This is useful to further process and generate statistical data of each experiment in the testbed network.

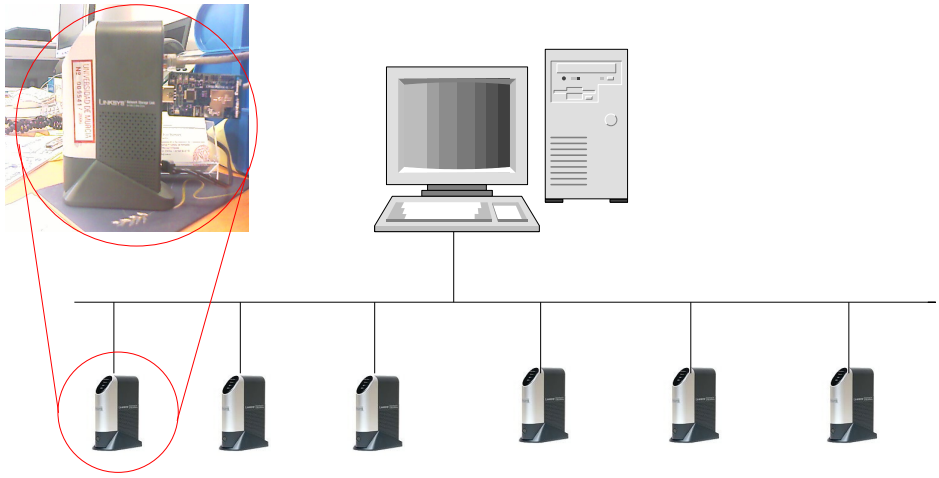


Figure 3.19: Event log system architecture

To compare fairly all geographic protocols, each sensor node is preloaded with information about its location, as well as the coordinates of the rest of nodes. This is done to avoid the additional overhead of positioning systems and location service mechanisms. In this way, we can focus on evaluating the performance of the routing itself.

3.3.5 Real-TestBed Experiments

For real-testbed evaluations, we randomly select 15 nodes as sources and 10 nodes as destinations. Then, each source transmits 25 data packets to each destination. The time between data packets generated by sources is fixed to 20 seconds to guarantee that no previous messages are still traveling in the network. We use the maximum size of data packets with headers which is 120 bytes.

In addition, we simulate the protocols in an equivalent 35 motes scenario using the TOSSIM simulator. The link quality between nodes is derived of our previous analysis (see Section 3.2.1) considering the relationship between the packet size of and the Packet Reception Ratio (PRR). By doing that we try to make the simulations as close to reality as possible.

All beacon-less protocols are configured to wait a maximum time of $300ms$ before starting their face strategies. That is, IGF and BOSS wait for a maximum of $300ms$ for receiving responses, and BLR waits for a passive acknowledgement (next DATA) during a maximum time of $300ms$. In relation to the number of retries, all protocols try to choose a next forwarder up to 3 times before the packet is dropped. Moreover BOSS is configured with 5 positive advance areas.

3.3.6 Analysis of Real-Testbed Results

Fig 3.20 illustrates again the problems of BLR with duplicated packets. As shown, BOSS is again the best protocol in terms of lower number of intermediate copies of data packets. In fact, in 95% of the cases BOSS does not produce any additional copy, and in the remaining 5% of the routing tasks, a single copy is generated. IGF follows closely the results from BOSS. However BLR becomes highly inefficient due to the high amount of duplicated packets.

In Fig 3.21 we plot the CDFs of the number of messages which are sent in face mode. As we see, BOSS and IGF do not often require face mode in our scenario. However, in some cases, they require to recover from some temporal voids generated by radio link variability. In the case of BLR, the usage of face routing is quite extensive. The reason is that BLR again creates so many duplicates

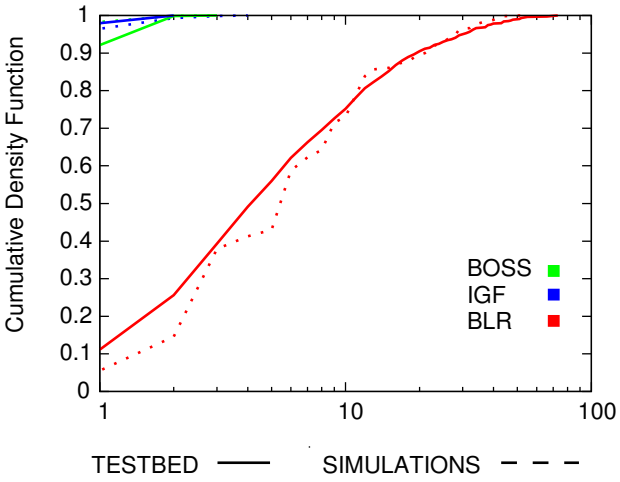


Figure 3.20: Duplicated Packets

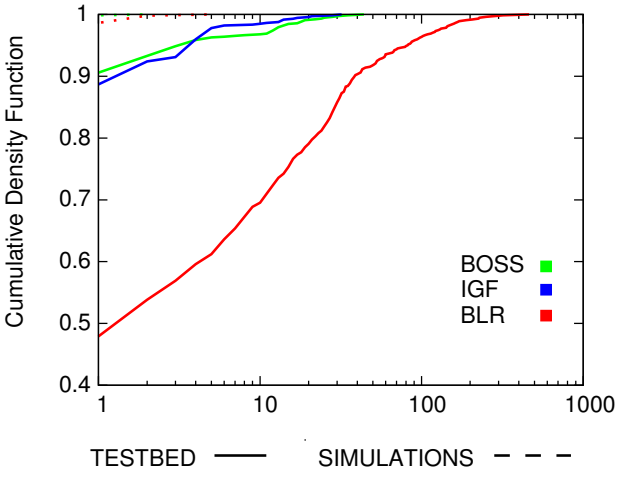


Figure 3.21: Total Face Transmissions

that many of them go along routes which require face mode.

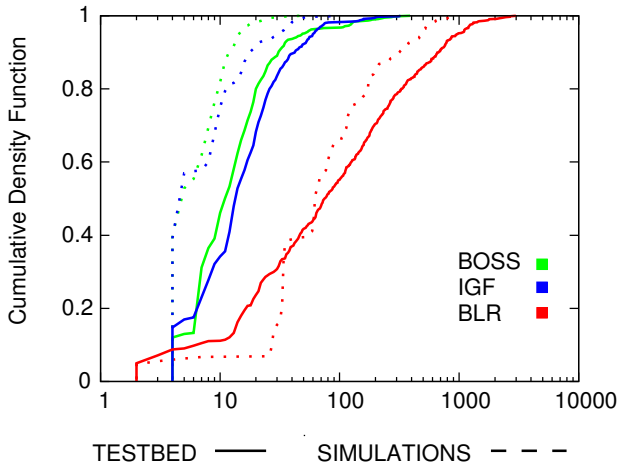


Figure 3.22: Total Transmissions

To study the overhead of the protocols, Fig 3.22 shows the CDF of the total number of messages used by each protocol to reach the destination. As expected, BOSS needs a lower number of messages than IGF and BLR. BOSS reaches 90% of the destinations using less than 30 messages, while IGF requires 40, and BLR needs more than 500. The reason is that in BOSS the neighborhood discovery technique reduces retransmissions in the selection phase since all candidate neighbors require the *DATA* message reception which guarantees a high delivery probability. The poor performance of BLR is also due to the high amount of data copies which are produced in its distributed forwarding scheme. This is totally aligned with the results presented in the previous simulation evaluation.

Similarly, in Fig 3.23 the CDF of the number of messages per hop demonstrates that BOSS offers a high efficiency. BOSS provides an excellent performance by just needing 4 messages per hop in most of the cases whereas IGF requires 6 messages and BLR needs more than 16. This shows that the neighborhood discovery of BOSS consisting of sending first the *DATA* message and using shorter *SELECT/ACK* messages avoids the problem of chosen unreachable next-hop which is presented in IGF. Moreover, BOSS uses the

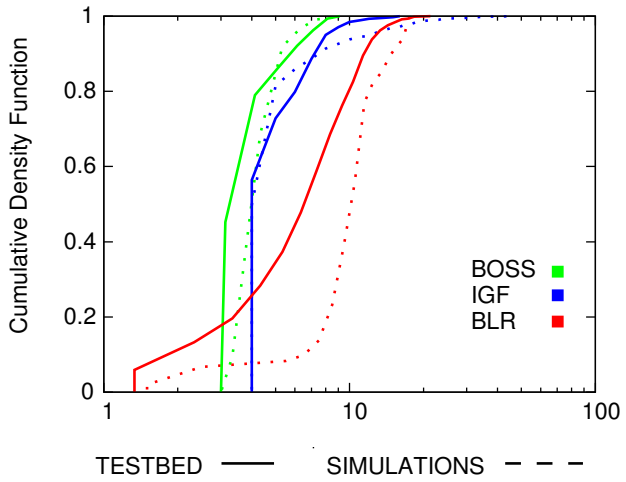


Figure 3.23: Packets per Hop

transmission of the *DATA* message as passive acknowledgement of the previous forwarder reducing the overhead in one packet per hop.

Fig 3.24 shows that BOSS outperforms IGF and BLR. Concretely, in BOSS 80% of the experiments successfully deliver more than a 90% of data packets. The results confirm that the design of BOSS adapts well to realistic wireless communications providing a high reliability. The main reason is that the discovery scheme avoids error-prone neighbors not being able to receive the data packet. Whereas IGF uses a small packet to discover the closest neighbors which may have lossy links.

To study the end-to-end performance of the protocols, we compare the end-to-end delay and the hop count to reach the destination. These metrics evaluate the goodness of the path created by each protocol. Fig 3.25 shows the CDF of the end-to-end delay. All protocols provide a similar average end-to-end delay. The reason is that in the testbed scenario the maximum distance between a source and a destination is very low. However, BOSS achieves clearly a lower end-to-end delay than IGF and BLR. This results show in BOSS the benefit of Discrete Dynamic Forwarding Delay (DDFD) which reduces the contention.

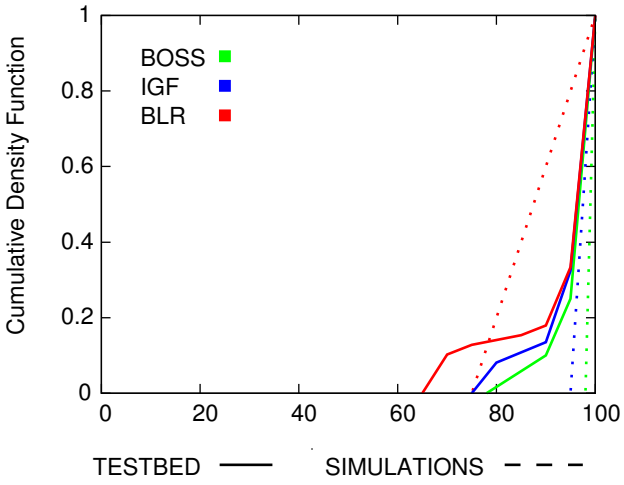


Figure 3.24: Packet Delivery Ratio

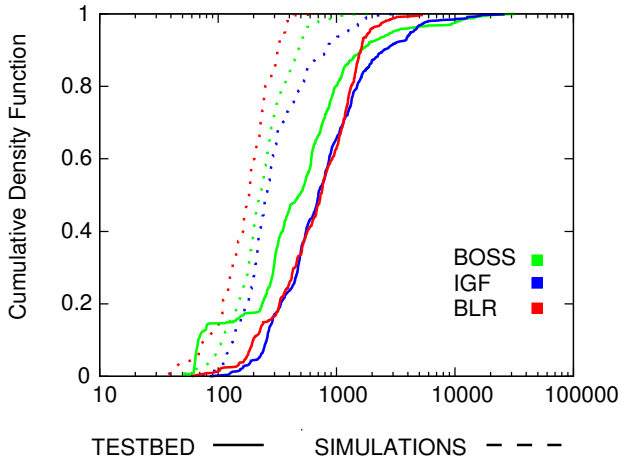


Figure 3.25: End-to-end Delay

On the other hand, BLR should be the fastest algorithm (as shown the previous simulation results) for its fully distributed forwarding that only contains an unique data packet per hop. However the BLR performance is highly penalized for the huge amount of duplicates producing excessive contention at the MAC layer. Therefore nodes employ frequently face routing incrementing the end-to-end delay. Additionally, all the protocols behave better in the simulator than in the real-testbed. The reason is that the simulator does not consider common communication problems of real deployment such as radio range variability and link asymmetry.

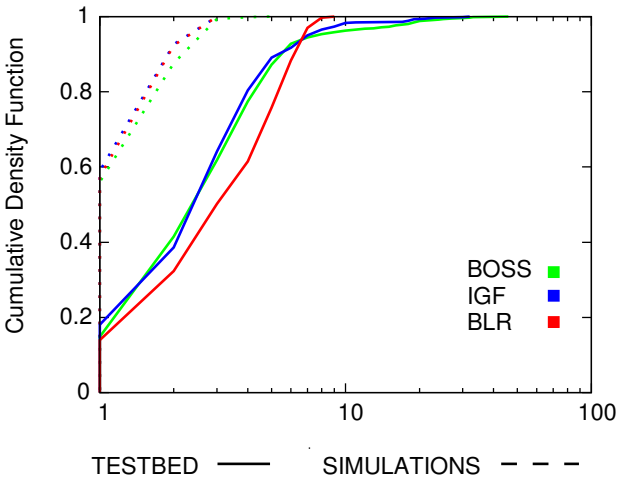


Figure 3.26: Hop Count

Fig 3.26 shows the length of the shortest path found by the protocols. As expected, BOSS and IGF tend to use paths with similar lengths whereas BLR uses slightly shorter paths. Because in the testbed the number of reasonable greedy paths between each source and each destination is not very high. So all protocols tend to choose very similar paths in terms of hop count. There are very few cases in which BOSS and IGF need more than 10 hops to reach the destination. The reason is that even though a greedy path exists from a source to a destination, these protocols rarely enter in face routing due to the losses of greedy responses

by collisions or interferences. On the other hand, in BLR many duplicates make that the packet travels along the shortest paths.

3.4 Conclusions

This chapter proposes BOSS, a new beacon-less routing protocol for realistic communications in WSNs. BOSS is designed to deal with losses, collisions and interferences common in wireless communications.

To improve the knowledge about wireless communications, we made several empirical experiments with the CC2420 radio used by most of existing sensor devices. The experimental results confirm the strong relationship between the packet size and the packet reception ratio. Our analysis concludes that the bigger packet is, the less reception probability is, when the distance between the sender and receiver is near to the radio range.

According to previous experiments, we design a neighborhood discovery scheme based on the idea of sending the big *DATA* message including the payload first to discard neighbors with weak links that are not able to receive it and become the next hop. In this way our selection phase is performed with smaller control messages reducing packet losses and retransmissions.

Moreover, BOSS incorporates a delay function that combines greedy and face strategies. The function divides a coverage area into subareas where the delay time is assigned randomly to neighbors in order to reduce collisions and simultaneous responses. BOSS also guarantees the hop-by-hop delivery of the packets using a selection scheme based on retries and passive acknowledgments which also reduce the control overhead.

Several experiments have been performed to evaluate the performance of BOSS against the most important protocols in the field of beaconless routing (IGF and BLR) employing both extensive simulations and a real-testbed network. The performance evaluation indicates that BOSS succeeds in achieving a much lower number of transmissions (totals and per hop) while keeping the delivery ratio above the 90%. In addition, the empirical study in the real-testbed also

confirms that BOSS outperforms IGF and BLR in terms of reliability, transmission efficiency and end-to-end performance. In conclusion, all results show that BOSS is an efficient and reliable solution to deal with the inherent problems of error-prone wireless communications in WSNs.

After studying the common problems generated by realistic wireless communications, we now focus on dealing with other assumptions considered by geographic routing protocols. Concretely, in the next chapter we treat to analyze the effects of inaccurate positions in geographic protocols in order to design an effective solution.

Chapter 4

Geographic Routing in Networks with Location Errors

As the previous chapters demonstrated, Geographic Routing (GR) is a scalable and efficient solution for Wireless Sensor Networks (WSNs). The reason is that nodes only need local information about neighbors positions to take forwarding decisions. Based on that, forwarding nodes select the next-hop based on neighbors providing advance toward the destination. This is called Greedy Routing Strategy (GRS) [112]. In same cases, the data packet reaches a node having no neighbors closer to the destination than itself. In that case we say that the packet reaches a local maximum. The node has a void area, and a recovery scheme (i.e face routing) is used to resume greedy mode.

However, most geographic routing protocols have been designed and evaluated assuming perfect location information. They neglect the inaccuracy of localization systems employed in real deployments [10]. Recent studies [11] have proven that existing geographic solutions are ineffective when the position of nodes is inaccurate. Both beacon and beaconless protocols experiment a huge increment of packets losses as location error increases [12]. Concretely, in greedy mode the main reason of dropped packets is void areas, and 90% happens in the destination range [13]. For this reason, this chapter is focused on studying the effect of location errors in greedy mode and designing an effective and efficient

solution.

Here, we propose an Effective Greedy routing protocol supporting Location Errors (EGLE). First, we study the presence of inaccurate positions and the causes of failures of greedy routing. Our study shows that void areas appear due to position inaccuracy and they are the main failure situations of greedy routing. Based on our study, EGLE provides three operation modes considering location errors to mitigate the packet losses for void areas. A greedy selection heuristic prevents reaching local maximums. An alternative forwarding in a limited region of the local maximum permits to exit the void areas. And, a lightweight broadcasting strategy propagates the data packet in a reduced area near the estimated position of the destination to guarantee the delivery.

We compare EGLE with existing solutions in the literature in simulated networks and a realistic testbed. By extensive simulations, we assess the scalability and efficiency of EGLE in dense networks with thousands of nodes. In the testbed, we validate the reliability and robustness of EGLE in realistic communications of WSNs. The simulation and experimental results show that EGLE exhibits delivery ratios higher than 90% even in scenarios with 100% location errors, outperforming existing solutions.

In this chapter, the main goals are:

- Classifying the issues caused by location errors that reduce the performance of greedy routing.
- Analyzing in detail the main cause of packet losses in greedy routing (i.e. void areas) due to location errors.
- Proposing a reliable beaconless geographic routing algorithm to mitigate the effects of location errors and achieve a high delivery ratio and a little overhead.
- Evaluating the proposed protocol in comparison with relevant beaconless geographic routing algorithms using extensive simulations and a real deployment.

4.1 Related Work: Geographic Routing with Location Errors

Most geographic routing protocols have been designed assuming perfect location information [130, 132]. They assume in their evaluation that nodes know their accurate positions. However, given that WSNs are formed by devices with constrained resources such as power and computation. In realistic deployments, sensor nodes often use efficient distributed positioning systems to estimate their positions [99, 100, 101]. These distributed systems generate inaccurate positions with an associated error deviation. In particular, localization systems may produce errors as high as 100% of the radio range, as proven in [10]. Authors of this paper analyze the error patterns of various location systems for WSN and the influence of location errors in geographic routing protocols. Their analysis demonstrates the importance of considering location errors in the design of effective geographic algorithms.

4.1.1 Issues of Location Errors in Greedy Routing

In this chapter, we focus our efforts on the issues caused by location errors in Greedy Routing Strategy (GRS) [112] in beacon and beaconless protocols. Several studies [12, 13, 142] show the impact of location errors in geographic routing protocols. Based on existing studies in scenarios with inaccurate locations, we classify the issues caused by location errors that reduce the performance of the greedy strategy. These issues affect greedy routing in terms of latency, number of transmissions, delivery ratio, etc.

- Transmission Failure. This happens when the current forwarder selects as next hop a neighbor which is really located outside its radio range and cannot receive the packet.

- Backward Progress. This occurs when the forwarder selects a neighbor which is located really farther toward the destination than itself.

- Void Area: This appears when the forwarding node has no neighbors closer to the destination, even though in reality there are neighbors closer to the destination.

Regarding transmission failures, this issue affect some beaconless protocols using small messages to discover neighbors as we shown in Chapter3. In particular, those protocols based on CTS/RTS handshake scheme (i.e. IGF and GeRaF) are affected whereas protocols (i.e. BLR, CBF and BOSS) based on the idea of sending the data packet first are not affected. The reason is that these schemes sending the data packet first guarantee that only reachable neighbors receiving this big packet from the forwarder participate in the selection phase employing small control messages. However, protocols based on periodic small beacons are highly affected since nodes select next-hops by their neighbors tables which may not be updated and may contain unreachable neighbors. In those cases, the packet does not reach the selected neighbor, and retransmissions are frequent. Note that this issue often happens in scenarios where nodes have some mobility and change their positions.

With regard to backward progress, this problem influences both beacon-based and beaconless protocols. Both protocols employ greedy forwarding to maximize the advance toward the destination in each hop. Concretely, in beacon-based schemes the next-hop selection considers the estimated positions of neighbors obtained by periodic beacon messages. However, a selected neighbor may posses a real position that is farther to the destination than the forwarder. Similarly, this issue happens in reactive beaconless discoveries where the order of neighbors replays is based on their estimated position. In those cases, the packet goes away from the destination increasing the number of hops and end-to-end delay.

Regarding void area, this issue affects both beacon-based and beaconless protocols. Both protocols assume a void area when the current forwarder has no closer neighbors to the destination than itself based on their estimated positions. In beacon-based protocols, this means that the forwarder does not find any closer neighbor in its routing table. In beaconless protocols, that occurs when the forwarder does not receive any response from its neighbors. The reason is that

	Beacon	**Beaconless**
Issues	GRS	$BOSS$
$Transmission\ Failure$	YES	NO
$Backward\ Progress$	YES	YES
$Void\ Area$	YES	YES

Table 4.1: Issues affecting beacon-based and beaconless protocols in greedy mode.

their estimated positions are father to the destination than the current forwarder. In void areas, greedy strategies fail, and the packet is dropped or a recovery strategy (i.e. face routing) is applied.

Table 4.1 summarizes issues affecting greedy routing for the GRS protocol (as representative of beacon-based algorithms) and for the BOSS protocol (as a beaconless). As a summary we can conclude that transmission failure is not a critical issue in WSNs where nodes are often stationary. While backward progress increases the destination distance and the end-to-end delay, but greedy routing can continue in the next hops. However as shown in [13], void area is the most important problem that severely degrades the delivery ratio of greedy routing.

4.1.2 Existing Greedy Solutions Supporting Location Errors.

Geographic routing with location errors has emerged as a relevant topic in the research community [10, 13, 12]. However, we have only found two greedy approaches considering location errors to address the aforementioned issues: transmission failure, backward progress and void areas [11, 142].

Flooding Mechanism and Second Order Neighborhood

Shah et al. [11] proposed two methods mitigating the performance deterioration of the GPSR [132] protocol in the presence of void areas. First, they use a flooding mechanism as an alternative to face routing to guarantee packet delivery

with location errors. When the current forwarder has no closer neighbors to the destination than itself then it floods the packet to them in order to reach to the destination. The packet flooding continues till the destination is found. A second order routing enhances the performance of greedy routing, but requires the overhead of including 2-hop neighborhood information in periodic beacon messages. Using this information, the current forwarder selects the next hop discarding 1-hop candidates which have no neighbors closer to the destination. The results show that the flooding mechanism is able to discover a route to the destination despite the presence of void areas. However, there is a high penalty in the number of transmissions. The 2-hop neighbors information enables a subtle reduction in the amount of packet losses in scenarios with small location errors.

Maximum Expectation within Transmission Range: MER

To avoid the overhead of 2-hop information, Kwon et al. [142] presented the first greedy selection function incorporating location errors to determine the goodness of candidates as next hops, called MER(Maximum Expectation within transmission Range). MER is a probability function that considers two types of greedy routing problems: transmission failure and backward progress. The MER function penalizes nodes whose real positions might cause both transmission failure and backward progress. Nodes forward packets to their neighbors that maximize the MER function. Simulated results confirm the improvement of the delivery ratio for scenarios with moderate location errors. The results also show that MER does not work property in scenarios with location errors higher than 31.5% of the radio range. This protocol is analyzed in the next section in order to better understand the effects of inaccurate positions, in particular the main problem of void areas.

4.2 Analysis of Greedy Routing with False Void Areas

This section analyzes the effects of location errors in two greedy routing protocols: GRS [112] and MER [142]. GRS employs the typical greedy selection based on minimizing the distance toward the destination. MER provides a probability selection function considering location errors to improve the delivery ratio. We assume an underlying topology in which nodes are well-distributed without void areas and greedy routing suffices to deliver the message. The analysis shows that even in this idealistic topology, location errors may make greedy routing to enter into local maxima as well as provoke delivery failures.

Let us assume a current relay i having a packet addressed to a destination node d outside of its radio range R. Then, i has a set Q of neighbors $j \in Q$ located within R to route the packet. Assuming a topology without void areas, in both GRS and MER i selects one of its neighbors j located closer to d than itself. However in realistic scenarios, every node a located at position A estimates an inaccurate position A'. So, in practice i selects a neighbor j whose inaccurate position J' is closer to D' than I'. If any neighbor satisfies this condition, i thinks that there is a void area and it becomes a local maximum.

A false void area happens for two reasons presented in Fig 4.1 where for simplicity we assume that $D' = D$. First, if the relay i has an estimated position I' that is closer to D than I the inaccurate greedy area is smaller. In Fig 4.1, the inaccurate greedy area is represented for the small gray region which does not contain any node acting as forwarding candidate. Second, if every greedy neighbor j has an estimated position J' that is farther to D than J, then J' is placed outside the greedy area of i. Both conditions produce false void areas and i becomes a false local maximum. This happens when:

$$\forall j \in Q, \quad dist(I', D') < dist(J', D') \tag{4.1}$$

Where, $dist(A, B)$ represents the Euclidean distance between positions A and B.

Below, we describe in detail the selection function used by GRS and MER and their behavior with false void areas. In particular, GRS selects the node

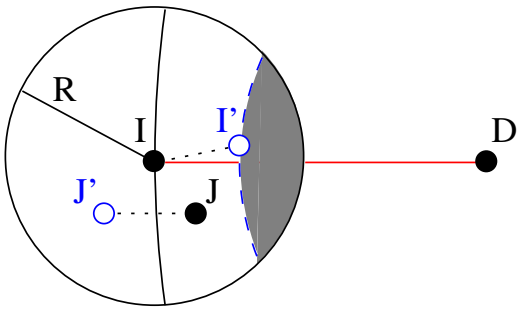


Figure 4.1: A false void area appears due to the inaccurate positions (I', J') of i and j, being I and J their real positions, respectively.

minimizing the distance toward the destination. This mean that the current relay i selects the neighbor j that maximizes the progress to the destination d:

$$P(i, j, d) = dist(J', D') - dist(I', D') \tag{4.2}$$

The authors of MER designed a probability selection function considering inaccurate location information. The MER selection function incorporates location errors to prioritize nodes whose real positions are likely inside the greedy area of the current node. To do that, MER function penalizes nodes that may cause the issues of backward progress or transmission failure. For preventing transmission failure, MER penalizes nodes whose real position may be outside its radio range. For avoiding backward progress, MER penalizes nodes whose real position may increase the real distance toward the destination. Thus, the current relay i penalizes neighbors j whose distances are near to the radio range and those neighbors j whose progress toward the destination is small using the following equation:

$$MER(i, j, d) = P(i, j, d) * F_i() \tag{4.3}$$

Where, F_i is a Cumulative Distribution Function taking values between zero and 1, and penalizes the goodness of a neighbor j as next relay for its proximity to the sender i or to its radio range R, considering their inaccurate positions J' and I', respectively.

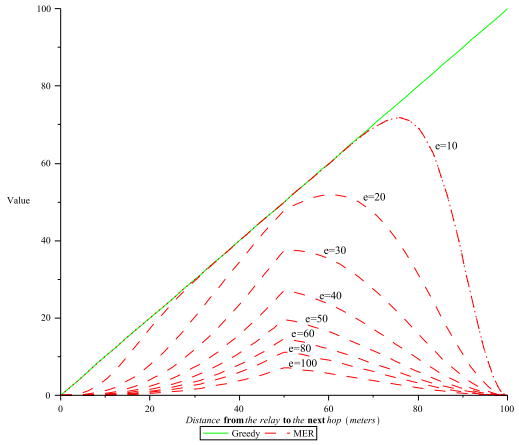


Figure 4.2: Comparing the selection function of MER and GRS at increasing distance from a relay to a candidate neighbor when the radio range is 100 and for different percentages of estimation errors.

A comparison between the selection function of GRS and MER is shown in Fig 4.2. In this figure, we assume that the neighbor j chosen as next hop is located on the line between the current relay i and the destination d and the radio range is $R = 100$. The green line represents the selection of GRS which does not consider location errors. In GRS, the selected next hop is determined by maximizing progress $P(i, j, d)$. Thus, the longer advance from the current relay i is, the better next hop is. The red lines represent the selection of MER when the location error (e) takes the following percentages of the radio range: 10, 20, 30, 40, 50, 60, 80, and 100 %. In MER, the selected next hop is determined by maximizing objective function $MER(i, j, d)$. When the location error is low the objective function selects the neighbor j that provides the most progress to the destination d. However, when the error is high the objective function penalizes neighbors j providing a high progress to the destination d or being located close to the current relay i. In scenarios with high location error, the current relay i tends to chose the neighbor j in the center of its radio range $P(i, j, d) = 50$ as we can see in Fig 4.2.

Now, we demonstrate through an example that both GRS and MER may reach a false local maximum employing their next-hop selection functions. That is, they are still prone to failures in scenarios with location errors. Fig 4.3 shows a node n_0 having a packet addressed to the destination d located at position D. For simplicity, we assume that every node n_i knows its real position N_i except the node n_2 located at position N_2 that estimates an inaccurate position N_2'. As we see in the figure, n_2 thinks based on its estimated position that it has no neighbor closer to d than itself even if that is not the realistic situation.

In GRS, the selection function minimizes the distance to the destination. Then, n_0 selects n_2 whose estimated position N_2' is closer to D than N_3. So, GRS fails.

In MER, the selection function penalizes the goodness of greedy neighbors for their proximity to the current relay and for being nearly as far as the radio range. In this example, n_0 selects n_1 because n_2 is near the radio range. In the next hop, n_1 must select a next relay among its closer neighbors n_2 and n_3. Neighbors n_2 and n_3 are good candidates because their positions N_2' and N_3 are within the greedy area of the position N_1. Then, n_1 selects the neighbor n_2 whose inaccurate position N_2' is closer to D than N_3. In this hop, the forwarding to n_2 generates backward progress because its real position N_2 is really farther to D than the previous position N_1. So, MER also fails in this situation.

Both GRS and MER end up falling into a false local maximum for two different causes. GRS is prone to reach a local maximum because its function selects nodes with excessive distance from the previous relay. Although MER avoids the local maximum in the first instance. After several hops, the selection of a previous candidate may generate a local maximum due to backward progress. Moreover, in both GRS and MER, n_2 discards n_3 as candidate relay because n_2 thinks that n_3 provides no progress. However using n_3, the packet is able to exit the false local maximum and advance through n_4 toward d.

Finally, we study a special case of false void area where the current relay i thinks that it is able to deliver the packet to d directly. Both GRS and MER assume the perfect delivery within the radio range R of the destination. However, the delivery fails if the real distance between i and d is larger than R. Fig 4.4

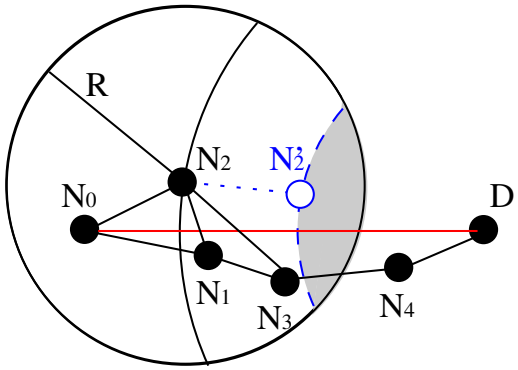


Figure 4.3: A greedy path exists between n_0 at position N_0 and d at position D through nodes n_1, n_3 and n_4, being really located at positions N_1, N_3, N_4, respectively. But the protocols (GRS and MER) reach a false local maximum n_2 at position N_2 because of the wrong estimated position N_2'.

shows an example of delivery failure where the inaccurate distance between i and d is lower than the radio range R.

Summing it up our analysis shows that false void areas may appear and existing greedy solutions fail to deliver packets in scenarios with location errors. Greedy routing protocols reach local maxima due to the selection of nodes that have been previous candidates or have excessive distance. According to inaccurate positions, a local maximum discards neighbors that are able to exit the false void area and provide advance toward the destination. The destination has an estimated coverage area that does not ensure the delivery of the packet. For all these reasons, it the next section we design a geographic routing protocol being able to deal with the effects of location errors.

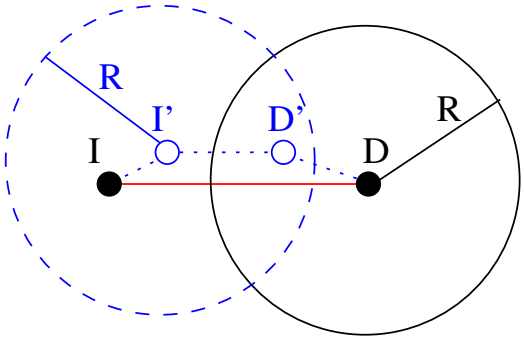


Figure 4.4: In (c) i fails the delivery to d because in reality the distance between them is larger than R ($dist(I', D') < R < dist(I, D)$).

4.3 Effective Geographic Routing with Location Errors (EGLE)

This section describes the operation of EGLE and shows how it addresses all the issues mentioned above. We first give an overview of the overall operation and then describe the details of the routing protocol.

The design of EGLE is based on three conclusions of our previous analysis in realistic networks with location errors. First, our analysis demonstrated that local maxima may be reached from backward progress and excessive distance. Second, although a current relay has really closer neighbors to the destination, a false void area might appear, and the relay discards these neighbors as next hop candidates due to location errors. Third, even if the distance among sender and destination is lower than the radio range, the packet delivery may fail.

The main contributions of EGLE consist of three effective mechanisms to deal with location errors and an efficient delay function to reduce the number of hop-by-hop transmissions. The main building blocks of the protocol are as follows:

1) Advance toward destination avoiding local maximums. EGLE provides a greedy heuristic to select the next hop that combines two objective functions to

prevent local maxima coming from backward progress and excessive distance. The first penalizes neighbors that already took part in the forwarding of the same data packet few hops before. The second penalizes neighbors whose positions are too far from the sender.

2) Continue through false void areas. In void areas, EGLE proposes an alternative mode that uses 1-hop neighbors discarded by the local maximum node to find a 2-hop neighbor closer than itself to the destination. The alternative mode requires only knowing neighbors of the local maximum, but highly improves the packet delivery ratio.

3) Reduce the number of transmissions. EGLE presents a sophisticated timer assignment function to prioritize the answers from good candidates as next hops. This alleviates the bandwidth consumption and reduces collisions at the MAC layer which increases EGLE's reliability.

4) Deliver the packet inside the destination radio range. Finally, EGLE applies a limited broadcast scheme [143] to disseminate the data packet in a region around the destination position. This low-overload broadcast scheme is only performed in a limited region, but improves significantly the delivery ratio.

4.3.1 Network Model and Assumptions

As most geographic protocols, we assume that all nodes know the fixed radio range R although in practice R is estimated as the mean length of all links in the network. Nodes can also calculate their own positions using any positioning system based on extra hardware (i.e GPS [98]), distributed algorithms (i.e DPE, APS, RPE [101, 99, 100]) or virtual coordinates [144, 145]. Moreover, the source node must employ any scalable location service to determine the position information of the destination [5]. So, the source includes in the packet the destination's estimated position which is employed by intermediate nodes to take routing decisions.

In realistic scenarios, the position information is inaccurate and contains an error with respect to the radio range. Each node a located at position A has an estimated position A', where $A = A' + W$, being W a Gaussian random vector

with zero mean and standard deviation σ_a. Based on the 3-sigma rule of the Gaussian distribution [146], 65% of the samples fall into the range of one standard deviation. For simplicity, σ_a is the maximum position difference between A and A'. Given the current relay i holding the packet addressed to the destination d and a set of neighbors $j \in Q$ receiving the transmissions of i in its radio range R , we define σ_{ij} as the location error of a neighbor j with respect to the current relay i, denoted as:

$$\sigma_{ij} = \sqrt{\sigma_i^2 + \sigma_j^2} \tag{4.4}$$

Where σ_{ij} represents the maximum distance difference between their real distance ($dist(I, J)$) and their estimated one ($dist(I', J')$). Thus, the maximum estimated distance between the current relay's position I' and each neighbor's position J' is denoted as:

$$MaxDist_{ij} = R + \sigma_{ij} \tag{4.5}$$

Considering the location errors, we define the progress of a neighbor j to the destination d with respect to the current relay i to be:

$$P_{ij} = dist(J', D') - dist(I', D') \in [-MaxDist_{ij}..MaxDist_{ij}] \tag{4.6}$$

This model of inaccurate positions and the estimation of the maximum progress $P_{ij} = MaxDist$ is summarized in Fig 4.5. Where for simplicity the relay i, neighbor j and destination d and their real positions (I, J and D) and estimated positions (I', J' and D') are placed along the same line.

4.3.2 BOSS Forwarding in Greedy and Alternative Modes

In greedy and alternative mode, EGLE applies the beaconless forwarding of BOSS adapted to networks with location errors. Similarly to BOSS, EGLE employs four messages *DATA*, *RESPONSE*, *SELECT* and *ACK* to discover neighbors reactively and select the next hop. As we demonstrated in Chapter 3, the first sending of the *DATA* message including the payload reduces packet losses and retransmissions

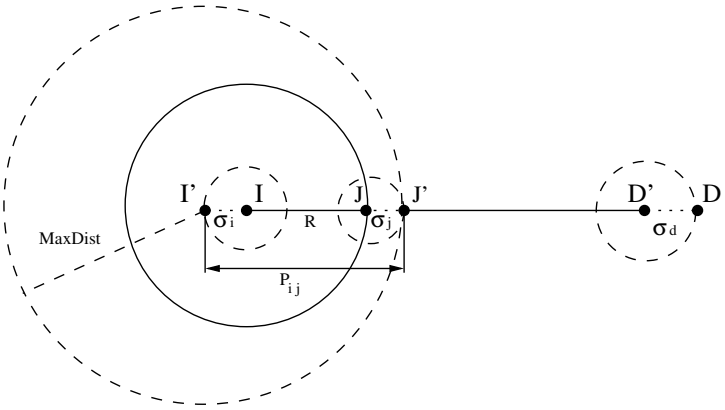


Figure 4.5: Modeling the maximum progress $P_{ij} = MaxDist$ between a relay i and a neighbor j toward a destination d considering their estimated positions I', J' and D', respectively.

in realistic wireless networks. For scenarios with inaccurate locations, the source node includes the identifier d, estimated position D' and associated error σ_d of the destination node in the *DATA* message such as:

$$< identifier, position, deviation >$$

EGLE uses greedy mode as much as possible and incorporates an alternative mode which is only used when no advance in greedy mode is possible. The bit of Routing Mode indicates greedy ($RM = 0$) or alternative ($RM = 1$) mode. In greedy mode, the current relay i holding the data packet broadcasts a query *DATA* message which also includes its identifier, estimated position I' and associated error σ_i. In this way, every neighbor $j \in Q$ that successfully receives the *DATA* message calculates its goodness as next hop considering the location errors of the relay σ_i, destination σ_d and itself σ_j. The neighbor j with the best goodness waits less time and transmits first its *RESPONSE* message including its identifier. Finally, the relay i indicates the next relay by sending a *SELECT* message containing the identifier of the chosen neighbor j.

When the current relay i has no neighbors j with $P_{ij} > 0$, it applies the

alternative mode. Then, the current relay i becomes a local maximum m located at position $I = M$ with estimated position $I' = M'$ and associated error $\sigma_i = \sigma_m$. In alternative mode, the *DATA* message contains the estimated position M' and its associated error σ_m of the local maximum m. Each neighbor $j \in Q$ calculates its answering delay time according to its goodness as next hop considering the location errors of the local maximum σ_m, destination σ_d and itself σ_j.

For simplicity, the next subsections describe the greedy and alternative modes considering that the current relay receives answers from all its neighbors and selects the best candidate as next hop. Later we present a combined delay function that determines greedy and alternative strategies in such a way that when the first node answers the other responses are canceled by the remaining neighbors.

4.3.3 Greedy Heuristic to Prevent Reaching Local Maxima

Our greedy selection of the next relay combines two objective functions to prevent reaching false local maxima. In greedy mode, EGLE uses the neighbors j of the current relay i providing advance toward the destination d, denoted as $P_{ij} > 0$. First, EGLE penalizes neighbors j that previously took part in the forwarding of this data packet one or a few hops before. Among neighbors with less number of previous forwardings, EGLE penalizes the goodness of neighbors j whose positions J' are too far from I'. Below we explain each one of two functions and then show the operation of EGLE using an example.

1) Penalizing neighbors that take part in the forwarding process several times. As we demonstrated in the section 4.2, a false local maximum may be reach from backward progress. In greedy routing the data packet advances toward the destination using the closest node in each hop. However in scenarios with location errors, the current forwarder is not able to select the neighbor whose real position is closest to the destination. For this reason, nodes may receive the packet from several forwarders, and they take part as candidates in the forwarding process in several hops. In greedy selection those nodes participating previously in the packet forwarding are prone to generate backward progress.

To avoid backward progress, we exploit the wireless medium and the

forwarding scheme of BOSS to identify and penalize nodes that participated in the forwarding before. In BOSS, the current relay i broadcasts first a *DATA* message including the data packet. So, the data message also serves to discover neighbors. Each neighbor j receiving the *DATA* message replies as a forwarding candidate. Among all neighbors, the relay i selects a neighbor as next hop. This process is repeated several times until the destination or a local maximum is reached. The main idea is to order the goodness of neighbors based on the number of times they have acted as candidates. To do that, every neighbor j saves temporally a counter $NumR_j$ storing the number of times the same *DATA* message is received from different relays i. To do that, data packet is identified by the sequence number and identifier of the source node. Each neighbor j incorporates its counter $NumR_j$ which is between zero and $MaxR-1$ in its response. We define $MaxR$ (Maximum Data Reception) as a constant representing the maximum number of allowed receptions of a data packet. This constant will use to determine the delay time of neighbors in order to reduce the number of responses. The lower values of $NumR_j$ is, the better are the candidates. Among neighbors j with the lowest value of $NumR_j$, the current relay i selects the next hop considering their positions and their location estimation errors as explained below.

2) Penalizing neighbors whose distances from the current relay are larger than the radio range. To select the next relay, we use a probability function considering the location errorσ_{ij}. The goal is to penalize the goodness of a neighbor j which may cause a false local maximum because of the probability of being in reality farther than the radio range. To do that, we define the margin of a neighbor j from the current relay i to the maximum distance $MaxDist_{ij}$ to be $M_{ij} = MaxDist_{ij} - dist(J', I')$. Similar to MER [142], our probability function is represented as a Cumulative Distribution Function following a Rayleigh Distribution. We define the probability distribution F_{ij} that a neighbor j is located within the area centered at J' and the radio $u_{ij} = M_{ij}$ with respect to I', denoted as:

$$F_{ij} = (1 - exp(-\frac{u_{ij}^2}{2\sigma_{ij}^2})) \tag{4.7}$$

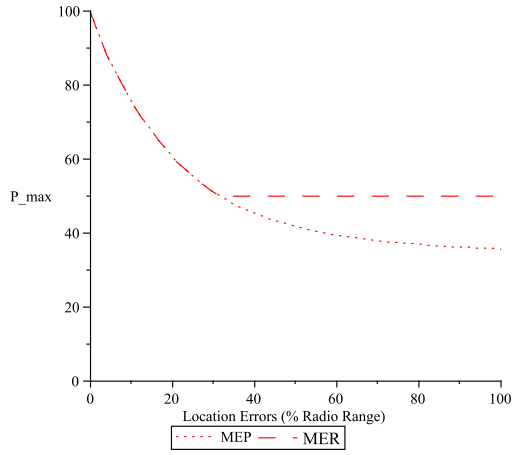


Figure 4.6: The threshold of MER in 31.5% of the radio range.

Where a larger location error (σ_{ij}) penalizes more the distance $dist(J', I')$. And, the higher distance $dist(J', I')$ between a neighbor j and the current relay i is, the lower values of M_{ij} and F_{ij} are. The goodness of a neighbor j with an estimated progress P_{ij} is defined as:

$$MEP_{ij} = P_{ij} \cdot F_{ij} \tag{4.8}$$

Where $MEP_{ij} \in [0..R]$ is called Maximum Expectation Progress (MEP). For $P_{ij} > 0$, the value of MEP_{ij} increases between zero and R almost lineally with the distance $dist(J', I')$. But, the value of MEP_{ij} decreases exponentially down to zero when the distance $dist(J', I')$ is between R and $MaxDist_{ij}$.

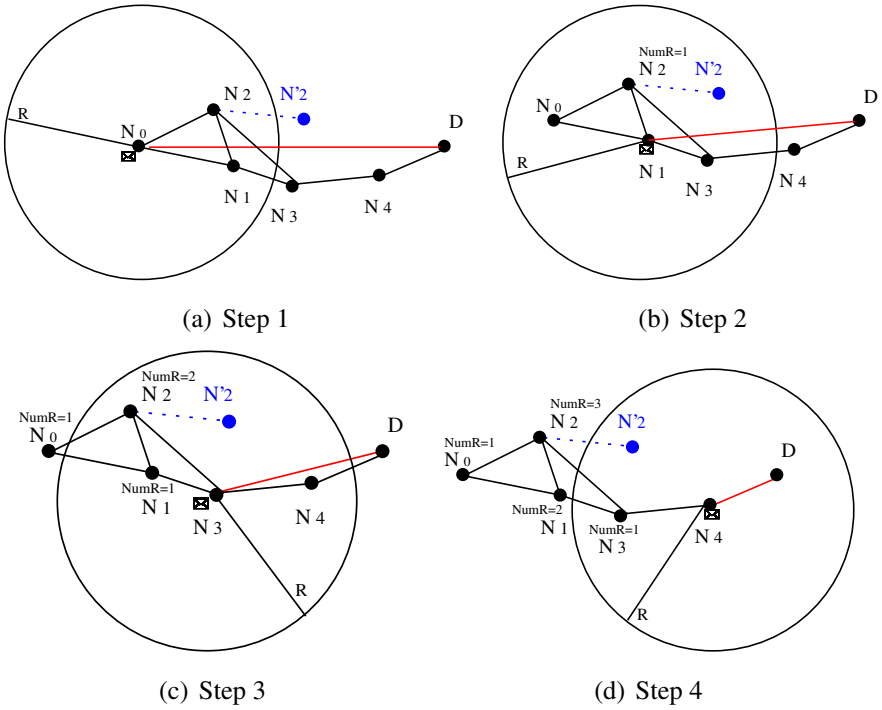
Analysis of EGLE Greedy Heuristic

Our function MEP_{ij} is different to the function used in MER [142] which employs $u_{ij} = min(M_{ij}, P_{ij})$. MER penalizes neighbors for two contradictory conditions: backward progress and excessive distance. This means that MER tends to choose neighbors j in the middle between the current relay i and the radio range R. As its authors determined, MER has a threshold location error

at $\sigma_{th} = 31.5\%$ of the radio range (see Fig 4.6). Where the lines represent the value of progress P_{max} to achieve the maximum value of the MER and MEP functions at increasing the location errors from zero to 100% of the radio range ($R = 100meters$). For simplicity, we consider that the selected neighbor j is located on the line between the relay i to the destination d. As we can see, after the threshold $\sigma_{th} = 31.5\%$, the increment of σ does not affect the MER selection function because MER_{max} is always obtained by neighbors with $P_{max} = 50$ and MER does not decrease. For this reason, MER behaves badly in networks with higher location errors(σ_{ij}) than 31.5%.

Unlike MER, the greedy heuristic of EGLE combines two different objective functions to prevent backward progress and excessive distance. The current relay i prevents backward progress by ordering neighbors with lower receptions of the same data packet. Among neighbors with the least data receptions, the current relay i selects the neighbor j that maximizes MEP_{ij}. The function MEP penalizes only neighbors with excessive distance according to the location error (σ_{ij}) of the radio range R. When the location error(σ_{ij}) increases from 0 to R, MEP tends to choose neighbors j which are nearer to the current relay i. For this reason, our function MEP behaves properly in networks with higher location errors even as high as a 100% of the radio range.

Example 4.7 shows the greedy operation of EGLE assuming no communication errors. The node n_0 has a data packet addressed to the destination d. In the step 1, the current relay n_0 broadcasts a *DATA* message to discover the forwarding candidates n_1 and n_2 with positive advances. Neighbors n_1 and n_2 send their *RESPONSE* messages with their number of previous data packet received $NumR_1 = 0$ and $NumR_2 = 0$, respectively. Among n_1 and n_2 with the same receptions, the current relay n_0 employs the probability function of MEP to choose the next relay. The current relay n_0 penalizes the neighbor n_2 because its estimated position N_2' is further than the radio range R from the position N_0. This means that the candidate n_2 may be a false local maximum. Then, n_0 sends the *SELECT* message to n_1 whose position N_1 is within the radio range R of the position N_0. In the step 2, the current relay n_1 broadcasts a *DATA* message to



(a) Step 1 (b) Step 2

(c) Step 3 (d) Step 4

Figure 4.7: An operation example of EGLE's greedy heuristic.

discover the forwarding candidates n_2 and n_3 with positive advance. Neighbors n_1 and n_2 send their *RESPONSE* messages including their number of previous data packet received $NumR_2 = 1$ and $NumR_3 = 0$, respectively. The candidate n_2 has more receptions than n_3 and may generate backward progress. Then, the current relay n_1 sends the *SELECT* message to the neighbor n_3 whose number of receptions $NumR_3 = 0$ is lower than $NumR_2 = 1$. In the steps 3 and 4, the current relay n_3 uses the next hop n_4 to deliver the packet to the destination d. This example shows how the greedy operation of EGLE improves the performance of existing greedy protocols in presence of inaccurate locations.

4.3.4 Alternative Strategy to Exit from False Void Areas.

When the data packet reaches a node without any closer neighbor to the destination than itself, greedy routing fails and a recovery scheme must be applied. However in scenarios with location errors, face routing is inefficient and ineffective as recent studies have shown [147]. The reason is that scalable and local algorithms are not able to calculate planar graphs without cross-links, and they generate many routing loops (see Section 2.3.3). For this reason, solving void areas produced by location errors in well-distributed networks is an important contribution of this thesis.

EGLE provides an alternative strategy based on our previous analysis about false void areas (see Section 4.2). This analysis shows that in a well-distributed network, a current relay i may have no closer neighbors j to a destination d according their estimated positions I', J' and D', respectively. Then the current relay i becomes a local maximum m located at $I = M$ with estimated position $I' = M'$. We define a false void area when the local maximum m has some neighbors j whose estimated progress is negative $P_{mj} < 0$ (also denoted as $dist(M', D') < dist(J', D')$), but these neighbors j are really located at positions J closer to D than M (denoted as $dist(M, D) > dist(J, D)$). A false void area is produced by the location error σ_{mj} between real and estimated positions of the local maximum m and each neighbor j. As we explained before, σ_{mj} represents the maximum distance difference between the real distance $dist(M, J)$ and the

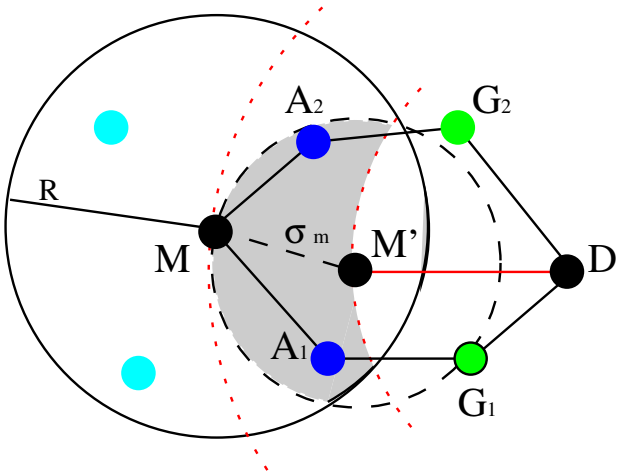


Figure 4.8: An example of false void area and alternative region with location errors

estimated one $dist(M', J')$. For this reason, we propose to forward the data packet in the alternative region A consisting of all neighbors j with negative progress $-\sigma_{mj} < P_{mj}$ that may be really closer to the destination d than the local maximum m.

An example of the alternative region A is shown in Fig 4.8. For simplicity, we assume that all nodes know their real positions except the local maximum m located at M with an estimated position M' which has a data packet addressed to the destination d located at $D = D'$. As we can see, the colored region A of m consists of neighbors a_1 and a_2 whose real positions A_1 and A_2 are closer to D than M. In this case, the packet does not need to apply face routing, since the void area is not real. So, the packet is able to advance using neighbors a_1 and a_2 in the region A whose coverage areas contain greedy nodes (g_1 and g_2) whose positions G_1 and G_2 are closer to D than the local maximum M'. In g_1 and g_2 nodes, the packet can continue in greedy mode.

To exit from a false void area, EGLE uses the beaconless forwarding of BOSS to route the data packet in the alternative region A. When the current relay i receives no *RESPONSE* messages in greedy mode. The current relay i broadcasts

a *DATA* message indicating the alternative mode to discover every neighbor j in the region A. The *DATA* message also contains the estimated position $I' = M'$ and associated error $\sigma_i = \sigma_m$ of the local maximum m. Using this information, each neighbor j determines if its estimated position J' is within the alternative region A, denoted as $-\sigma_{mj} < P_{mj}$. Those neighbors $j \in A$ send their *RESPONSE* messages with their location information (J', σ_j). Among all neighbors, the relay i sends a *SELECT* message to the neighbor j as next hop that minimizes negative progress $(P_{mj} < 0)$. This process is repeated several times until reaching a 2-hop neighbor h of the local maximum m with positive estimated progress $P_{mh} > 0$. Through h, the packet can advance in the greedy mode $RM = 0$.

The main idea of alternative mode is forwarding the packet through neighbors $j \in A$ minimizing negative progress $(P_{mj} < 0)$ The reason is that neighbors inside $j \in A$ with less negative progress have more probability to be closer to d than the local maximum m if real positions are considered. Those neighbors $j \in A$ may have better coverage area than the local maximum m, where there may be some 2-hop neighbors h providing advance to exit from the false void area.

To avoid cycles in alternative mode, we ensure that the data packet is forwarded to different nodes in each hop. To do that, each node can only forward the packet one time to avoid repeating the same selection. Nodes that acted as relays store temporally the data packet marked as sending and ignore the same *DATA* message received later.

The operation of alternative mode provides a high delivery ratio with very small overhead. All neighbors $j \in A$ are candidates to forward the packet. The maximum overhead is limited by the number of neighbors in the area A. If all neighbors $j \in A$ forward the packet without finding a closer 2-hop neighbor h. In this case, the packet should be routed with face routing because that means that is is in a real void area.

Fig 4.9 shows an example of EGLE's alternative mode operation. In this case, we assume for the sake of simplicity that all nodes know their real positions except the node n_2 with estimated position N_2'. The current relay n_2 has a

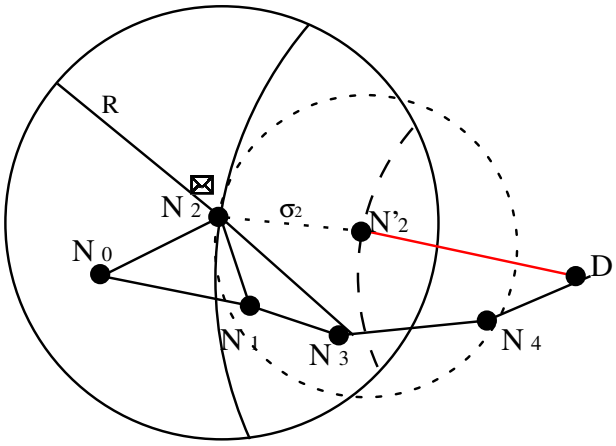


Figure 4.9: An example of alternative mode to exit from a local maximum with location errors

data packet addressed to the destination d and broadcasts a *DATA* message with its estimated position N_2' to discover greedy candidates. The current relay n_2 receives no *RESPONSE* messages because n_2 has a false void area due to its estimated position N_2'. Then, n_2 becomes a local maximum $n_2 = m$ with position $N_2' = M'$ and associated error $\sigma_2 = \sigma_m$. The relay n_2 broadcasts a *DATA* message indicating the local maximum information (M', σ_m) and the usage of alternative mode $RM = 1$ to discover the alternative candidates. Neighbors n_1 and n_3 inside the alternative region of n_2 send their *RESPONSE* messages with their positions N_1 and N_3, respectively. To minimize negative progress, the current relay n_2 sends a *SELECT* message to the chosen neighbor n_3 as the next relay. In the next hop, the current relay n_3 broadcasts a *DATA* message indicating the greedy mode $RM = 0$ with the local maximum information (M', σ_m) to discover the greedy candidates. The neighbor n_4 sends a *RESPONSE* message with its position N_4 closer to D than M'. Then, the current relay n_3 sends a *SELECT* message to the neighbor n_4 indicating the greedy mode $RM = 0$. So, the packet is able to continue in greedy mode $RM = 0$ through the node n_4 until reaching to the destination d. This example shows how the alternative operation of EGLE is able

to exit from false void areas.

4.3.5 Delay Function for Greedy and Alternative Modes

This section provides a delay function to reduce the transmission overhead in the beaconless forwarding for greedy and alternative modes of EGLE. For the beaconless forwarding, the number of responses may be high in dense networks. Moreover a *DATA* transmission is enough to discover greedy and alternative neighbors at the same time. In the forwarding scheme the current relay i broadcasts a *DATA* message to discover its neighbors j. The *DATA* message includes the estimated position information of the current relay i and the destination d. Before replying, every neighbor j delays its *RESPONSE* message according to its goodness as next relay. The neighbor having the best goodness must transmit the *RESPONSE* message first. The rest of neighbors overhearing that first *RESPONSE* message must cancel their *RESPONSE* messages. The current relay i must wait until receiving only a *RESPONSE* message and selects the neighbor that replied first.

In the following, we design the delay assignment function to ensure that alternative neighbors wait more than all greedy neighbors. So, alternative mode is only used when greedy mode is not able to provide advance. As we mentioned above, greedy neighbors ($P_{ij} > 0$) set their waiting times according to their goodnesses to advance toward the destination. This goodness is based on two parameters: the number of the same data packet received $NumR \in [0..MaxR - 1]$ (i.e. number of forwarding participations) and the Maximum Expectation Progress $MEP \in [0..R]$. Thus, each greedy neighbor j with $NumR_j$ and MEP_{ij} determines its waiting time (T_{ij}), according the following equation:

$$T_{ij} = (T_G/MaxR) * (NumR_j + ((R - MEP_{ij})/R)) \tag{4.9}$$

T_G is a constant representing the interval reserved for greedy neighbors. T_G is divided by $MaxR$ to ensure that neighbors j with different $NumR_j$ will never wait the same time. Neighbors j with a lower $NumR_j$ obtain a smaller T_{ij}. So, the responses of neighbors j is ordered based on the number of times $NumR_j$ they

acted as forwarding candidates to avoid backward progress. Among neighbors with the least $NumR_j$, those with a higher MEP_{ij} obtain a lower T_{ij}. The goal is to assign the less waiting time to neighbors with the least $NumR_j$ and higher MEP_{ij}.

Alternative neighbors ($-\sigma_{ij} < P_{ij} < 0$) set their waiting times according to their goodness to minimize negative progress. Each alternative neighbor j with ($P_{ij} \in [-\sigma_{ij}..0]$) determines its waiting time (T_{ij}), according the following equation:

$$T_{ij} = T_G + (T_A * (-P_{ij}/\sigma_{ij}) \tag{4.10}$$

T_A is a constant representing the interval reserved for all alternative neighbors. Neighbors with a lower P_{ij} obtain a smaller T_{ij}. The goal is to assign the less waiting time to neighbors minimizing the increment of the distance toward the destination. The minimum delay time of alternative neighbors is established by T_G. So, T_G ensures that alternative neighbors wait more than greedy neighbors.

When the current relay i broadcasts a *DATA* message, but receives no *RESPONSE* messages from greedy neighbors, then the packet enters in alternative mode for the local maximum $i = m$ with the local maximum position $I' = M'$ and associated error $\sigma_i = \sigma_m$. In alternative mode $RM = 1$, the forwarding process is limited to the alternative region A of the local maximum m. Therefore, the local maximum information (M', σ_m) is included in the *DATA* and *SELECT* messages. And, only 2-hop-greedy neighbors ($P_{mj} > 0$) and alternative neighbors ($-\sigma_{mj} < P_{mj} < 0$) of the local maximum m compete to send their *RESPONSE* messages. Both 2-hop-greedy and alternative neighbors j set their waiting times T_{mj} according to their values of $NumR_j$, MEP_{mj} and P_{mj}. When the packet reaches a 2-hop-greedy neighbor ($P_{mj} > 0$), then the packet is routed again in greedy mode $RM = 0$.

4.3.6 Broadcast Dissemination for Delivery to the Destination

Considering location errors, several studies [13] show that in greedy routing most of packet losses occur in the destination range. Because the data packet

advances till reaching an estimated position where the destination node is not really located. Only when the destination is located in the trajectory of the packet, the packet is delivered successfully. To address this issue, we designing a solution that looks for the destination in an area around its estimated position. In the literature, geocasting algorithms have been proposed to disseminate data packets to some destinations in a determined geographic area. However, geocasting protocols employ face routing to border the specific area which is inefficient and ineffective in location error scenarios [147]. In those cases, we first determine the specific area where the destination node is really located to utilize a broadcast dissemination technique in order to obtain a low overhead and a high delivery ratio. We describe our proposal below.

When the current relay i is within the radio range of the destination d. As we showed in section 4.2, even if the estimated distance between i and d is lower than the radio range R (denoted as $dist(I', D') < R$), the delivery may fail. The delivery failure is produced because the destination d may be outside of its estimated radio range due to the location error σ_{id}. As we demonstrated before, σ_{id} represents the maximum distance difference between the real distance $dist(I, D)$ and the estimated one $dist(I', D')$. And the maximum estimated distance between the destination's position D' and every neighbor's position J' is denoted as $MaxDist_{jd} = R + \sigma_{jd}$. For this reason, we define an estimated delivery area E centered in the destination's position D' with radio $MaxDist_{jd}$ where d must be really located.

In the estimated destination area E, we apply a counter-based broadcast scheme to guarantee the packet delivery. The main idea is exploiting the wireless medium to propagate the data packet in the area E. All nodes inside E are candidates to transmit the packet. So, the size of E and the density of nodes determine the number of possible transmissions. To reduce the transmission overhead, nodes inside E wait a random time before they decide whether they transmit the packet or not. They forward the packet when they received the packet less than a maximum number of transmissions times, denoted as $MaxT$. For our purposes we set $MaxT = 2$.

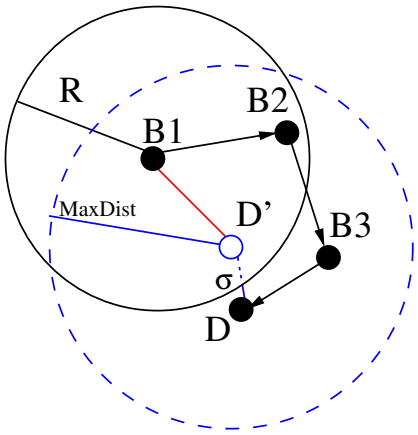


Figure 4.10: An example of broadcast strategy to reach the destination node in its real location.

To implement the broadcast scheme, the current relay i transmits the *DATA* message in broadcast mode with the routing mode $RM = 2$. When a node j located inside E ($dist(J', D') < MaxDist_{jd}$) receives the *DATA* message with $RM = 2$ for the first time, then it sets a random timer. During its waiting time j counts the number of the same *DATA* messages received. When the timer is fired, j transmits the *DATA* message if its counter is lower than $MaxT$. Otherwise, j drops the packet.

Additionally, we reduce the broadcasting overhead by sending a cancellation from the destination d. When the destination d receives the data packet, it broadcasts immediately the packet to cancel the propagation in its coverage area. So, each node d receiving the cancellation drops the data packet. In this way, we address the most important cause of packet losses in networks with location errors. Moreover, our adaptation of counter-based broadcast is efficient to avoid unnecessary transmissions.

Fig 4.10 shows an example of the estimated destination area E where the data packet is propagated until finding the destination d in its real position D. The current relay $b1$ inside the destination radio range (i.e. $dist(B1', D') < R$) sends the *DATA* message in greedy mode $RM = 0$ and receives no responses from d

or closer neighbors toward D'. Then, the relay $b1$ considers a void area in the destination radio range and becomes a local maximum. In that case, the relay $b1$ transmits the *DATA* message indicating the broadcast mode $RM = 2$. The *DATA* message is propagated by nodes $b2$ and $b3$ located at $B2$ and $B3$ until the destination d hears the *DATA* message in its realistic coverage area.

4.4 Simulation and Testbed Results

This section evaluates EGLE comparing with three relevant geographic protocols: BOSS, GRS and MER. As we saw in Section 3.3, BOSS provides the most efficient and effective beaconless scheme for WSNs with realistic radio communications. GRS [112] is a beacon-based protocol with the original greedy strategy based on selecting the next hop that maximize its advance toward the destination. MER [142] is a beacon-based protocol with the best performance of greedy routing with location errors because it is based on a probabilistic function considering such errors. Our EGLE solution is based on the BOSS forwarding scheme and provides three routing modes to improve progressively the performance of the protocol. To measure the effect of the different enhancements proposed in EGLE over the overall performance, we evaluate three versions of the protocol increasing the modes used (Greedy, Alternative, Broadcast): EGLE-G, EGLE-GA, EGLE-GAB.

As commented above, we implement the routing protocols in TinyOS [16] operating system and perform the evaluation using the TOSSIM [17] simulator which scales to thousands of nodes and facilitates the development of network applications.

To evaluate the protocols, we provide two type of experimental studies in a simulator and a real testbed. The first analysis was done to check the reliability and scalability of these protocols in the presence of location errors in ideal communication networks with thousands of nodes. The second study was performed in order to validate the robustness and reliability of the protocols in a real network with location errors where there also are wireless interferences,

irregular radio ranges and obstacles.

4.4.1 Performance Metrics

To evaluate the performance of the protocols, we considered the following metrics.

- Packet Delivery Ratio. This metric shows the reliability of the protocols. It determines the percentage of packets that successfully reach the destination node. This is the most important performance metric in scenarios with location errors.
- Percentage of Lost Packets. This metric accounts for the percentage of lost packets due to local maxima and delivery failures. In the presence of location errors, this indicates the main causes of packets losses due to void areas in greedy routing.
- Percentage of Local Maxima from Backward Progress. This metric calculates from all the local maxima reached by the protocol, the percentage of those which is explicitly caused by backward progress. It evaluates how much each protocol is affected by backward progress.
- Number of Backward Progress. This metric indicates the total number of forwarded packets generating backward progress. This determines the number of erroneous next-hop selections increasing the distance to the destination.
- Total Transmissions per Delivery. This metric measures the total number of packets transmitted per destination reached during the routing process from the source to the destination. It also accounts for transmissions of packets reardless of whether they were delivered to the destination or not.
- Total Packet Forwardings per Delivery. This metric estimates the total number of forwardings per destination reached during the routing process. It considers also the forwardings performed for packet that may get lost in their path to the destination.

4.4.2 Simulation Evaluation

In the simulation evaluation we consider networks without void areas where packet losses in greedy routing are only produced by location errors. The protocols are executed on the TOSSIM [17] simulator using a UDG model of the wireless communications. Where there are perfect links between a sender and a receiver whose distance is less than the radio range $R = 150$. To avoid void areas, we distribute nodes in the network by means of an hexagonal tessellation technique [148]. The idea is to divide the network area into regions with a regular tessellation which make up of congruent regular hexagons polygons where nodes are located randomly. So, a node in one region can reach any other node in a neighboring region guaranteeing the greedy strategy in all directions. The reason for these settings is to focus the evaluation in networks with location errors and avoid any possible influence of other issues.

The simulated network is a 2000x2000m^2 area with 900 nodes and a mean density of 15 neighbors per node. To emulate localization systems [99], we model location errors as a Gaussian Distribution with zero mean and a deviation from 5% to 100% of the radio range. We have considered 14 different deviation errors to represent a wide spectrum of realistic scenarios. Also, we consider that the packet source employs a location service mechanism to determine the inaccurate location information of the destination [5]. For each scenario, 100 random sources transmit a data packet to a destination which is always located in the center of the network. And the results are the average over a total number of 50 simulations that are enough to achieve a small 95% confidence interval.

Regarding the configuration of the algorithms, GRS and MER use a beacon period of 4 seconds. On the other hand, BOSS and EGLE use a 3-way handshake scheme and are configured with a greedy delay time $T_G = 300ms$. Moreover, EGLE has two additional modes and needs the following parameters: an alternative delay time $T_A = 300ms$, the maximum number of receptions $MaxR = 2$ and the maximum number of transmitted broadcasts $MaxT = 2$.

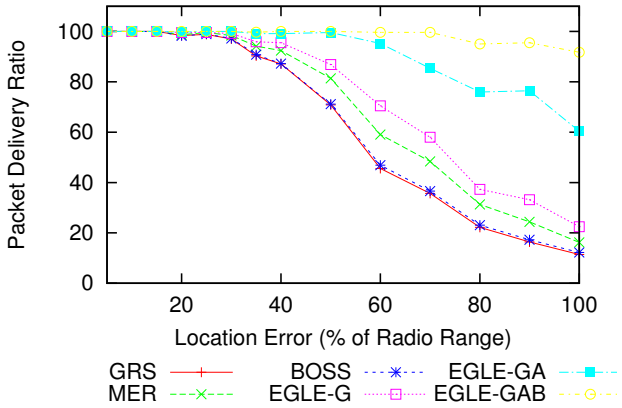


Figure 4.11: Packet Delivery Ratio

4.4.3 Analysis of Simulation Results

The main goal of EGLE is to mitigate the effects of location errors achieving a high packet delivery ratio. Fig 4.11 compares the packet delivery ratio for previous protocols. We can see that the three versions of EGLE clearly outperform GRS, BOSS and MER regardless of the deviation error. As our studies showed, when the location error is higher the probability of packets looses due to false void areas is also higher. In scenarios with large location errors GRS and BOSS provide a very low delivery ratio because they have been designed for scenarios with perfect positions. MER has a little bit better delivery ratio because MER considers location errors in its probabilistic routing decisions. However, the figure shows that EGLE's greedy, alternative and broadcast modes are able to mitigate progressively the effects of location errors. So, EGLE achieves over 90% of delivery ratio even with a 100% location error.

To analyze in more detail the delivery ratio of these protocols, Fig 4.12 shows the percentage of packet losses grouped by two causes: local maximums and delivery failures. Clearly, the three versions of EGLE exhibit a lower number of lost packets than GRS, BOSS and MER in all simulated scenarios. Concretely, GRS, BOSS and MER have many delivery failures in the destination radio range

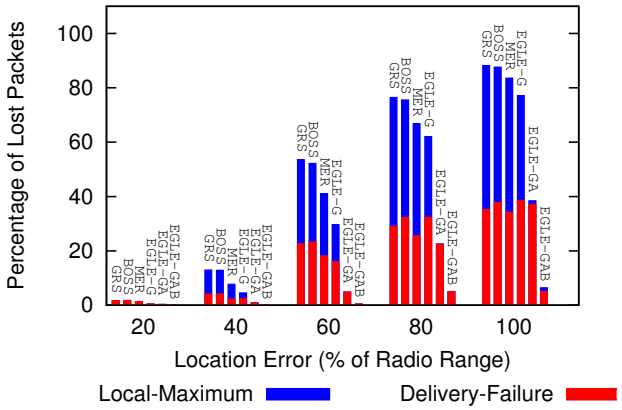


Figure 4.12: Percentage of Lost Packets

because they assume perfect knowledge of the destination position. Moreover GRS and BOSS are prone to reach local maxima because their next-hop selection functions maximize the advance neglecting the inaccuracy of nodes positions. The probabilistic function of MER avoids some local maxima by incorporating location errors to prioritize nodes whose positions are likely inside the greedy area preventing backward progress and excessive distance. However, MER does not consider the situations where local maximums may come from backward progress owing to nodes participating in the packet forwarding several times before (see Fig 4.13). Unlike MER, EGLE's greedy mode provides better results by a greedy heuristic that penalizes nodes with excessive distance and also penalizes nodes taking part in the packet forwarding few hops before. Moreover we can see that EGLE's alternative mode is able to exit from local maximums using their discarded neighbors. Finally EGLE's broadcast mode avoids almost all delivery failures by means of disseminating the packet in a limited area around the destination position.

To assess the good behavior of EGLE's greedy heuristic, we analyze in detail the number of local maximums reached from backward progress which is shown in Fig 4.13. GRS and BOSS have the lower number of local maxima from

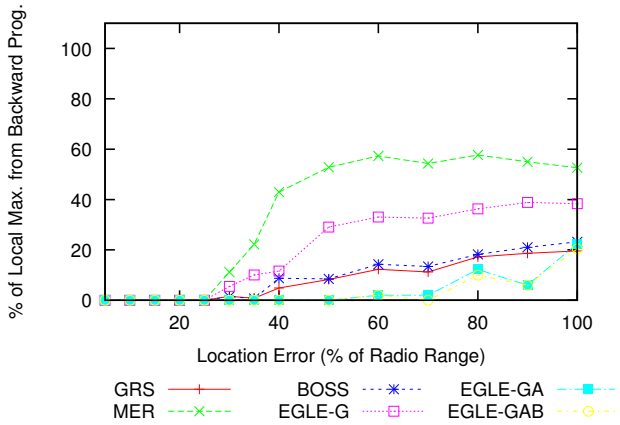


Figure 4.13: Percentage of Local Maxima from Backward Progress

backward progress since that their greedy functions maximize the advance and frequently select nodes with excessive distance. MER has the higher amount of local maximums in those cases because it neglects nodes participating as forwarding candidates several times before. The results show the unreliability of MER to prevent backward progress in networks with high location errors. This confirms that MER tends to choose nodes whose positions are near to the center of the radio range and behaves badly in scenarios with higher location errors than its threshold (31.5%), as we shown in Section 4.3.3. Unlike MER, EGLE's greedy heuristic reduces a 15% the number of local maxima in all simulated scenarios with more 40% of location errors.

As expected, all protocols experience the lowest performance in scenarios with high location error. The reason is that geographic protocols are based on position of neighbors to select the next relay reducing the destination distance. Also, a higher location error means a higher probability of suboptimal selections. This means that the current relay does not choose really the closest neighbor to the destination. And even the selected neighbor may be really farther to the destination than the current relay generating backward progress as confirmed in Fig 4.14.

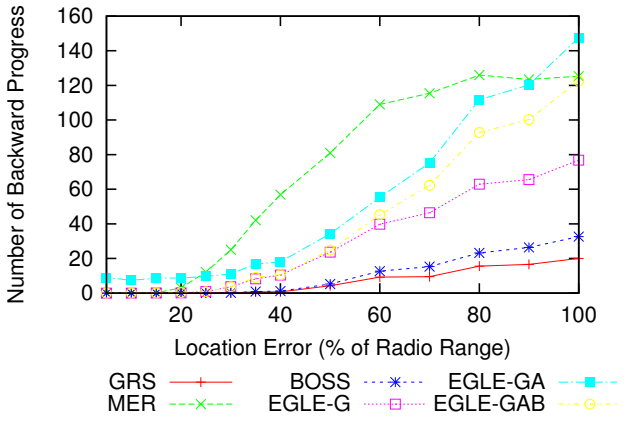


Figure 4.14: Number of Backward Progress

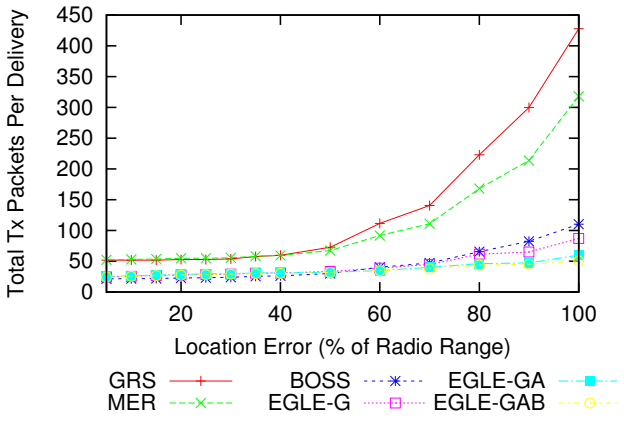


Figure 4.15: Total Transmissions per Delivery

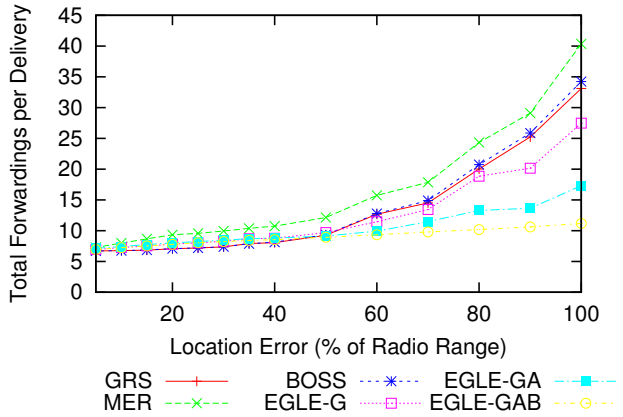


Figure 4.16: Total Packet Forwardings per Delivery

Regarding the overall number of transmitted packets per delivery, Fig 4.15 shows that, in addition to achieving a higher efficiency, EGLE's modes also have a lower transmission overhead than GRS and MER in all scenarios. The main reason is that the beacon-less nature of EGLE scales with the number of nodes in the network. EGLE avoids periodic beacon transmissions especially for the nodes not taking part in the routing process. Also, EGLE's delay function avoids unnecessary transmissions from neighbors being forwarding candidates. And EGLE's alternative and broadcast modes provide few more overhead because they use reduced areas around local maximums and the destination node. All versions of EGLE improve the performance of BOSS owning to their good balances between a high delivery ratio and only a little more overhead.

To analyze in detail the degradation of the path used for each protocol to reach the destination, Fig 4.16 shows the total number of forwardings per delivery. In that case, MER obtains even worse results than GRS and BOSS because MER does not work property for large location errors due to its threshold at 31.5%. Unlike MER, EGLE's greedy operation obtains a good performance due to the combination of its two objective functions to penalize separately neighbors with excessive distance and previous forwarding candidates. Moreover, the results

prove that the design of EGLE adapts perfectly to large location-error scenarios getting similar results without location errors.

4.4.4 Testbed Evaluation

To test the performance of EGLE in realistic wireless networks, we used the testbed shown in Section 3.3.4. This testbed network consists of a 75x40 m area in the Computer Faculty at the University of Murcia where 35 nodes were deployed with a mean density of 8 neighbors. Nodes were well-distributed to guarantee that greedy routing is always possible. According to all wireless links among neighbors, we determined that the mean radio range was 45 meters. This network represents a typical WSNs where collisions, interferences, irregular radio ranges and unidirectional links are frequent.

Each experiment comprises 5 random sources that transmit a data packet to 10 random destinations. The time between data packets from sources has been fixed to 5 seconds for guaranteeing no influence from previous packets transmitted in the network. The size of the data packet is 120 bytes considering the headers of MAC and network layers. Moreover, location errors are modeled as a Gaussian Distribution with zero mean and a deviation from 5 to 100% of the mean radio range. We have considered 6 different deviation errors to represent a wide spectrum of realistic scenarios: 5, 20, 40, 60, 80 and 100. For each scenario, nodes are pre-configured with the same inaccurate positions and the same set of sources and destinations in order to compare fairly all algorithms. The results are the average over a total number of 50 simulations to achieve a sufficiently small 95% confidence interval.

With regard to the protocols configuration, BOSS and EGLE utilize a beaconless forwarding scheme that needs a greedy delay time $T_G = 300ms$. Moreover EGLE has two extra modes and requires the following parameters: an alternative delay time $T_A = 300ms$, the maximum number of receptions $MaxR = 2$ and the maximum number of transmitted broadcasts $MaxT = 2$. As most beaconing protocols, GRS and MER employ a beacon period of 4 seconds during the simulation time of 250 seconds. To make a fair comparison, the four

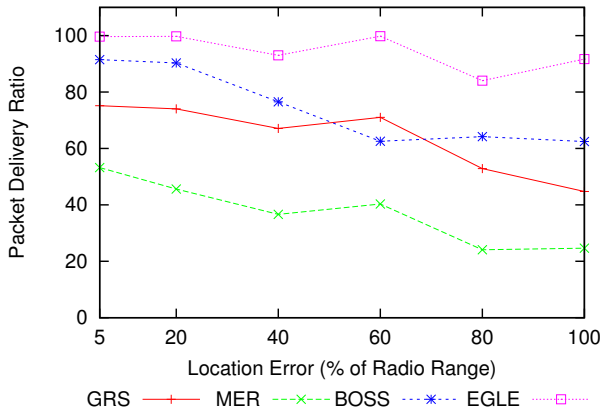


Figure 4.17: Packet Delivery Ratio

protocols provide acknowledgement mechanisms for avoiding packets losses due to communication errors. Concretely, we modify GRS and MER design to include a DATA/ACK mechanism being repeated up to 5 times.

4.4.5 Analysis of Testbed Results

Note that the design of EGLE employs the BOSS forwarding scheme to support realistic communication errors as well as three operation modes to deal with location errors in order to achieve a high delivery ratio. Fig 4.17 compares the packet delivery ratio of each protocol. Clearly, EGLE outperforms all protocols regardless of the location error. As our studies showed, the increment of the location error increases the probability of dropped packets in greedy routing due to void areas. GRS and BOSS have a low packet delivery ratio because their designs neglect the inaccuracy of nodes positions. Although MER incorporates the location errors in its probabilistic selection, it has the worse performance. The main reason is that the MER probabilistic function penalizes farther neighbors to avoid excessive distance, but does not consider neighbors participating as forwarding candidates various times before. In particular the MER probabilistic function is not able to select neighbors in the center of forwarding areas in sparse

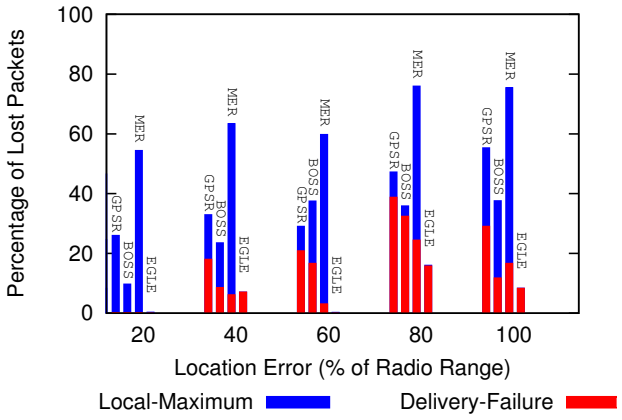


Figure 4.18: Percentage of Lost Packets

networks with a little mean density equal to 8. The penalization of neighbors for their proximity to the current relay is not able to prevent backward progress , and the penalization of farther nodes reduces the advance toward destinations which this means more number of forwardings. These two issues make that most neighbors are penalized and cause more probability to reach local maxima from backward progress. Unlike MER, the three operation modes of EGLE are able to mitigate the effects of location errors achieving over 90% of delivery ratio even with 100% of location errors.

Now, we study in detail the main causes of packet losses for the protocols in scenarios with inaccurate positions. Fig 4.18 shows the percentage of lost packets for every protocol grouped by two causes: local maximums and delivery failures. As abovementioned, EGLE provides a lower number of lost packets than GRS, BOSS and MER. We can see that GRS and MER experience many local maximums in scenarios with few location errors. The reason is that although nodes are well distributed to guarantee the greedy routing, the error-prone nature of wireless communications increases the probability of void areas due to interferences and collisions of periodic beacon messages used to discover neighbors. Unlike GRS and MER, BOSS and EGLE utilize a beaconless scheme

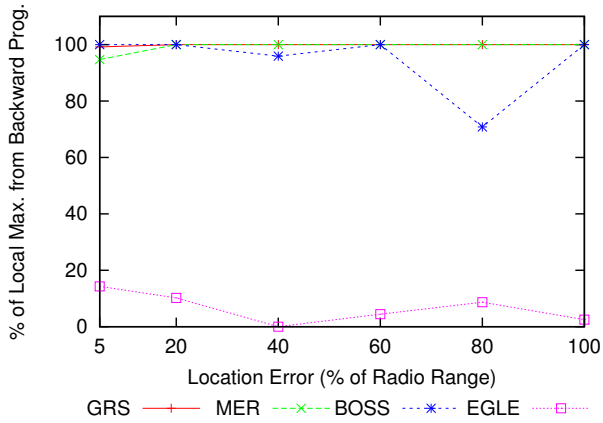


Figure 4.19: Percentage of Local Maxima from Backward Progress

providing an improved neighborhood discovery strategy in realistic wireless communications. As it happened in the simulations, GRS, MER and BOSS have many delivery failures with large location errors since these protocols assume the perfect knowledge of the position of the destination. Moreover in sparse networks there are less forwarding candidates, and the inaccuracy of nodes positions generates more void areas. However, EGLE proposes two operation modes to deal with void areas where there are neighbors enabling the advance toward the destination. The figure shows the efficiency of the proposed alternative and broadcast modes to exit from local maximums and enhance the packet delivery to the destination.

Regarding void areas, Fig 4.19 shows the percentage of each protocol to reach local maximums from backward progress. In that case, GRS, BOSS and MER reach almost 100% of local maximums due to the next hop selection of neighbors which are really located farther the destination node than the current relay. Note that the pre-configured radio range $R = 45$ of the protocols limits the next hop selection inside the forwarding area to avoid transmission failures what discards farther neighbors with good links. Moreover the results confirm that the probability selection of MER penalizing neighbors for their proximity

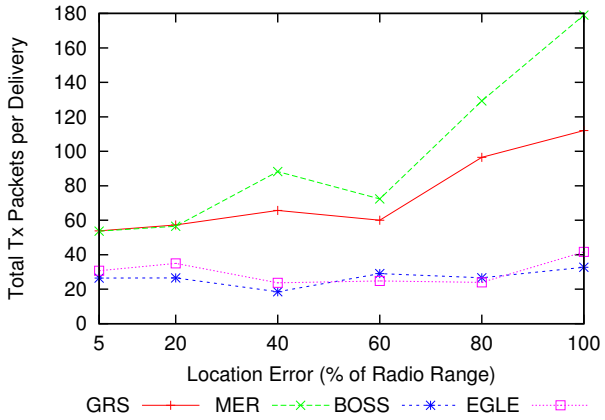


Figure 4.20: Total Transmissions per Delivery

to the current relay is not able to prevent backward progress. Nevertheless, EGLE provides an enhanced greedy mode that determines the maximum distance $MaxDist$ to each location error where there are neighbors which can act as relay candidates. In addition the greedy heuristic penalizes the number of times that neighbors take part in the forwarding process. For these two reasons, EGLE's greedy mode is able to avoid almost all backward forwardings even with high location errors.

To analyze the overhead of each protocol to reach the destination, Fig 4.20 shows the number of transmitted packets per delivery. We can see that BOSS and EGLE exhibit a much lower transmission overhead than GRS and MER. Because in GRS and MER the beaconing scheme produces a large number of periodic messages even when nodes are not participating in any data packet forwarding. Moreover MER has a higher number of transmissions than GRS since that its probabilistic function penalizes the progress to the destination requiring more hops to deliver the data packet. The results show the good balance of EGLE between a high packet delivery ratio and a small transmission overhead due to the efficient alternative and broadcast modes which succeed in reducing the effects of location errors.

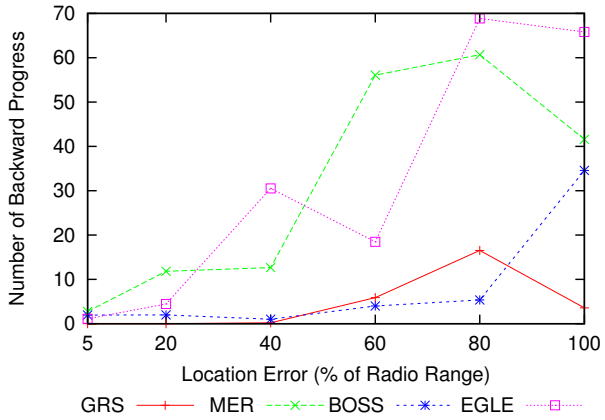


Figure 4.21: Total Backward Progress

Regarding the path used by the protocols, Fig 4.21 shows the total number of backward progress during the routing process. We can see that EGLE and MER have higher amount of backward progress than GRS and BOSS in all tested scenarios. The reason is that the greedy selection of GRS and BOSS maximizes the progress toward the destination employing the closest neighbors to the destination in each hop. EGLE and MER penalize nodes whose position is near to the radio range which may be local maximums.

We study the total number of forwardings to reach the destination in Fig 4.22. We can see that MER experiences the highest forwarding overhead to deliver the packet even more than GRS and BOSS. As abovementioned, the reason is that the greedy function of MER is not able to work fine with large location errors due to its threshold at 31.5%. In addition, EGLE and BOSS outperforms GRS since their beaconless scheme is based on a reactive neighborhood discovery that avoids collisions and beacon losses in realistic wireless communications. This confirms that EGLE's alternative and broadcast modes provide a reduced overhead like the most efficient geographic protocol (BOSS) in terms of transmissions and forwardings achieving a high delivery ratio above 90% even with location errors of 100%.

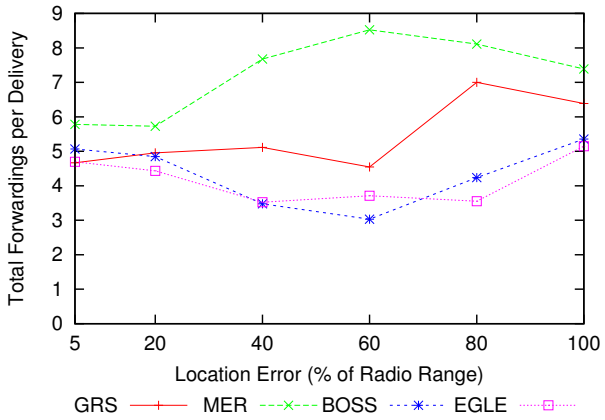


Figure 4.22: Total Packet Forwardings per Delivery

4.5 Conclusions

In this chapter, we propose and evaluate an Effective Greedy routing protocol supporting Location Errors (EGLE) for WSNs. EGLE's main goal is to mitigate the effects of location errors achieving high delivery ratio with little control overhead.

According existing studies, we classify the error factors reducing the performance of greedy routing in networks with inaccurate positions. Also, we analyze in detail the effects of location errors in greedy routing protocols. Our study concludes that void areas appear due to the positions inaccuracy which is the main cause of packets losses in greedy routing.

To reduce the packets losses, EGLE provides three routing modes to deal with void areas generated by location errors. Firstly, its greedy routing penalizes too distant neighbors and previous relay candidates to prevent reaching void areas. Secondly, its alternative routing mode employs discarded neighbors to exit from a local maximum in order to advance through the void area. Third, its broadcast routing propagates the data packet in a reduced area around the destination estimated position to ensure the delivery in its real one.

On the other hand, EGLE utilizes the beaconless forwarding of BOSS

for realistic conditions of wireless communications. During the greedy and alternative modes, EGLE combines the BOSS beaconless forwarding and a new delay assignment function to guarantee hop-to-hop delivery and reduce the transmission overhead.

Simulated and testbed experiments have been made to evaluate the performance of the EGLE proposal against three relevant geographic protocols (BOSS, GRS and MER). The simulated evaluation determines that EGLE's three routing modes enhance progressively the performance of the protocol that outperforms BOSS, GRS and MER not only in terms of delivery ratio, but also in terms of the number of transmissions and forwardings to reach the destination. Moreover, the practical analysis also confirms that EGLE succeeds in achieving a much higher delivery ratio above the 90% with a little more number of transmissions even in a real testbed with 100% of location errors. In conclusion, all results show that EGLE provides a good balance among high reliability and efficiency.

In addition to the perfect behavior of wireless communications and location systems, most geographic routing algorithms assume that WSNs are deployed in secure areas. However, open wireless medium is prone to be attacked by malicious nodes which want to avoid the communication among nodes. To defense against malicious nodes, Chapter 5 deals with study and developing a self-secure geographic routing protocol.

Chapter 5

Geographic Routing in Networks with Malicious Nodes

Previous chapters propose two efficient and reliable beaconless geographic routing protocols for WSNs with realistic wireless communications and inaccurate positions. Beaconless protocols are based on a reactive forwarding scheme to discover neighbors and a delay function to determine the next hop selection. In beaconless routing, nodes whose positions provide more advance toward the destination wait less time and become to next hops to forward the data packet.

However, geographic protocols have not been designed to work in unsafe environments where an attacker can exploit the open wireless medium to severely affect communications [14, 15]. For instance, a sinkhole attacker acts as a neighbor in the forwarding process without any delay time to intercept and drop data packets. Moreover a sybil attacker can use multiple identities associated with different positions to pretend being the best forwarder for all forwarding operations that happen within its radio range. Once an attacker gets selected as the next forwarder it can drop the packet. To provide security in geographic routing, complex mechanisms have been proposed such as trust-management, location-verification and cryptography. Trust management is based on neighbors' reputation to penalize the anomalous behavior of attackers. Location verification [149, 150] employs ultrasonic hardware to measure the

distance between nodes to verify the reported location. Using cryptography [76, 75] nodes share symmetric keys to authenticate and encrypt the packets. However, these mechanisms require extra hardware and energy. In addition, cryptography is insufficient against insider-attackers that are able to get private keys and compromise routing protocols [14]. For these reasons, we discuss in this chapter that unlike existing solutions which are very complex and require a los of extra overhead, a simple routing protocol is enough in most scenarios to offer a reliable routing solution even in the presence of insider attacks.

In this chapter we propose a Self-Protected Beaconless Geographic Routing protocol (SBGR) for WSNs. We have made a detailed study to determine the effects of insider attacks (sinkhole and sybil) in the performance of beaconless routing. SBGR provides two simple forwarding modes based on distributed competition and limited flooding of notification messages. First, nodes forward data packets by competing distributively in order to prevent sinkhole attackers intercepting and dropping packets. Second, nodes detecting a sybil attacker flood data packets in a reduced area to guarantee that data packets can continue being forwarded toward the destination.

The performance of SBGR is evaluated against the unique secure protocol for beaconless geographic routing (SIGF [151]) employing extensive simulations and realistic experiments. Extensive simulations assess the efficiency of both protocols at increasing number of insider attackers. To validate their robustness and reliability we compare the protocols in a realistic testbed described in Chapter 3.

In this chapter, the main goals are:

- Describing all attacks against routing algorithms in WSNs and existing secure solutions presented in the literature.
- Analyzing in detail the operation of insider attacks (sinkhole and sybil) and their effects in the performance of geographic routing algorithms.
- Designing a simple geographic routing algorithm that considers the constrained resources of sensor nodes and defends from insider-attackers.

- Evaluating the performance of SBGR with SIGF (the unique known secure beaconless algorithm) employing simulation and testbed experiments.

5.1 Related Work: Geographic Routing with Malicious Nodes

Several studies [14, 152]) have described and modeled attacks for routing protocols in WSNs. WSNs are prone to many routing attacks taking advantage of wireless communications in a shared medium. Given that WSNs are formed by devices with power and computation, secure and efficient routing mechanisms in terms of radio transmissions and complexity are required.

Geographic protocols are efficient routing solutions for WSNs, but they do not consider possible attacks. In GR, routing decisions are based on neighborhood location information which is a critical factor. Attacks using false location information can compromise routing decisions. An attacker can simulate a false location closer toward the destination than any candidate neighbor to get selected as next-hop and drop all traffic. Moreover an attacker can inject false positions of its neighbors to generate routing loops or void areas.

To prevent false location information, various verification mechanisms have been proposed [153, 149, 154, 155]. Sastry et al.[153] provide a location verification employing ultrasonic hardware to measure the distance between nodes and check the trusted location. SeRLoc [149] is a range independent localization algorithm based on beacons transmitted from fixed nodes acting as trusted reference points. Capkun et al. [154] present a range dependent positioning system based on distance bounding and verifiable multi-lateration. Zang et al. [155] assess the trusted location of nodes through triangulation and RF-based fingerprinting methods.

Trust management is one of the most popular techniques to secure routing in WSNs. The basic idea is to overhear the transmissions from neighbors and keep reputation information according their behaviors. Nodes with low reputation are penalized to select legitimate neighbors to forward the packets. Boukerche and Li [156] propose a localized trust and reputation management that reduces the energy and bandwidth consumption.

In addition, there are cryptographic techniques to provide authentication

and encryption at the link-layer for routing protocols in WSNs. SPINS [76] and TinySec[75] use symmetric cryptography or hashing to maintain routing or discovering new routes, respectively.

However, existing solutions do not adapt to the limited resources of sensor nodes. These mechanisms require specific hardware, maintaining reputation tables and complex cryptographic operations. Moreover, cryptography is insufficient to provide a reasonable protection against insider attacks [14]. The reason is that insider attackers being capable of getting valid cryptographic keys can participate in routing process and send valid data packets. To achieve protection against insider attacks, routing protocols need efficient and effective mechanisms guaranteeing the routing process.

5.1.1 Routing Attacks in Beaconless Greedy Strategy

Several works [14] study attacks against routing protocols in WSNs. The common goal of these attacks is to degrade the routing performance in terms of latency, number of transmissions, delivery ratio, etc. These routing attacks are summarized as follows.

- State corruption: As every node acts as a router, an attacker might provide false routing information.
- HELLO flooding: Some protocols use HELLO packets (beacons) to generate routing tables based on 1-hop neighbors. An attacker with powerful transmitter can be seen as a legitimate neighbor by many nodes.
- Wormholes: In networks where packets can be routed through tunnels between two distant nodes. An attacker simulates to tunnel packets received in one part of the network over a low latency link, but the packets are dropped.
- Denial of service: An attacker could record and replay legitimate messages. So it generates duplicated packets and unnecessary traffic.

- Sinkhole: An attacker tries to cause that in its coverage area all traffic is forwarded through itself. So it drops the packets and creates an artificial hole in the network.
- Sybil: An attacker employs multiple identities to pretend being the best forwarder for all packets transmitted in his coverage area. The attacker's goal is catching and dropping the packets.

Before describing the effects of these routing attacks, we give some background about beaconless greedy routing and the two main forwarding schemes employed.

As shown in Chapter 3, beaconless greedy routing uses a reactive neighborhood discovery to avoid the overhead caused by periodic beacon transmissions. That is, the current node routing the data packet broadcasts a message, and neighbors reply reactively as forwarding candidates. Neighbors' replies are usually ordered according to a delay function, and the first transmission cancels the remaining replies. In greedy mode the routing metric is the distance toward the destination, so that the neighbor providing the largest advance replies first.

Existing beaconless routing protocols differ on small details regarding on how the forwarding scheme is performed. For instance, some protocols broadcast a control message to discover neighbors, other protocols broadcast the data packet first, etc. We can group existing beaconless solutions into two different approaches based on their forwarding schemes: three-way handshake (i.e. IGF [110]) and distributed forwarding (i.e. BLR [157]).

Three-way handshake consists of the exchange of three messages (query-response-select) between the current forwarder and its neighbors. The current forwarder broadcasts a query message to discover its neighbors, and so they set their delay times according their positions. After waiting for its computed delay, a neighbor reports back its position and identifier in a response message. The closest neighbor waits less time, so its response cancels the ones from other neighbors. Finally, the forwarder explicitly sends the data packet to the neighbor responding first.

Distributed forwarding employs an unique broadcast message to discover and forward the data packet. Neighbors receiving the data packet utilize their delay times to compete in a distributed way for becoming the next hop. Finally, the neighbor closest to the destination broadcasts first the data packet canceling the rest of candidates. Note that this also initiates again the forwarding process.

Now, we classify the routing attackers into three categories according to their effects in beaconless routing protocols.

1. Compromising attackers introduce false routing information used by nodes in the forwarding process such as state corruption, hello flood and wormholes. A malicious node is able to create routing loops, simulate void areas, generate false error messages, etc. These attackers do not affect beaconless geographic protocols because such protocols do not maintain any routing information. Beaconless protocols perform a reactive next hop selection that requires storing no state.

2. Denial-of-service attackers generate useless traffic to waste the constrained resources of sensor nodes (i.e. energy and bandwidth) and jeopardize the overall routing performance. These attackers record and replay legitimate messages to reduce the performance of beaconless protocols. Attackers replace the identity of the sender and generate unlimited duplicated packets. To avoid denial-of-service, beaconless protocols require sophisticated cryptographic mechanisms to limit the forwarding of data packets from senders whose identifies cannot be validated.

3. Interception attackers try to catch all traffic in their coverage area to suppress it (i.e. sinkhole and sybil). They can act as legitimate neighbors taking part in the routing process in order to get selected as next hop. A sinkhole attacker exploits the reactive neighborhood discovery scheme of beaconless algorithms and replies without any delay time to cancel the remaining neighbors in order to become the unique forwarding candidate. A sybil attacker exploits the next-hop selection based on positions and uses multiple identities associated with different positions to pretend being the

closest forwarding candidate to the destination. For these reasons, sinkhole and sybil are the worst enemy for beaconless geographic routing protocols, and thus we analyze them in detail below.

5.1.2 Existing Beaconless Protocols Supporting Routing Attacks.

In the literature there are few secure routing protocols for geographic routing in WSNs. Concretely, we have found only one beaconless geographic solution considering the routing attacks in its design presented below.

SIGF: Secure Implicit Geographic Forwarding

SIGF (Secure IGF) [151] is the unique secure proposal for beaconless geographic routing. The authors proposed a family of secure routing protocols based on the IGF [110] protocol whose beaconless forwarding prevents the attacks compromising routing information. These secure protocols provide novel mechanisms to protect against the sinkhole, sybil and denial-of-service attacks discussed above. The proposed mechanisms increase progressively the complexity and security of the three SIGF protocols: SIGF-0, SIGF-1 and SIGF-2.

SIGF-0 is a stateless protocol that maintains no routing information and provides only probabilistic defenses against sinkhole attacks. The main SIGF-0 defense is a longer collection window for waiting several responses of neighbors to avoid that the first transmission of an attackers can cancel the remaining of forwarding candidates. Moreover, SIGF-0 uses a small forwarding area (60^o) to ensure the overhearing among neighbors.

SIGF-1 keeps a local history and a trust management to protect against sinkhole and sybil attacks. The local history is learned from interactions of neighbors during the data packets forwardings to obtain four reputation parameters:

- Forwarding success ratio (α) measures the reliable to forward the packet.

- Forwarding fairness ratio (β) determines the next hop distribution among forwarding candidates.
- Position consistency (γ) calculates the position variance.
- Forwarding performance (ζ) estimates the average delay during the packet forwarding.

Based on these parameters, nodes employ the trust management to discard those neighbors whose reputation is less than a specific threshold ($R_{threshold}$).

Finally, SIGF-2 utilizes keys and sequence numbers shared among neighbors to enable cryptographic guarantees during the routing process. This solution requires a pairwise shared state within the neighborhood to provide guarantees for authenticity, confidentiality, integrity and freshness.

However, the SIGF-2 protocol is an insufficient solution for insider attacks because they are able to get valid cryptographic keys and participate in the routing process as legitimate nodes [14]. For this reason, we consider the SIGF-1 version as the most efficient and reliable proposal of the SIGF family for the rest of this chapter. Moreover, we focus our efforts in dealing with sinkhole and sybil attacks without requiring cryptographic mechanisms. In the next section, we analyze sinkhole and sybil attacks and how they affect the performance of beaconless geographic routing.

5.2 Analysis of Insider Attacks for Beaconless Forwarding Schemes

This section analyzes the effects of insider attacks in beaconless geographic routing protocols. We present a detailed study of the operation of the sinkhole and sybil attacks which severely degrade the performance of such protocols.

Sinkhole and sybil attackers exploit the forwarding process of beaconless protocols to become the next forwarder and intercept all traffic. This is done in the reactive discovery scheme by replying before other neighbors to cancel

their replies. When a sinkhole attacker gets the data packet, then it just drops the packet. Additionally a sybil attacker can create multiple fake identities with positions closer than other neighbors toward the destination for the same purpose of intercepting traffic.

Both attacks behave slightly different depending on the forwarding scheme (i.e. three-way-handshake or distributed forwarding) employed by beaconless geographic protocols. For the case of protocols such as IGF [110] which uses a three-way-handshake, we describe below existing solutions provided by SIGF [151]. Moreover, we also explain our proposed solutions for protocols based on distributed forwarding such as BLR [157].

5.2.1 Sinkhole Attack in IGF and BLR

In beaconless forwarding, a current sender located at position S and denoted as s holds a data packet addressed to a destination d located at position D outside of its radio range R. The sender s has a set of neighbors n_i being located at position N_i, denoted as $N = \{n_1, n_2, \ldots n_k\}$. To select the next hop, s broadcasts a message indicating its position S and the destination position D. All neighbors located closer to D than S participate in the forwarding process. In insecure environments, an attacker m located at position M inside the radio range R can pretend to be the best forwarding candidate in order to become the next hop.

Fig 5.1 shows the operation of a sinkhole attack in IGF's handshake-based next-hop selection. In the example, the current forwarder s located at position S broadcasts a short message (*RTS*) including the positions S and D to ask for available neighbors. Neighbors n_1 and n_2 wait a delay time according to their positions, before they send their response messages (*CTS*). A sinkhole attacker m sends immediately its response (*CTS*) without any delay to pretend to be the neighbor providing the largest advance towards d. Thus, m cancels responses from n_1 and n_2, and s ends up sending the *DATA* message to m that drops it.

To avoid sinkhole attacks, SIGF proposes a contention window where s waits some time to receive more than one response. In this example, neighbors n_1 and n_2 do not cancel their responses even if some other neighbors answered first.

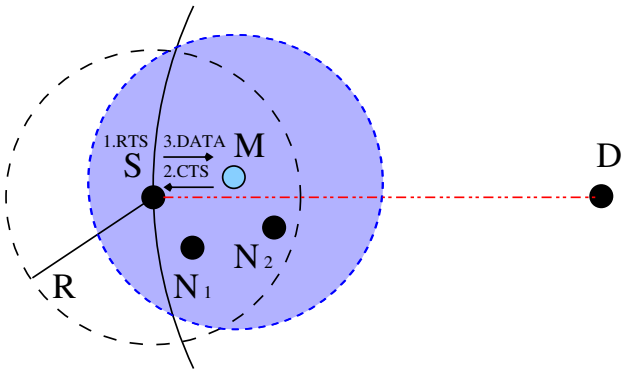


Figure 5.1: A sender s uses the IGF protocol to forward a packet toward d in presence of a sinkhole attacker m

Among all responses received, s selects the neighbor n_2 whose position N_2 is the closest to D. SIGF also proposes a reputation mechanism to discard anomalous behavior as the sinkhole attacker m that drops all data packets.

Fig 5.2 shows the operation of a sinkhole attack in BLR's distributed next-hop selection. The current forwarder s broadcasts a unique *DATA* message including the data packet addressed to D and its position S to discover its neighbors (i.e. n_1 and n_2). Neighbors n_1 and n_2 compete in a distributed way delaying their forwarding of the data packet depending upon their advance towards d. A sinkhole attacker m can only cancel the forwarding of n_1 and n_2 by broadcasting the *DATA* message first. In this example, the forwarding performed by m cancels a neighbor n_2 located at position N_2 closer to D than M.

Given that the sinkhole node has to forward the *DATA* message to cancel the remaining forwarder candidates, the attack is not very severe as the packet is not dropped. The main reason is that the *DATA* message of m restarts the distributed forwarding process in its coverage area, and there may be closer neighbors to d than m (i.e. n_3) which facilitate the forwarding of the data packet toward d. Moreover, the canceled neighbor n_2 is able to easily check if its position N_2 is closer to D than M. In that case n_2 waits for its timer to expire before broadcasting its *DATA* message.

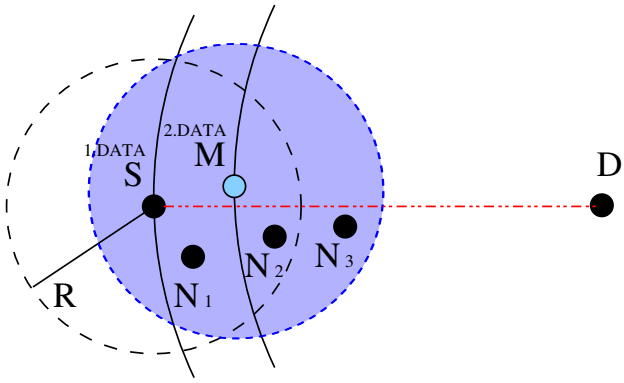


Figure 5.2: A sender s uses the BLR protocol to forward a packet toward d in presence of a sinkhole attacker m.

5.2.2 Sybil Attack in IGF and BLR

The most interesting insider attack in geographic routing is the sybil attack. This attack tries to overcome reputation mechanisms where neighbors overhear the forwardings and penalize attackers for their malicious behaviors (i.e. they drop data packets). A sybil attacker can create multiple virtual identities to always pretend to be the neighbor with the closest position toward the destination and become the next hop.

Fig 5.3 shows the operation of a sybil attack in IGF's next-hop selection. The current forwarder s at position S broadcasts a *RTS* message including the positions S and D to discover its neighbors (i.e. n_1 and n_2). A sybil attacker m receiving the *RTS* message creates an identity with a fake position M' closer to D than S. The attacker m can choose the false position M' as the closest position inside the radio range R of s. The goal is to ensure that its response cancels the responses of all neighbors n_1 and n_2 whose positions N_1 and N_2 are farther than M' from S to D. So, m replies first a *CTS* message including its false identity M', and the forwarder s sends the data packet to m that drops it.

SIGF proposes a reputation scheme to control the behavior of nodes and detect sinkhole and sybil attacks. Nodes overhear the transmission of their neighbors

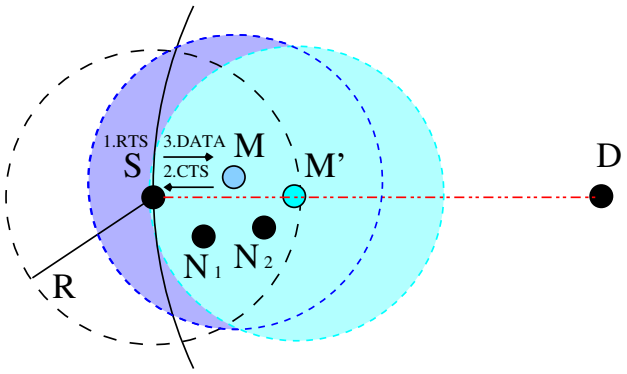


Figure 5.3: A sybil attacker m located at position M creates a false identity M' to become the next hop in the three-way handshake of s.

and keep a reputation table in terms of forwarding success, location consistency, average delay, etc. So, a node penalizes a sinkhole attacker for not forwarding *DATA* messages. A node also penalizes a sybil attacker for changing its position in CTS messages. However, the sybil attacker can create multiple identities to fool the reputation scheme. Thus, SIGF requires an additional cryptographic solution to verify the identities of nodes in the network. Even in that case, a sybil attacker may steal some identities from legitimate nodes.

Fig 5.4 shows the operation of a sybil attack in BLR's next-hop selection. The current forwarder s broadcasts a *DATA* message including the data packet addressed to D and its position S to discover its neighbors (i.e. n_1 and n_2) as forwarding candidates. Neighbors n_1 and n_2 compete in a distributed way delaying their forwardings according to their advance towards d. A sybil attacker m creates a virtual identity with the position M' which is the closest toward d inside the coverage area of s. So m employing its false identity broadcasts a *DATA* message to cancel the forwardings of n_1 and n_2. In this case, the false position used by m is not a big problem since n_3 receiving the *DATA* message provides advance and continues the forwarding. However, the attack can be more elaborated by m creating an additional new identity with a false position M'' which is the closest inside its radio range R. Thus, the forwarding with the identity

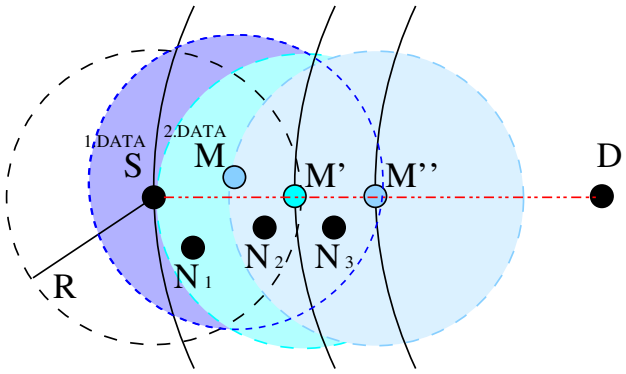


Figure 5.4: A sybil attacker m located at position M creates a false identify at M' to become the next hop in the distributed forwarding of s.

located at M'' cancels not only n_1 and n_2, but also n_3.

To deal with this attack we present below a detailed study which demonstrates that even in this highly elaborated strategy, the sybil attack can be detected by its neighbors. In the example, the neighbor n_1 located at position N_1 is farther to D than M, and n_1 detects the sybil attack by the reception of a *DATA* message from a false position (M'') outside of its radio range R.

5.2.3 Study of False Positions in Sybil Attacks

This subsection studies the relation between the detection of its false position and the cancellation of forwarding candidates in the presence of a sybil attack. To do that, we divide the coverage area of a sybil attack into three subareas according to the false position used (see Fig 5.5):

- Detecting subarea consists of neighbors that are able to detect the false position outside their radio ranges (pink region).
- Canceling subarea contains neighbors that are canceled by the false position, but are really located closer than the sybil attacker to the destination (green region).

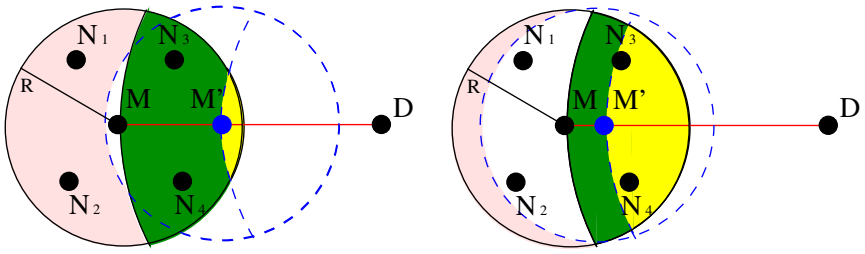
- Forwarding subarea comprises neighbors located closer to the destination than the false position and can continue to forward the data packet (yellow region).

To determine the relation between the defined subareas, we study two opposite cases where a sybil attacker employs two different false positions to cancel distributed BLR's next-hop selection as shown in Fig 5.5. As shown, the sybil attacker m located at M receives a *DATA* message addressed to the destination d at D.

In the first case (see Fig 5.5(a)), m sends immediately a *DATA* message with a position M' near to the radio range R to cancel all closer neighbors n_i. We can see that the forwarding subarea is very small, and the false position M' permits to cancel all relay candidates (i.e. n_3 and n_4). However, the detecting subarea is big, and the message supposedly sent from the false position M' is received by some nodes (i.e. n_1 and n_2) that are outside of their radio range R and should not have received it. Therefore, a false position close to the radio range R provides a higher probability of the attacker being selected as next hop , but provides a higher probability of detection by remaining neighbors.

In the second case (see Fig 5.5(b)), m sends immediately a *DATA* message with a position M' near to its real position M to avoid the detection of farther neighbors n_1 and n_2. We can see that the detecting subarea is very small, and the false position M' is not detected by farther neighbors (i.e. n_1 and n_2) inside of their radio range R. However, the forwarding subarea is very big, and the false position M' is not able to cancel any forwarder candidates (i.e. n_3 and n_4). Thus, a false position near to the real position provides a lower probability of detection, but enables a higher probability of the packet being forwarded by a legitimate node.

In conclusion, a closer false position to d cancels more forwarding candidates, but the detection probability is higher. This relation is considered in the design of our self-protected routing proposal to overcome the issue of sybil attacks.



(a) First Case: the sybil attacker m creates a virtual node with a closer position M' to its radio range R.

(b) Second Case: the sybil attacker m creates a virtual node with a closer position M' to its real position M.

Figure 5.5: Studying the coverage area of a sybil attacker.

5.3 Self-Protected Beaconless Geographic Routing (SBGR)

This section presents our Self-protected Beaconless Geographic Routing protocol (SBGR) that is able to deal with routing attacks in WSNs. SBGR is based on the Beacon-Less Routing protocol (BLR [157]) proposed by Heissenbüttel et al. In BLR neighbors compete distributively to forward the data packet first and cancel the possible forwardings of the remaining neighbors. As discussed above, this distributed forwarding provides an stateless scheme to defend against attacks compromising routing information. Moreover the effects of insider attacks (i.e. sinkhole and sybil) can be prevented easily by the simple mechanisms proposed below.

As in most secure geographic routing protocols, SBGR assumes that nodes know their stationary position pre-configured in the deployment or using precise hardware (i.e. GPS [98]). There are two system-wide parameters, which are known by all the nodes, $MaxDelay$ determines the maximum delay to compete in the distributed forwarding and a maximum transmission radius R. Each packet source employs some secure location service to determine the location of a destination [158]. The greedy path between the source and the destination is almost guaranteed in high density networks where the packet always can advance

using closer neighboring nodes. Moreover, nodes check the integrity of each packet transmitted by overhearing during the forwarding process. To do that, each packet includes the identifier and sequence number of the source, the identifier and position of the destination and the data payload. Finally, malicious nodes have similar features of the normal nodes in the network, so they have the equivalent transmission power.

SBGR is a reliable routing solution that improves the efficiency of the distributed forwarding and includes effective defenses against sinkhole and sybil attacks. The idea of SBGR is to ensure a correct packet forwarding by overhearing neighboring nodes in the network. We exploit the open wireless medium to control the packet transmission from the source until reaching the destination. In each hop, all neighbors receiving the packet must guarantee its advance toward the destination. To do that, SBGR provides two different forwarding schemes based on two types of messages: *DATA* and *NOTIFY*. SBGR forwards distributively the data packet in a *DATA* message which advances toward the destination avoiding sinkhole attackers. When nodes detect a sybil attacker, they apply a limited flooding of *NOTIFY* messages containing the *DATA* message to inform the attack and continue the distributed forwarding outside its radio range. However, BLR's distributed forwarding of *DATA* message is prone to generate duplicated packets in realistic scenarios with communication errors, as we demonstrated in Chapter 3. Thus, SBGR also incorporates a simple mechanism to avoid propagating duplicated packets in the case in which multiple neighbors forward the same packet. The next subsections illustrate how our proposed protocol deals with sinkhole and sybil attacks as well as with duplicated messages.

5.3.1 Dealing with Sinkhole Attacks

Here, we describe a simple defense of SBGR against sinkhole attacks. As we aforementioned, sinkhole attacks exploit the delay time in beaconless forwarding of *DATA* messages. So, a sinkhole attacker receiving a *DATA* message replies immediately to cancel the remaining forwarding candidates.

To avoid sinkhole attacks, we include a condition in the forwarding process of

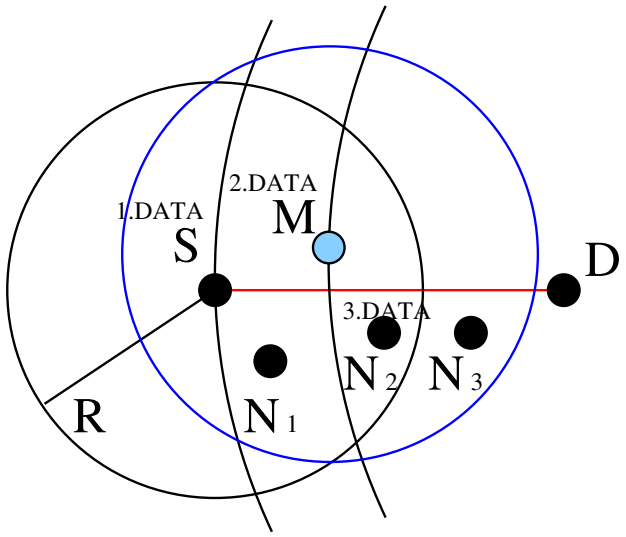


Figure 5.6: The defense of SBGR against sinkhole attacks.

DATA messages. Each *DATA* message contains the data packet and the positions of the sender and destination. In the *DATA* processing, neighbors verify simply if they are located closer to the destination than the sender. This situation only happens for attackers replying before their waiting time. That is, neighbors only cancel their forwardings if they receive a *DATA* message from a sender closer to the destination.

For instance, in Fig 5.6 the current sender s located at position S forwards a *DATA* message toward the destination d located at position D. The sinkhole attacker m broadcasts immediately a *DATA* message with its positions M to cancel all neighbors n_1 and n_2. In SBGR, n_2 receiving the message realizes that its position N_2 is closer to D than the position of the attacker (M). Thus, n_2 ignores the message of m and forwards its *DATA* message after its timer expires.

5.3.2 Dealing with Duplicated Packets

As Chapter 3 showed, the distributed forwarding proposed by BLR is prone to generate duplicated packets in realistic wireless communications. The reason is

that the next-hop selection is fully distributed, and communication errors avoid overhearing between neighbors which transmit duplicates. So, the propagation of duplicates reduces severely the performance of BLR in terms of transmission overhead and energy consumption.

Moreover duplicated packets are generated as a side-effect of the proposed mechanism for dealing with sinkhole attackers. In the previous example, the sinkhole m forwards first the *DATA* message, and a closer neighbor n_2 ignoring it generates a duplicated *DATA* message. Thus, the neighbor n_3 receives two *DATA* message from different senders m and n_2. Therefore, our idea of ignoring sinkhole forwardings produces duplicated packets.

To address these two issues, we provide a duplicate avoidance condition without extra overhead in the forwarding process of *DATA* messages. In each hop, nodes store temporally the received *DATA* message with the sender position. For each *DATA* duplicate received, a node checks if the current sender provides less advance than the stored sender. In that case, the node ignores the duplicated message. Otherwise, the node resets its waiting timer and stores the current sender as $BestRelay$. The key is that nodes receiving multiple duplicates only consider the *DATA* message from the $BestRelay$ of the previous forwarding. So, an unique *DATA* message is processed in each hop.

Fig 5.7 shows a common example of duplicates appearing for sinkhole attackers. A current sender s forwards a *DATA* message toward the destination d. The sinkhole m broadcasts immediately the *DATA* message to cancel the forwarding from all legitimate neighbors n_1 and n_2. However, n_2 does not cancel its transmission and ignores the broadcast of s because the position N_2 is closer to D than M. Moreover, the node n_3 receiving the broadcast stores the message with $BestRelay = M$ and sets its waiting timer. When the timer of n_2 expires it broadcasts its *DATA* message. In that case, n_3 receives a duplicated *DATA* message from n_2 of a previous forwarding. Then n_3 realizes that its stored $BestRelay = M$ is farther from D than the current sender N_2. So, n_3 resets its timer for the *DATA* message and updates its $BestRelay = N_2$. Thus all nodes would behave as if a single *DATA* message had been forwarded in the previous

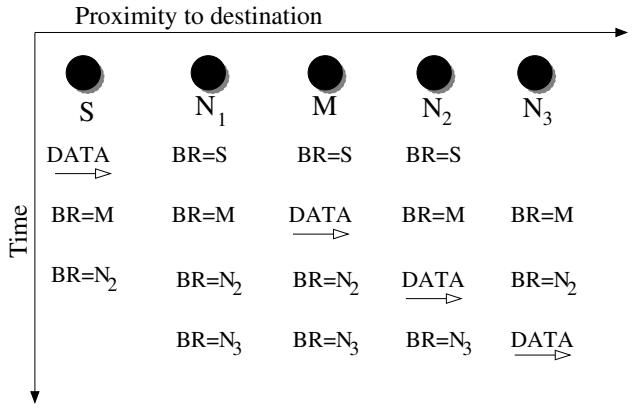


Figure 5.7: The duplicate avoidance of SBGR using $BestRelay(BR)$ with a sinkhole attacker m.

step (the one from n_2).

5.3.3 Dealing with Sybil Attacks

As we have explained before, sybil attackers are more difficult to deal with because they create multiple identities with positions closer toward the destination than the real one. This attack exploits geographic routing decisions where packets advance through the closest neighbor in each hop. Receiving a *DATA* forwarding, a sybil attacker replies immediately using a position closer than its neighbors in order to cancel their transmissions and avoid the packet propagation.

To guarantee the packet propagation in presence of a sybil attacker, SBGR provides a constrained flooding in the sybil radio range. The presence of a sybil attack is detected by neighbors overhearing a *DATA* message with a false position outside their radio coverage which we proved in Section 5.2.3. These neighbors start the flooding of a *NOTIFY* message including the *DATA* message in the coverage area of the sybil attacker. The goal is to make that legitimate nodes located outside the sybil radio range receive the *NOTIFY* message and continue

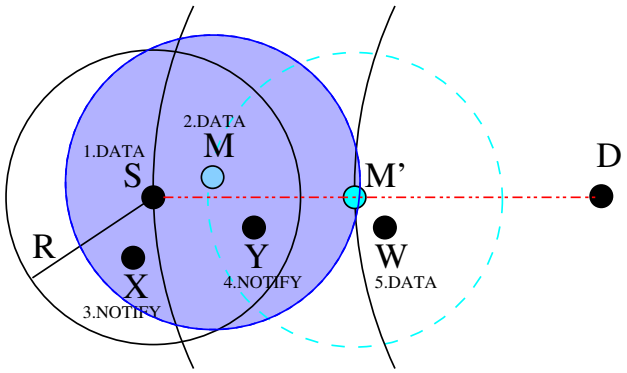


Figure 5.8: SBGR using the notification flooding to defend against sybil attacks.

the distributed forwarding of the *DATA* message. To do that, SBGR develops a notification scheme divided into 3 steps: detection, diffusion and recovery. Next, we introduce its operation by means of an example (see Fig 5.8).

1. **Detection.** A current forwarder s located at position S sends a *DATA* message with its identifier and position as well as including the data packet addressed to the destination d located at position D. Then a sybil attacker m with real position M replies immediately a *DATA* message using a virtual identifier m' with a false position M'. When a node x receives the *DATA* message from a position M' which is outside its coverage area then x detects a possible sybil attack. Thus, the node x creates a *NOTIFY* message to start the constrained flooding.

   The *NOTIFY* message contains the same information of the original *DATA* message and also the identifier and position of the detector x. Moreover it includes a new field called $ReliableRelay$ which the detector x fills in with its $BestRelay$ stored. This field is used to inform the position of the closest forwarder to the destination from which the node x has received the *DATA* message (excluding the detected sybil attack). In the example, the field $ReliableRelay$ is the position S which belongs to s.

2. **Diffusion.** To exit the sybil radio range, each node that receives the *NOTIFY*

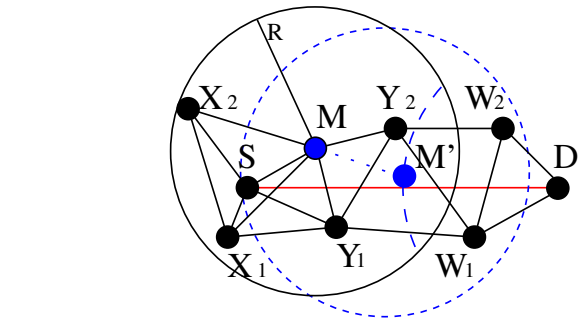
message and is closer to D than $ReliableRelay$ (e.g. y located at Y), broadcasts the same *NOTIFY* message with its identifier and position. The main idea is to flood the *NOTIFY* message through neighbors y canceled previously by the sybil attacker m. Those neighbors y really providing more advance than the sybil m may reach nodes located closer to D than M that are outside the radio range of the attacker m.

3. **Recovery.** The recovery happens when the *NOTIFY* message reaches a node w that never participated in the forwarding of this *DATA* message. This means that the node w located at position W is outside the coverage area of the sybil attacker m, and the *DATA* message can be forwarded in the normal distributed mode.

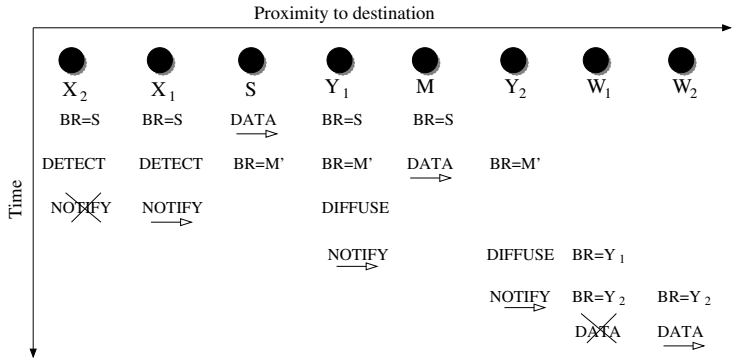
Design of the Notify Scheme for Reducing Overhead

The notification scheme is designed to minimize the transmissions of *NOTIFY* messages and guarantee that the packet advances to the destination. The flooding of *NOTIFY* messages is limited to only those nodes that really provide advance toward the destination. Moreover the *NOTIFY* requires no propagation when the sybil attacker is not able to cancel all forwarder candidates competing to send the *DATA* message in the distributed way. To achieve these two objectives, we describe in detail the nodes behavior in the three notification steps.

To reduce the number of transmissions in the detection, only nodes that have already received the *DATA* message before are candidates to make *NOTIFY* messages. Moreover, those nodes detecting a sybil attack apply the distributed forwarding mechanism to generate the first *NOTIFY* message. In the distributed way, they delay their *NOTIFY* transmissions based on minimizing the distance toward their $BestRelay$. Those nodes located closer to their $BestRelay$ have coverage areas providing more probability to reach forwarding candidates canceled by the sybil attacker. So, the closest node broadcasts first the *NOTIFY* message which cancels other *NOTIFY* messages from other nodes detecting the attack. The



(a) Network Topology



(b) Nodes Operation

Figure 5.9: The notification scheme of SBGR against a sybil attacker m located at M employing a virtual identity m' at M'.

The *NOTIFY* diffusion is limited to only nodes whose *DATA* forwarding was canceled previously. Because nodes that are located outside the coverage area of the sybil m do not receive the *DATA* transmission of m and continue the distributed forwarding. Moreover nodes that receive the *NOTIFY* message and are located farther to d than $ReliableRelay$ do not participate in the diffusion.

In addition, the recovery step avoids a large number of duplicated *DATA* messages. To achieve that only nodes closer than $ReliableRelay$ extract the *DATA* message and compete distributively in the forwarding. In distributed way, nodes delay their *DATA* transmissions by maximizing the advance from the *NOTIFY* sender.

Fig 5.9 shows an example of our notification scheme with limited overhead in

the presence of sybil attackers. A current forwarder s sends a *DATA* message which is received by its neighbors x_1, x_2, y_2 and the sybil attacker m. The attacker m broadcasts immediately a *DATA* message indicating its virtual identity m' at position M' to cancel all forwarding candidates y_1 and y_2. However, neighbors x_1 and x_2 detecting the false position M' create a *NOTIFY* message with their $BestRelay = S$. Based on the distributed forwarding according to the $BestRelay = S$, the timer of x_1 expires first and x_1 broadcasts a *NOTIFY* message with the $ReliableRelay = S$ canceling the transmission of x_2. When y_1 and y_2 receive the *NOTIFY* message then they broadcast the *NOTIFY* to inform about the sybil attack. Finally, the *NOTIFY* message reaches nodes W_1 and W_2 which compete distributively to forward first the *DATA* message toward d.

Implementation of the Notify Scheme against Sybil Attacks

The design of the notification scheme considers the cases where a sybil attacker generates *NOTIFY* messages for reducing the performance of SBGR. First, an attacker tries to avoid the diffusion of the *NOTIFY* message for forwarding candidates canceled previously. Second, an attacker tries to increase the transmission overhead of SBGR producing high congestion and energy consumption. Below these two situations are described in detail, and how the notification scheme is implemented to support limited sensor nodes.

In the first case, a sybil attacker transmits a false *NOTIFY* message to cancel the flooding of the real *NOTIFY* message. The sybil attacker may exploit that the first *NOTIFY* message cancels the remaining detecting nodes, and the included $ReliableRelay$ limits the diffusion step. This situation is shown in Fig 5.10 where the current forwarder s sends a *DATA* message addressed to a destination d, and a sybil attacker m replies immediately a *DATA* message employing a false position M' which is detected by a node x. Then the sybil attacker m broadcasts immediately a *NOTIFY* message with a fake $ReliableRelay = M''$ that is closer to D than X and Y to cancel the *NOTIFY* messages from nodes x and y.

To avoid the notification cancellation, an extra condition is included in the detection step. Every node detecting the sybil attack only cancels its transmission

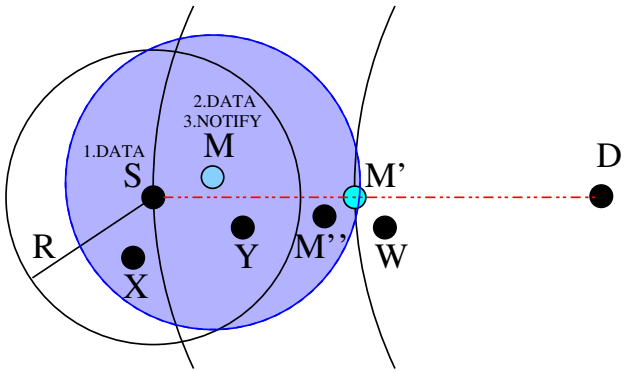


Figure 5.10: A sybil attacker employs the notification flooding to damage the protocol performance.

if the node receives a *NOTIFY* message containing the same data packet and $ReliableRelay$. In the example, node x ignores the *NOTIFY* message from the sybil m with a different $ReliableRelay = M''$ and broadcasts its *NOTIFY* message indicating its real $ReliableRelay = S$. In the diffusion phase, the node y receiving the *NOTIFY* message with the fake $ReliableRelay = M''$ ignores the message because its position Y is farther to D than M''. Nevertheless, when the node y receives the *NOTIFY* message with the real $ReliableRelay = S$ then y broadcasts the *NOTIFY* message because its position Y provides advance toward D.

In the second case, a sybil attacker saves a legitimate *NOTIFY* message and rebroadcasts it for generating duplicated floodings and the congestion in its coverage area. This attack is specially dangerous in dense networks where *NOTIFY* duplicates may produce a broadcast storm causing many communication failures such as collisions and interferences.

To avoid this unnecessary overhead in resource constrained sensor networks, we include a second condition in the notification scheme that nodes in the network can only send a *NOTIFY* message per each data packet. To implement that, sensor nodes save only the identification tuple of each *NOTIFY* message transmitted consisting of the identifier and sequence number of the source node that generates

the data packet. Nodes store a temporal list of identification tuples if the list is full then it deletes the least recently used entry. So, they can discard *NOTIFY* messages duplicated by a sybil attacker using very little memory.

5.3.4 Algorithmic Description of the SBGR Operation

SBGR provides a simple self-protected beaconless design to deal with insider attacks. The SBGR protocol uses two different forwarding schemes: distributed and flooding. The distributed scheme requires only an unique *DATA* message per hop and avoids the interception of sinkhole attackers. The flooding scheme employs a limited broadcast of *NOTIFY* messages to propagate the *DATA* message beyond the influence area of sybil attackers. Both forwarding schemes are shown in Algorithm 1 and Algorithm 2. These algorithms describe the operation of SBGR in four parts: the distributed forwarding of *DATA* messages, the detection of a false position, the diffusion of a *NOTIFY* message and the recovery of a *DATA* message.

The distributed forwarding of the *DATA* message is illustrated in lines 1-9 of Algorithm 1. A node a located at position A receives a *DATA* message containing a data packet addressed to the destination d at position D from the node b at position B. If the node a has a position A closer to D than B and a saved $BestRelay$ farther from D than B, Then a participates in the forwarding. And the node a saves the sender B as $BestRelay$ and sets its waiting time based on its progress. If the node a receives no *DATA* messages with a closer sender than itself, a broadcasts the *DATA* message when its waiting timer expires.

The detection of a false position is presented in lines 10-18 of Algorithm 1. If a node a receives a *DATA* message from a node b located outside of its radio range, a did not broadcast its *NOTIFY* message previously , then a creates a *NOTIFY* message where the field $ReliableRelay$ equals to its $BestRelay$ and sets a waiting timer according to its distance to $BestRelay$. After its waiting timer expires, if a has received no *NOTIFY* messages for the same data packet, a broadcasts the *NOTIFY* message to inform the attack to neighboring nodes.

The diffusion of a *NOTIFY* message is shown in lines 1-4 of Algorithm 2.

Algorithm 1 ProcessData(DATA, A, B, D): The node a at A receives a $DATA$ message addressed to the destination d at D from the node b at B.

```
1: if dist(A, B) <= R then
2:    if (dist(A, D) < dist(B, D)) and (dist(B, D) < dist(bestRelay, D))
      then
3:       bestRelay ← B
4:       t ← delayTimer(A, B, D)
5:       wait(t)
6:       if IsNotReceived(DATA) then
7:          broadcast(DATA)
8:       end if
9:    end if
10: else
11:    if (isNotSent(NOTIFY)) then
12:       t ← notifyTimer(A, bestRelay)
13:       wait(t)
14:       if isNotReceived(NOTIFY) or isNotSame(DATA, ReliableRelay)
          then
15:          broadcast(NOTIFY)
16:       end if
17:    end if
18: end if
```

Algorithm 2 ProcessNotify(NOTIFY, A, B, D, ReliableRelay): The node a at A receives a $NOTIFY$ message with the field $ReliableRelay$ from the node b at B to the destination d at D.

1: **if** $isSaved(DATA)$ **then**
2: **if** $(dist(A, D) < dist(ReliableRelay, D))$ **and** $(isCancelledForwarding(DATA))$ **and** $(isNotSent(NOTIFY))$ **then**
3: $broadcast(NOTIFY)$
4: **end if**
5: **else**
6: $ProcessData(extract(NOTIFY), A, B, D)$
7: **end if**

A node a receives a *NOTIFY* message referring to a saved data packet. Node a broadcasts the *NOTIFY* message if its position A is closer to D than $ReliableRelay$, its *DATA* forwarding was canceled and a did not broadcast this *NOTIFY* message previously. Note that this imposed conditions limit the flooding to a few nodes around the sender and the attacker.

Finally, the recovery step is shown in lines 5-7 of Algorithm 2. A node a receives a *NOTIFY* message regarding a not stored data packet. In that case, the node a is outside the sybil radio range and extracts the *DATA* message to continue the normal forwarding using the distributed mode.

In conclusion, SBGR provides two efficient forwarding schemes to defend from insider attacks (i.e. sinkhole and sybil). These efficient schemes require only that nodes store temporarily a few messages received. Unlike previous protocols, SBGR avoids the overhead of keeping neighborhood reputation, complex cryptography or location verification. This efficiency makes SBGR as scalable as traditional geographic routing protocols which is the most important feature in WSNs.

5.4 Simulation and Testbed Evaluation

This section compares the performance of our proposal SBGR (Self-Protected Beaconless Geographic Routing) and SIGF [151] (Secure Implicit Geographic Forwarding) which is the unique known secure beaconless geographic routing protocol in the literature. Our evaluation shows the performance of both protocols in the presence of sinkhole and sybil attacks. For comparison, we consider insider attackers which are able to get cryptographic keys as shown several studies [14]. For this reason, we use the most secure version of SIGF which does not require complex cryptographic techniques. The SIGF-1 version posses a local history and a reputation scheme to protect against sinkhole and sybil attacks.

As the previous chapters, we develop the routing protocols in the TinyOS [16] operation system created by UC Berkeley. The protocols are implemented a component-oriented variant of C programming language, called NesC.

The configuration of SIGF is equal to the one used by its authors. That is, a 60^o forwarding area and a fixed collection window for 3 messages. Next-hop selection is done by reputation whose parameters are configured as $\alpha = 5/8, \beta = 1/8, \gamma = 1/8, \zeta = 1/8$ and $R_{threshold} = 0.45$, described in Section 5.1. For SBGR and SIGF we employ $MaxDelayTime = 300ms$.

We implement sinkhole and sybil attackers that behave exactly as in the models described in Section 5.1. That is, in SIGF and SBGR a sinkhole attacker replies immediately without delay. And a sybil attacker creates a new identity with the closest position inside the radio range of the current forwarder and replies immediately.

To evaluate both protocols with sinkhole and sybil attacks, we utilize two types of experiments: simulations and testbed networks. The simulation study is used to validate the scalability and efficiency of these protocols in networks with a large number of nodes. The testbed analysis is employed to assess the reliability and robustness of these protocols in a realistic network where there are also irregular radio ranges and obstacles.

5.4.1 Performance Metrics

In the performance evaluation of the protocols, we consider the following metrics:

- Packet Delivery Ratio. This metric shows the percentage of packets reaching the destination node. It determines the robustness of the protocols against attackers.
- Number of Packets per Hop. This metric accounts for the average number of packets transmitted by a sender and its neighbors in the data forwarding to the next hop. It measures the efficiency of the protocols in the next-hop selection process.
- Tx Packets per Delivery. This estimates the mean number of packets used in the routing process from a source until reaching a destination. It also determines the efficiency of each protocol.
- Number of Hops. This metric indicates the average length of routes from every source to the destination.
- Time per Hop. It measures the time needed for a sender and its neighbors during the data forwarding to the next hop.

5.4.2 Setting in the Simulation Evaluation

The simulated analysis focuses on the effects of insider attacks (i.e. sinkhole and sybil) in secure beaconless protocols for greedy routing. Thus, we distribute nodes uniformly to ensure that there are no void areas, and greedy routing is enough to deliver the packets to destinations. To avoid void areas, we distribute nodes in the network by means of an hexagonal tessellation technique [148]. The whole network area is divided into congruent hexagons regions where nodes are located randomly. So, a node in one region can reach any other node in a neighboring region guaranteeing that greedy routing can be applied in all directions.

Both SBGR and SIGF are executed in the TOSSIM [17] simulator. The simulation network is a 2000x2000m^2 area with a mean density of 20 neighbors

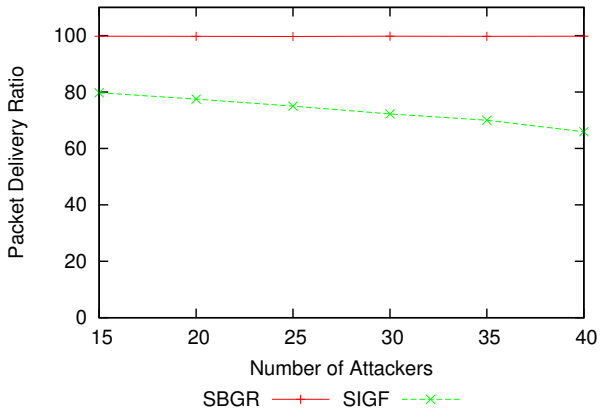


Figure 5.11: Packet Delivery Ratio

per node using a radio range $R = 150$. The number of attackers is increased from 15 to 40, and they are randomly located in the network. For each scenario, we simulate 100 random sources sending a 110 byte packet to a destination located at the middle of the network. The results are the average of 50 simulation runs in order to achieve a small 95% confidence interval.

5.4.3 Simulation Results with Sinkhole Attackers

These results measure the performance of SBGR and SIGF at increasing the number of sinkhole attackers. During the beaconless forwarding, a sinkhole attacker replies immediately without delay to become the next hop. To deal with sinkhole attacks, SIGF provides a contention window to receive several responses and a reputation scheme to penalize nodes with malicious behaviors. SBGR nodes check that the sender is really closer than themselves before they cancel their responses. Moreover nodes employ their stored $BestRelay$ to avoid propagating duplicated packets.

To study the robustness of the protocols to mitigate the effects of sinkhole attacks, Fig 5.11 compares their packet delivery ratio (PDR). SBGR clearly outperforms SIGF regardless of the number of attackers. The SIGF reputation

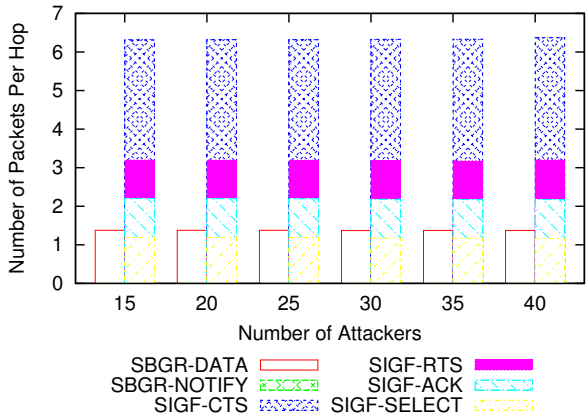


Figure 5.12: Number of Packets per Hop

needs several forwardings to detect a malicious node dropping the packets. At increasing number of attackers, SIGF decreases its PDR because in each hop the probability of selecting a sinkhole attacker increases. Unlike SIGF, the simple condition used by SBGR is very effective at dealing with sinkhole attackers, making that the PDR reaches a 100% in all tested scenarios.

Regarding the control overhead in the forwarding process, Fig 5.12 shows the number of transmissions per hop for each message type. As we can see, SBGR also has a lower number of transmissions per hop than SIGF. While the distributed forwarding of SBGR only requires one transmission per hop, the three-way handshake used by SIGF requires at least 4 transmissions: RTS, CTS, *DATA* and ACK. Moreover to avoid the selection of the first sinkhole response, SIGF employs a collection window needing at least 2 extra responses per hop.

To analyze the balance between efficiency and reliability of both protocols, Fig 5.13 shows the total number of transmissions per destination successfully reached. Again SBGR needs a lower total number of transmissions than SIGF in all the simulated scenarios. In SIGF the increment of attackers increases the number of transmissions per destination reached. The reason is that the number of failed deliveries increases making the packet forwardings of undelivered messages

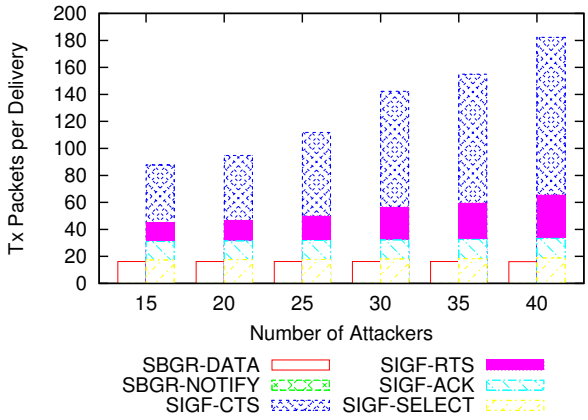


Figure 5.13: Tx Packets per Delivery

useless. While in SBGR, sinkhole attackers are ignored, but generate duplicated packets. The reason is that neighbors whose positions are closer than sinkhole attackers to the destination continue the distributed forwarding. However, the results show that SBGR provides a high efficiency because nodes are able to stop propagating duplicates from previous forwarders.

5.4.4 Simulation Results with Sybil Attackers

These experiments measure the performance of SBGR and SIGF at increasing number of sybil attackers. In each forwarding, a sybil attacker creates a false identity with the closest position inside the radio range of the current forwarder and replies immediately to cancel the remaining neighbors and become the next hop. Thus the performance degradation of SIGF is high because its reputation scheme is not able to detect attacks using multiple identities, as shown in Section 5.2. However in SBGR the notification scheme is able to detect false positions used by a sybil attacker and guarantee that the data packet advances using a limited flooding within its coverage area.

Regarding the reliability of the protocols to reach the destination, Fig 5.14 shows their packet delivery ratios. Clearly, SBGR outperforms SIGF in all tested

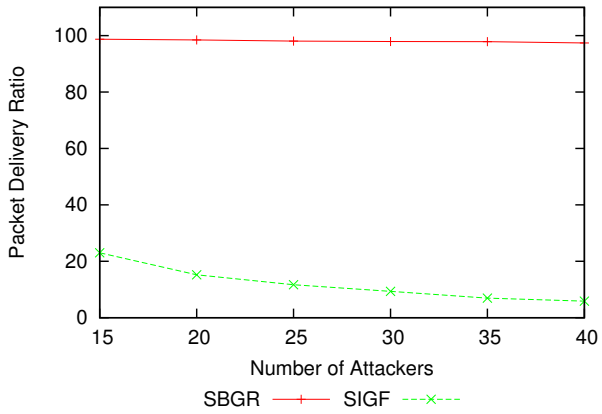


Figure 5.14: Packet Delivery Ratio

scenarios. As we mentioned, the main problem of SIGF is that its reputation schemes are ineffective to protect against sybil attackers that intercept and drop all packets in their radio ranges. The reason is that sybil attackers use different false identities in each forwarding with the closest position from the current sender to the destination. Thus, the reputation scheme is not able to find enough evidences about the malicious behavior of a false identity. In SIGF the packet is only delivered if there are no sybil attackers in the path from the source to the destination. At increasing number of attackers, the probability of reaching the destination is dramatically reduced. However the results shows that SBGR keeps nearly a 100% delivery ratio even with many sybil attackers demonstrating the reliability of the notification scheme.

To analyze the overhead required by SBGR to achieve such a high delivery ratio, Fig 5.15 presents the number of transmissions per hop. As expected SBGR increases moderately the number of transmissions, when the number of attackers increases. Because the flooding of notifications is designed to minimize the extra transmission overhead inside the radio range of sybil attackers. Thus in all simulated scenarios, the number of packets per hop never goes beyond 4 which is still below the number of messages required by SIGF.

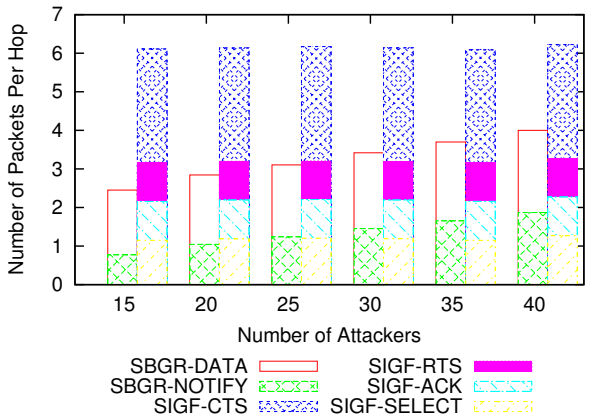


Figure 5.15: Number of Packets per Hop

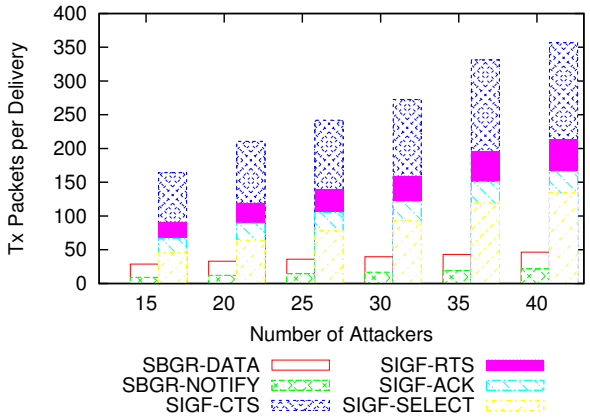


Figure 5.16: Tx Packets per Delivery

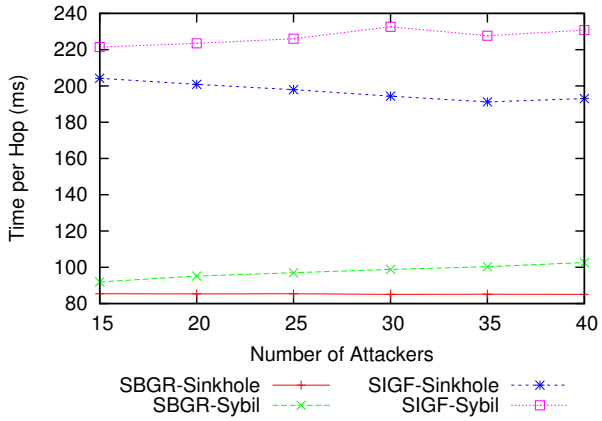


Figure 5.17: Time per Hop

To measure the balance between efficiency and reliability, Fig 5.16 shows the average number of transmissions per successful delivery to the destination. Although SBGR has a much higher delivery ratio than SIGF, SBGR requires a lower number of transmissions than SIGF. In SIGF the number of transmissions for reaching the destination increases quickly with the increment of sybil attackers, because the transmitted messages for undelivered data packets increases. However SBGR combines the distributed forwarding and the notification flooding to advance efficiently toward the destination and successfully avoid the packets dropped by sybil attacks. The results prove the good balance of the two routing modes of SBGR achieving an almost perfect delivery ratio with a moderate transmission overhead.

5.4.5 Simulation Results with Sinkhole and Sybil Attackers

This subsection shows the effects of sinkhole and sybil attacks in the protocols performance in terms of the hop-by-hop forwarding time and the average number of hops to reach the destination.

Fig 5.17 shows the time required by both protocols to forward the packet in each hop. The results confirm that SBGR requires a lower time than SIGF.

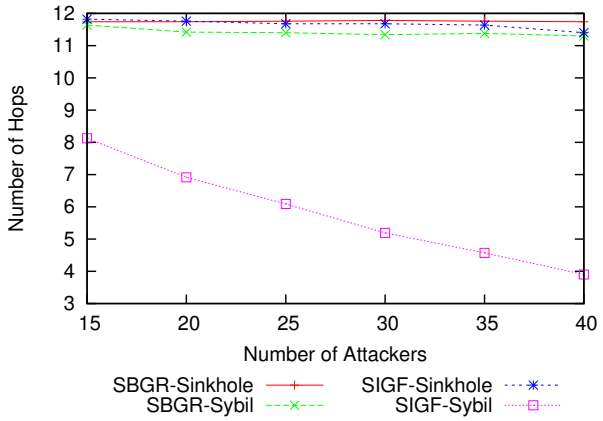


Figure 5.18: Number of Hops

Because SBGR uses a distributed forwarding based on neighbors competing to transmit first, while the SIGF handshake scheme delays the forwarding due to needing several responses to select explicitly the next hop. As we can see, sybil attacks make both protocols taking more time per hop. The reason is that in SBGR the notification scheme requires three steps for detecting the attacker, diffusing *NOTIFY* messages and recovering the distributed forwarding. However even with many sybil attacks SBGR still requires less time per hop than SIGF.

Fig 5.18 shows the mean number of hops required by both protocols for delivering packets. As expected, SBGR and SIGF need similar number of hops because they employ the same selection function minimizing the distance to the destination. The notable difference is the case of sybil attackers in which SIGF has a much lower number of hops than SBGR. However, this is not due to a good performance of SIGF, in reality its forwarding scheme is only able to deliver packets when no sybil attacker is present in the source-destination path. Thus, the only paths where SIGF is successful are very short.

5.4.6 Setting in the Testbed Evaluation

This subsection assesses the performance of SBGR in a realistic sensor network. As Chapters 3 and 4 show, we used a testbed network deployed in the Computer Science Faculty at the University of Murcia. This network consists of a 75x40m area where 35 motes were deployed with a mean density of 8 neighbors per node as shown in Section 3.3.4. We distributed uniformly nodes in this area in order to guarantee greedy routing. The mean radio range is 45 meters obtained for all wireless links between neighbors in the topology.

This indoor scenario is the worse situation for the SBGR protocol due to the error-prone nature of wireless communications [9]. As shown in the results of Chapter 3, in those conditions the distributed forwarding proposed by BLR suffers highly for duplicated packets which generate a huge transmission overhead and a lot of contention at the MAC layer. To address this issue, SBGR provides a simple solution by saving temporally the $BestRelays$ of each data packet to ignore duplicates from previous forwarders. Moreover the notification scheme of SBGR is based on the detection of false positions when messages are received outside the pre-established radio range. According the mean radio range, a node may detect wrongly a neighbor position as a sybil attack when the distance is longer than 45 meters due to a good wireless link without obstacles. For these reasons, the evaluation of SBGR in this indoor scenario has a special importance.

On the other hand, SIGF is designed for perfect wireless communications and suffers in realistic sensor networks [8]. In the real testbed, we study in detail SIGF's packets losses classified into four causes: unreceived responses, unreached next-hop, wrong reputation or insider attacks.

- Unreceived responses. The current sender broadcasts its RTS message to discover neighbors and receives no CTS message from candidates in the 60^o forwarding area. This metric determines the limitation of small forwarding areas in sparse network with a little density.

- Unreached next-hop. The sender discovers its neighborhood using short RTS/CTS messages, but the *DATA* message can not reach the selected

next-hop due to its weak wireless link. This metric determines the unreliability of discovering neighbors using shorter messages than the data payload.

- Wrong reputation. The forwarder is not able to select the next-hop among its neighbors because its reputation scheme discards all legitimate candidates due to their low forwarding success ratios. This happens frequently in realistic wireless communications where the packets reception is not guaranteed for the theoretical radio range.

- Insider attacks. The packets losses are generated by the malicious behavior of insider attackers (sinkhole or sybil).

Both protocols are also simulated in the TOSSIM simulator emulating the equivalent 35 nodes scenario. The link quality among nodes is represented as the Packet Reception Ratio (PRR) determined in the real deployment using a periodic beacon mechanism (see Section 3.3). Although obstacles are not simulated, these links qualities indicate the irregularities of wireless communications such as interferences and radio range variability. By doing that we try to simulate the reality with high accuracy.

Each experiment consists of 5 random sources sending a data packet to 10 random destinations. The data size is 110 bytes, and the delay among generated packets is 5 seconds to guarantee that there are no previous messages in the network. To compare the protocols fairly, nodes know their exact positions and are pre-configured with the same set of sources and destinations for each experiment. The experimental results are obtained by cumulative distribution functions (CDF) for the different performance metrics. Concretely the measured metrics are: packet delivery ratio (PDR), number of packets per hop and transmitted packets per delivery, described before. Moreover, we calculate the number of duplicated packets received per destination to evaluate the extra overhead of SBGR in networks with realistic communications.

5.4.7 Testbed Results with Sinkhole Attackers

Here, we analyze the performance of SBGR and SIGF with 3 sinkhole attackers in realistic sensor networks. Each sinkhole attacker receiving a DATA message replies immediately to pretend to be the best forwarding candidate. In realistic networks, sinkhole attackers participate in the forwarding process even when their positions are outside the mean radio range from the current forwarder. Although the SBGR protocol provides a simple protection against sinkhole attackers in order to achieve a high packet delivery ratio. Radio ranges variability causes in SBGR that nodes often detect sinkhole distant positions as sybil attacks and apply the flooding of notifications even when not needed. On the other hand in realistic networks, SIGF employing a contention window for defending from routing attacks often requires excessive CTS responses. In SIGF the usage of two small control messages (RTS/CTS) for discovering the neighborhood makes that bigger DATA messages are retransmitted for reaching selected next hops, as shown in Chapter 3. In addition, SIGF's reputation scheme based on the neighbors overhearing messages does not work property in realistic networks where the packet reception within the radio range is not guaranteed due to error-prone wireless medium.

Regarding the packet delivery ratio, Fig 5.19 shows that SBGR clearly outperforms SIGF. SIGF has many packet losses in all tested experiments due to the causes described below. However SBGR delivers successfully 98% of data packets in 80% of the experiments. The reason is that in the SBGR distributed forwarding nodes check the sender positions of *DATA* messages to avoid the cancellations of better candidates when a sinkhole attacker replies first. In few cases, the SBGR distributed forwarding presents packets losses due to wireless communication issues such as interferences and obstacles. Despite that results show the robustness of SBGR to defense against sinkhole attackers achieving a high delivery ratio.

To study in detail packets losses of SIGF, Table 5.1 shows the average percentage in all testbed experiments grouped into four causes: unreceived responses, unreached next-hop, wrong reputation and sinkhole attacks. As

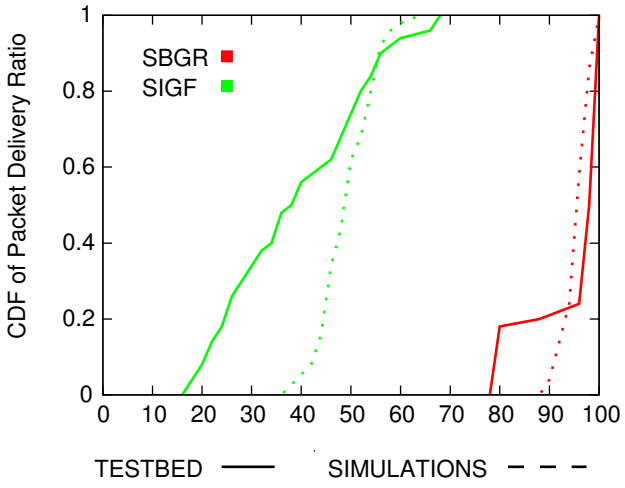


Figure 5.19: Packet Delivery Ratio

Lost Type	**SIGF**
Sinkhole Attacks	9%
Unreceived Response	10%
$Unreached\ Next-Hop$	2%
Wrong Reputation	79%

Table 5.1: Percentage of SIGF's lost packets grouped into four causes

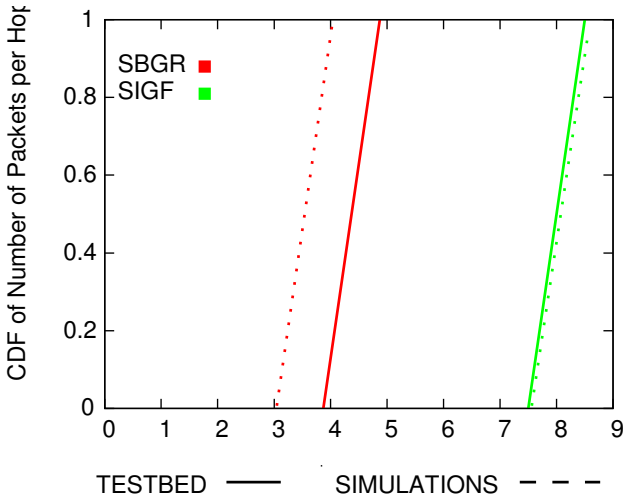


Figure 5.20: Number of Packets per Hop

expected, SIGF needs a learning time for its reputation scheme before penalizing sinkhole attackers for their dropped packets. Moreover to keep the reputation scheme, SIGF employs a small 60^o forwarding area reducing the number of candidates which can overhear each other. That has a performance degradation in sparse networks with little density. In addition, SIGF provides a RTS/CTS discovery scheme considering nodes with weak wireless links which can not receive bigger DATA messages. However the SIGF results show that wrong reputation is the cause of 79% of packets losses. That is, nodes discard as forwarding candidates all their neighbors due to their low reputations caused by wireless communications errors.

To demonstrate the efficiency of SBGR, Fig 5.20 shows the CDF of the messages transmitted per hop. Even with a higher delivery ratio, SBGR also has a lower transmission overhead than SIGF during the forwarding process. SBGR employs between 3 to 5 messages per hop in most of the experiments in comparison with at least 7 messages required by SIGF. Although the three-way handshake and contention windows of SIGF needs only 6 messages (1 RTS, 3 CTS, 1 DATA and 1 ACK), communications errors often generate unnecessary

Message Type	SBGR	SIGF
$NOTIFY$	43%	–
$DATA$	57%	14%
RTS	–	14%
CTS	–	60%
ACK	–	12%

Table 5.2: Percentage of messages types used by the protocols

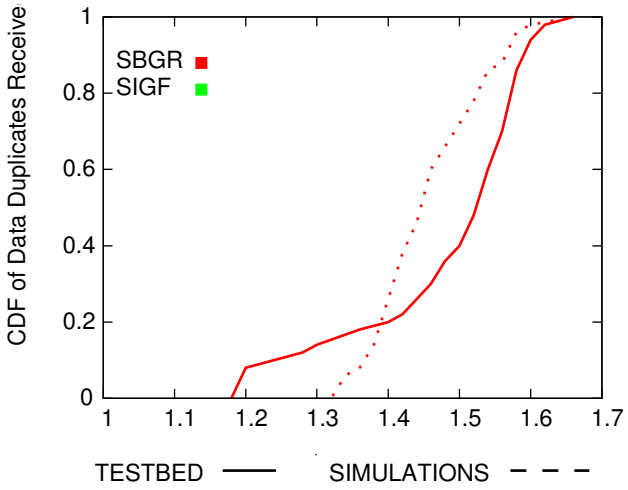


Figure 5.21: Duplicated Packets

CTS responses and DATA retransmissions to deliver the packet to the next hop. Despite that the SBGR distributed mode consists of an unique DATA message, the remaining transmissions are produced by duplicated *DATA* messages and *NOTIFY* messages even when there are no sybil attackers as shown in Table 5.2. The reason is that the notification scheme sometimes detects as sybil attackers legitimate nodes whose positions are outside the mean radio range (45 meters) of the current forwarder due to longer wireless links in realistic conditions. Despite those cases, the results prove that SBGR offers a better performance than SIGF in terms of a transmission overhead per hop even in networks with radio range variability.

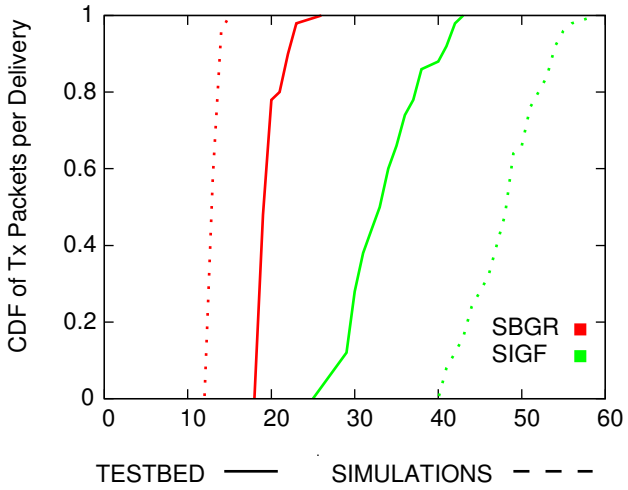


Figure 5.22: Tx Packets per Delivery

For realistic communications, we study the efficiency of the SBGR distributed forwarding in terms of duplicated packets as illustrated in Fig 5.21. As we expected, SBGR has a good performance with very few data duplicates received at destination. In fact, in all experiments SBGR produces less than an additional copy per data packet. The reason is that nodes providing advance compete distributively to forward the *DATA* message, and several duplicates may be transmitted simultaneously. In the next hop, nodes receiving various *DATA* messages save the copy with the closest sender as $BestRelay$ and discard the another copy. So, the destination only receives the data duplicates generated by the last hop. This simple solution prevents the forwarding of duplicated messages without extra overhead. The results confirm the reliability of SBGR to avoid the propagation of intermediate duplicates of *DATA* messages.

Regarding the end-to-end overhead of the protocols, Fig 5.22 shows the CDF of the messages transmitted per destination reached. As we can see, SBGR requires a lower amount of messages per data delivery than SIGF. In SBGR, simulated and testbed results are different due to realistic communications errors causing the increment of *DATA* and *NOTIFY* messages. Because the distributed

forwarding and notification scheme are based on the neighbors overhearing to minimize the number of transmissions. In the same way, SIGF has worse results in testbed experiments than simulated ones due to higher messages transmitted for unreachable destinations. SIGF has more lost packets due to the error-nature of wireless communication and the transmitted messages of those cases are useless. Again SBGR achieving almost perfect delivery ratio requires lower transmission overhead than SIGF.

5.4.8 Testbed Results with Sybil Attackers

This subsection presents the evaluation of SBGR and SIGF in the presence of 3 sybil attackers in a real deployment. A sybil attacker makes a virtual identity with a closer position than the current forwarder to become the next hop and cancel the data packet forwarding. To protect against sybil attackers, SBGR utilizes the notification scheme that detects and informs about false positions. However in realistic networks, the notification scheme detects legitimate nodes as sybil attacks and generates unnecessary overhead. SIGF is not able to detect the malicious behavior of sybil attackers by their multiple identities. Moreover SIGF produces more transmission overhead because of the DATA retransmissions to reach next-hops discovered by shorter RTS/CTS messages.

Regarding the robustness of the protocols against sybil attacks, Fig 5.23 confirms that SBGR outperforms clearly SIGF. SIGF has a poor performance which is less than 50% of delivery ratio in all realistic scenarios due to many causes explained in the following. Unlike SIGF, SBGR provides a 90% of successful delivered packets in 80% of the experiments. Again, collisions and interferences in wireless medium generate *DATA* messages losses of SBGR in a few cases. The distributed forwarding of SBGR fails when the *DATA* message is lost, and no neighbors receive it successfully. The notification scheme of SBGR fails when in the detection step the *DATA* message from a sybil attacker is lost, and no farther neighbors detect its false position. Despite that case, the notification flooding is almost always able to inform the false positions used by sybil attackers. Even in testbeds with few neighbors per node, the results show the reliability of

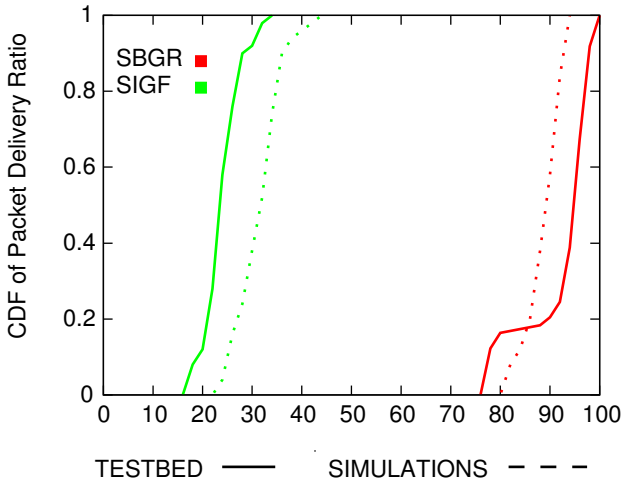


Figure 5.23: Packet Delivery Ratio

Lost Type	**SIGF**
Sybil Attacks	80%
Unreceived Response	3%
Unreached Next − Hop	3%
Wrong Reputation	14%

Table 5.3: Percentage of lost packets in SIGF grouped into four causes

SBGR notification scheme to defend from sybil attacks and SBGR achieves an almost perfect delivery ratio.

To analyze in more detail the delivery ratio of SIGF, Table 5.3 determines the percentage of lost packets grouped into four causes. As expected, the reputation scheme of SIGF is not able to protect against sybil attacker in fact it is the main reason of lost packets. Moreover, the reputation scheme discards legitimate nodes due to their low reputations for communication errors in wireless links. SIGF also provides a forwarding area and a neighborhood discovery scheme that are not designed to low density networks and realistic wireless communications.

Regarding the control overhead during the forwarding process, Fig 5.24

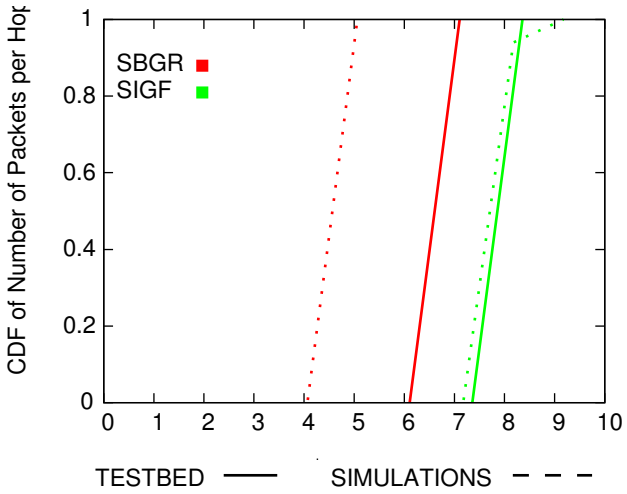


Figure 5.24: Number of Packets per Hop

demonstrates that SBGR requires less hop-by-hop transmissions than SIGF. In realistic scenarios with sybil attackers, SBGR needs about 6 to 7 messages per hop in most of the cases due to some *DATA* duplicates and the usage of the *NOTIFY* flooding. However SIGF requires about 7 to 9 messages during the three-way handshake scheme consisting of 4 messages. As Table 5.4 shows, SIGF needs a large number of *CTS* responses in the contention window which is useless to defend from sybil attackers. Again, *DATA* retransmission is used to deliver the packet to neighbors discovered by shorter *RTS/CTS* messages. The results show that SBGR achieves a high packet delivery ratio providing a moderate transmission overhead which is less than SIGF.

Regarding the efficiency of the SBGR distributed forwarding in realistic deployments, Fig 5.25 shows the data duplicates received per destination. Again, SBGR provides an excellent performance with less than a duplicated copy per data packet. As we mentioned above, nodes save temporally $BestRelays$ to prevent the propagation of intermediate *DATA* duplicates. The figure proves that SBGR is able to avoid duplicated packets without extra overhead.

According to the transmission overhead in the whole path, Fig 5.26 shows

Message Type	SBGR	SIGF
$NOTIFY$	64%	–
$DATA$	36%	14%
RTS	–	14%
CTS	–	65%
ACK	–	7%

Table 5.4: Percentage of messages types used by the protocols

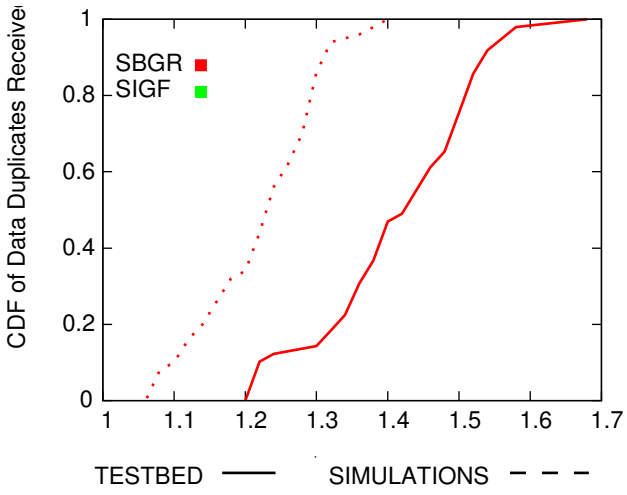


Figure 5.25: Duplicated Packets

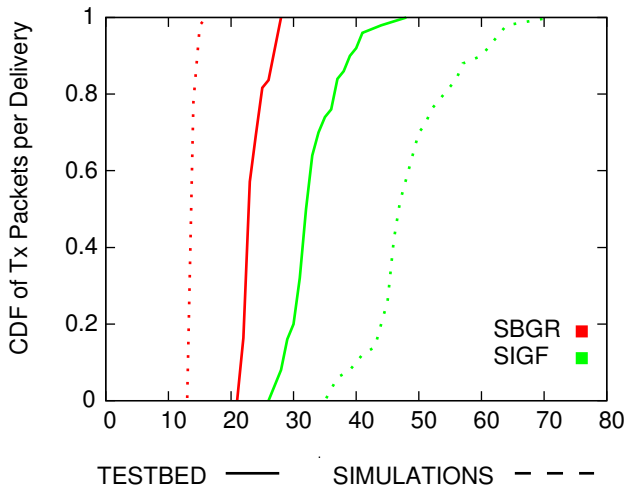


Figure 5.26: Tx Packets per Delivery

that SBGR also provides a better performance than SIGF. As in the case of sinkhole, SIGF has worse results in real experiments than simulation ones due to wireless interferences generating unnecessary *DATA* and *CTS* messages. Like SIGF, SBGR has a lower transmissions in simulation than in real experiments because communication errors cause more *DATA* and *NOTIFY* messages. The reason is that in SBGR the distributed forwarding and flooding scheme are based on the neighbors overhearing to cancel unnecessary transmissions. Despite those cases, SBGR offers a good balance between high delivery and low overhead even in sparse networks with error-prone communications.

5.5 Conclusions

This chapter presents a Self-Protected Beaconless Geographic Routing (SBGR) for WSNs. SBGR is designed to provide a high delivery ratio in the presence of insider attacks such as sinkhole and sybil.

To improve the knowledge about insider attacks, we have studied their effects in beaconless geographic routing protocols. In fact, those insider attacks

damage severely beaconless routing protocols by dropping packets during the forwarding process. These attacks are more dangerous in beaconless protocols using three-way handshake schemes to select explicitly next hops than those protocols providing a fully distributed forwarding. Moreover a detailed analysis of sybil attackers demonstrates the relationship between their false position, the probability of neighbors detection and the probability of successful packet forwarding. Our analysis concludes that false positions closer to the destination cancel more forwarding candidates, but the detection probability is increased.

According our previous studies, SBGR provides two forwarding mechanisms to protect data packets against sinkhole and sybil attackers. In the distributed forwarding, each node avoids sinkhole attackers by ignoring their first *DATA* transmissions if it provides more advance than themselves. The limited flooding of *NOTIFY* messages is used to defend from sybil attackers that create false identities with closer positions to the destination for canceling forwarding candidates. In that case, nodes detect the false position and diffuse a *NOTIFY* messages in order to propagate the *DATA* message beyond the influence area of the sybil attacker where the distributed forwarding is resumed.

For sensor networks with constrained resources, the SBGR implementation requires only storing temporarily the state of messages to avoid duplicated *DATA* and malicious *NOTIFY* messages. Sensor nodes store only the identification tuple of each data packet received. Moreover they maintain a $BestRelay$ per data packet received to discard messages duplicated from previous relays. To avoid the malicious usage of the notification scheme, nodes check the identification tuple of data packets and its $BestRelay$ to avoid the flooding overhead of *NOTIFY* duplicates and the notification cancellation for false *NOTIFY* messages.

Finally, SBGR has been evaluated against SIGF, the only known secure beaconless geographic routing protocol in WSNs, using both extensive simulations and a real testbed network. The simulated results show that SBGR is able to obtain almost 100% delivery ratio even in networks with tens of sinkhole and sybil attackers. In contrast SIGF needs some dropped packets to detect sinkhole attacks, and its reputation scheme is not able to prevent sybil attackers

employing multiple identities. Moreover the real testbed evaluation proves that SBGR outperforms SIGF in terms of delivery reliability and transmission efficiency. In conclusion, all results confirm that SBGR provides an efficient and robustness communication solution for wireless sensor applications deployed in insecure environments.

Chapter 6

Conclusions

6.1 Summary and Main Contributions

This thesis studies the problem of providing reliable communication solutions in realistic wireless sensor networks (WSNs). Concretely, after analyzing the WSNs requirements we focus on designing multihop routing protocols adapting to specific features of scalability and efficiency. The organization of the thesis shows the progression of the work. Starting with a set of assumptions considered for previous solutions which have been eliminated to achieve solutions that are able to work in realistic conditions.

Initially we consider the requirements of WSNs to study the routing solutions proposed in the literature. For WSNs, we summarize the state-of-the-art of the main routing algorithms classified into four paradigms: data-centric, hierarchical, QoS and geographic. As we have shown, data-centric, hierarchical and QoS protocols need routing table maintenance or flooding discovery activities that confront with the WSNs requirements of scalability and efficiency. In contrast, the geographic routing paradigm has been proposed as the most effient and scalable solution where nodes only require local neighborhood information to take routing decisions. Given that each node needs a minimum state to store only 1-hop neighbors positions, geographic routing decreases memory, traffic, computation and consumption.

Most geographic routing algorithms combine greedy and face strategies to provide an efficient and robust solution for networks with any density of neighbors. The greedy forwarding strategy advances by reducing the distance toward the destination selecting the closest neighbor in each hop in order to achieve the shortest path in uniformly-dense networks. The face routing strategy guarantees the packet delivery even in sparse networks with void areas where nodes may have no neighbors providing advance. To discover neighborhood positions, several geographic routing algorithms employ periodic transmissions of short control messages, called beacons, between 1-hop neighbors. Alternatively, beacon-less geographic algorithms have been proposed to discover reactively 1-hop neighbors in order to reduce the bandwidth and increase the efficiency.

For realistic wireless sensor networks, we propose BOSS (*Beaconless On demand Strategy*), a geographic routing protocol designed to deal with communications errors. The design of BOSS is based on our empirical analysis with real wireless radios determining the strong relationship between big packets and low reception probatilities. This protocol introduces a neighborhood discovery technique which sends first the big data payload to guarantee that only neighbors that are able to receive the data participate in the selection phase with smaller control messages. Moreover, BOSS includes a delay function combining greedy and face strategies and a passive acknowledgment mechanism guaranteeing the hop-to-hop delivery in order to reduce collisions and control overhead. The evalution of BOSS against the two better beaconless protocols (IGF and BLR) were performed both in simulated networks with thousands of nodes and in a real-testbed network with irregular radio ranges and unidirectioanl links. In all tested scenarios, BOSS outperforms IGF and BLR in terms of delivery reliability, bandwitdh efficiency and end-to-end performance. The results confirm that the BOSS design mitigates the inherent problems of wireless communications in WSNs.

After dealing with communications errors, a BOSS extension called EGLE is presented as an effective greedy routing solution supporting the inaccuracy of distributed positioning systems used in real deployments. We analyze the

effects of location errors in greedy routing and determine that greedy routing fails even in dense networks due to false void areas when a node has no neighbors closer to the destination based on their estimated positions, but their real positions provide advance. To reduce the failures for false void areas, EGLE provides three routing modes: greedy to prevent voids, alternative to exit from voids and broadcast to ensure the delivery to the destination. The performance of EGLE is evaluated against three relevant protocols: BOSS (our robust beaconless scheme for wireless communications), GRS (the original greedy strategy) and MER(the best performance approach considering location errors). In simulation and testbed evaluations, EGLE's three routing modes enhance progressively its performance outperforming BOSS, GRS and MER in terms of delivery ratio and in bandwidth overhead per reached destination. All results show that EGLE provides a good balance between little control overhead and high delivery ratio (above the 90%) even in real networks with a location error of 100% of the radio range.

In addition, WSNs are often deployed in unsafe environments where malicious nodes may affect wireless communication, and thus we present a Self-Protected Beaconless Greedy Routing (SBGR) protocol that provides a high delivery ratio in the presence of attackers. Our analysis of routing attacks shows that in beaconless greedy routing the two most dangerous attacks are sinkhole and sybil which interpect any packet in their coverage areas and prevent data forwarding by exploiting the reactive next-hop selection based on delayed responses and position information. Moreover, the analysis demonstrates that existing security mechanisms (i.e. cryptography, reputation and location verification) are ineffective against these attackers and need expensive developments in terms of complexity and hardware. Based on our analysis, SBGR enhances the fully distributed scheme proposed in BLR to ignore sinkhole attackers as well as a sophisticated flooding mechanism to guarantee that the packet advances in the coverage area of sybil attackers. In SBGR the two protection mechanisms require only that nodes store temporally the status of forwarded packets in their coverage areas enabling simple solutions for routing attacks in sensor networks with constrained resources. The SBGR evaluation is performed by simulation and

testbed experiments. We compare our proposal against SIGF, the only known secure beaconless protocol in WSNs so far. All results confirm that SBGR outperforms SIGF defending from routing attacks and provides an efficient and robust communication solution for WSNs deployed in unsafe environments.

The next sections summarize the main publications that are derived from this thesis and some future works.

6.2 Publications

This section enumerates the publications which are directly related to the development of the thesis. Here, we only consider peer-reviewed international publications.

6.2.1 Journals and magazines

- Sanchez J.; Ruiz P. and Marin-Perez R. "Beacon-less geographic routing made practical: challenges, design guidelines, and protocols", IEEE Communications Magazine, Vol 47(8), pages 85-91, 2009.
- Sanchez J.A.; Marin-Perez R. and Ruiz P.M. "Beacon-less geographic routing in real wireless sensor networks", Journal of Computer Science and Technology, Vol 23(3), pages 438-450, 2008.
- Sanchez J.A.; Marin-Perez R. and Ruiz, P.M. "Beacon-less geographic multicast routing in a real-world wireless sensor network testbed", Wireless Networks, pages 1-14, 2012.

6.2.2 Conferences

- Sanchez J.A.; Marin-Perez R. and Ruiz P.M. "BOSS: Beacon-less on demand strategy for geographic routing in wireless sensor networks", The 4th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), pages 1-10, 2007.

- Sanchez J.A.; Marin-Perez R. and Ruiz P.M. ”BRUMA: Beacon-less geographic routing for multicast applications”, The 34th IEEE Conference on Local Computer Networks (LCN), pages 522-529, 2009.

- Marin-Perez R. and Ruiz P.M. ”Effective geographic routing in wireless sensor networks with inaccurate location information”, The 10th International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW), 2011.

- Marin-Perez R. and Ruiz P.M. ”A simple self-protected beaconless geographic routing for wireless sensor networks”, The 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems. (MASS), 2011.

6.3 Future works

Although geographic routing has been a hot research topic for several years, there are many interesting issues which still need a deeper study for realistic wireless sensor networks. Among all them, we just introduce here those issues which are directly related to the work developed along this thesis.

Our research on geographic routing assume the knowledge of position information for stationary nodes, however mobility can lead to wrong or out-dated position information. In mobile scenarios like Vehicular Adhoc NETworks (VANET), most geographic routing solutions employ periodic beacons messages to know available neighbors and take routing decisions. These beacons include among other information geographic coordinates of the vehicle, direction and speed. However, the usage of that information for taking routing decisions can result in inefficiencies such as temporal loops in the forwarding path, backward progress due to stale information and transmission failures for low-quality links [159]. To address these issues, we proposed a beaconless geographic routing protocol called BRAVE [160] in which neighbor selection is done opportunistically in collaboration with neighbors. That is, the next forwarder

for the DATA message is reactively selected among those neighbors that have successfully received this message. The results show that BRAVE provides a high delivery ratio but a high end-to-end delay. Reducing the end-to-end delay is fundamental in emergency situations where the response time is key. Therefore, we plan to incorporate the fully distributed scheme of SBGR to the BRAVE design in order to provide a reliable and fast routing solution for vehicular applications.

In addition, our geographic routing protocols consider that nodes have some rendezvous schemes to establish sleeping and waking cycles for saving energy and increasing the network lifetime. Many proactive rendezvous approaches have been proposed for the MAC (Medium Access Control) layer such as SPAN [161], STEM [162] and GAF [107]. SPAN identifies multiple sets of disjoint sets where each set provides connectivity to the whole network. STEM allows nodes to sleep periodically and uses beacons to rendezvous with the targeted node. GAF defines square grids where all nodes in neighboring grids can communicate with each other. But these rendezvous schemes require flooding mechanisms or periodic beacon exchanging. Thus, they are not scalable in dense network with thousands of nodes and waste constrained resources such as energy and bandwidth, concretely in nodes not taking part in any routing process. To overcome such issues, pseudo-asynchronous schemes [163] have been presented where nodes establish rendezvous on demand. The current forwarder having a data packet sends a RTS packet specifying the forwarding region and waits for a CTS reply. The forwarder retransmits the RTS packet every N seconds until it receives successfully a CTS packet. Then the forwarder forwards the DATA packet to the neighbor in the forwarding region which sent replies. However as we demonstrated in Section 3, the traditional RTS/CTS mechanism may discover a neighbor whose reception probability of the bigger DATA packet may be very low and generates many retransmissions and packet losses in the selection phase. For this reason, we propose to develop an on-demand cross-layer routing protocol that combines a pseudo-asynchronous rendezvous mechanism and our beaconless geographic routing scheme proposed in BOSS. The main idea is that the current forwarder employs a series of DATA messages until its neighbors

are discovered and rendezvoused. Neighbors receiving DATA message keep awake and participate in the forwarding process. When the forwarder receiving RESPONSE messages from neighbors can perform the selection phase. The neighbor being selected become to the next forwarder and the remaining of neighbors can go to sleep mode.

Finally, some applications of WSNs require distributed control of many sensor device. These applications make extensive use of many-to-one communications. Thus, the design of efficient multicast routing protocols is fundamental to support the distributed operation of WSNs. Most geographic multicast routing protocols in the literature are based on extensions of Greedy-Face-Greedy [130] such as PBM [164] and GMR [165]. PBM was the first geographic multicast protocol proposed, and GMR solved some scalability problems of PBM while achieving better results in terms of bandwidth consumption. Both protocols employ periodic beacon messages to exchange their positions and identifiers of neighboring nodes. As we have shown, the usage of beacons introduces severe problems in real deployments in terms of collisions, unnecessary waste of resources, etc. To avoid periodic beacons, we proposed BRUMA [166], a beacon-less geographic multicast routing protocol for wireless sensor networks. BRUMA employs DATA packets to discover reactively the set of candidate relays for taking multicast routing decisions. By doing that, BRUMA is able to perform very well in realistic wireless communications with interferences, collisions, etc. However, BRUMA neglects the location inaccuracy of positioning system used in realistic WSNs. Therefore similarly to EGLE, we plan to enhance the BRUMA algorithm by considering location errors in routing decisions.

Bibliography

[1] I. Khemapech, I. Duncan, and A. Miller, "A survey of wireless sensor networks technology," *Proceedings of the 6th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, 2005.

[2] J. L. Hill, "System architecture for wireless sensor networks," in *Doctoral Dissertation*. University of California, Berkeley, 2003.

[3] C. yee Chong, Ieee, S. P. Kumar, and S. Member, "Sensor networks: evolution, opportunities, and challenges," in *Proceedings of the IEEE*, vol. 91, no. 8, 2003, pp. 1247–1256.

[4] Y. Yu, V. Prasanna, B. Krishnamachari, and V. Kumar, "Information processing and routing in wireless sensor networks," in *World Scientific Pub Co Inc*, 2006.

[5] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. 6th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*. New York, NY, USA: ACM Press, 2000, pp. 120–130.

[6] S. Giordano, I. Stojmenovic, and L. Blazevie, "Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy," *Ad Hoc Wireless Networking*, pp. 103–136, 2004.

[7] H. Füßler, M. Mauve, H. Hartenstein, C. Lochert, D. Vollmer, D. Herrmann, and W. Franz, "Position-Based Routing in Ad-Hoc Wireless Networks," in *Inter-Vehicle-Communications Based on Ad Hoc Networking Principles—The FleetNet Project*, W. Franz, H. Hartenstein, and M. Mauve, Eds. Karlsruhe, Germany: University of Karlsruhe, Nov 2005, pp. 117–143.

[8] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," in *Proc. First International Conference on Embedded Networked Sensor Systems (SenSys '03)*. New York, NY, USA: ACM Press, 2003, pp. 1–13.

[9] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '04. New York, NY, USA: ACM, 2004, pp. 78–82.

[10] H. A. Oliveira, E. F. Nakamura, A. A. F. Loureiro, and A. Boukerche, "Error analysis of localization systems for sensor networks," in *GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems*. New York, NY, USA: ACM, 2005, pp. 71–78.

[11] R. Shah, A. Wolisz, and J. Rabaey, "On the performance of geographical routing in the presence of localization errors," in *IEEE International Conference on Communications*. ICC, 2005.

[12] M. Witt and V. Turau, "The Impact of Location Errors on Geographic Routing in Sensor Networks," in *Proceedings of the Second International Conference on Wireless and Mobile Communications (ICWMC'06)*, Bucharest, Romania, 2006, p. 76.

[13] Y. Kim, J.-J. Lee, and A. Helmy, "Modeling and analyzing the impact of location inconsistencies on geographic routing in wireless networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 1, pp. 48–60, 2004.

[14] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, vol. 1, no. 2–3, pp. 293–315, September 2003.

[15] A. Mei and J. Stefa, "Routing in outer space: fair traffic load in multi-hop wireless networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '08. New York, NY, USA: ACM, 2008, pp. 23–32.

[16] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," *Architecture Support for Programming Languages and Operating Systems*, vol. 35, no. 11, pp. 93–104, 2000.

[17] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *Proc. of the First ACM Conference on Embedded Networked Sensor Systems, (SenSys 2003)*, pp. 126–137, November 2003.

[18] T. J. Van Der Werff, "10 emerging technologies that will change the world," *Technology Review*, vol. 2, no. February, pp. 32–52, 2003.

[19] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, March 2006.

[20] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th international conference on*

Information processing in sensor networks, ser. IPSN '07. New York, NY, USA: ACM, 2007, pp. 254–263.

[21] L. Schwiebert, S. K. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 151–165.

[22] A. Milenković, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Computer Communications (Special issue: Wireless Sensor Networks: Performance, Reliability, Security, and Beyond)*, vol. 29, no. 13–14, pp. 2521–2533, 2006.

[23] J. G. Koomey, S. Berard, M. Sanchez, and H. Wong, "Implications of historical trends in the electrical efficiency of computing," *IEEE Annals of the History of Computing*, vol. 33, pp. 46–54, 2011.

[24] C. Schurgers and M. Srivastava, "Energy efficient routing in wireless sensor networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, vol. 1. IEEE, 2001, pp. 357–361.

[25] W. E. R. R. P. D. T. L. H. M. Fox, C. G. and A. E. Schreiner, "Acoustic detection of a seafloor spreading episode on the juan de fuca ridge using military hydrophone arrays," in *Geophysical Research Letters*, vol. 22, 1995, pp. 131–134.

[26] W. P. Delaney, "Air defense of the united states: Strategic missions and modern technology," in *International Security*, vol. 15, 1990, pp. 181–211.

[27] M. Metcalf, "Acoustics on the 21st century battlefield," in *Technical Report in Joint Force Quarterly*, 1996, pp. 44–47.

[28] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in Packet Radio Technology," in *Proceedings of the IEEE*, vol. 66, no. 11. IEEE, nov 1978, pp. 1408–1496.

[29] W. Fifer and F. Bruno, "The Low-Cost Packet Radio," in *Proceedings of the IEEE 75 (1)*, jan 1987, pp. 33–42.

[30] R. S. A. Leiner, B; Ruth, "Goals and challenges of the darpa glomo program," in *IEEE Personal Communications*, dec 1996, pp. 34–43.

[31] G. Compare, "Us army puts tactical internet to test," *Defense News*, p. 3, 1997.

[32] E. Althouse, "Extending the Littoral Battlespace (ELB)," *Advanced Concept Technology Demonstration (ACTD), NATO Information Systems Technology Panel Symposium on Tactical Mobile Communications*, jun 1999.

[33] *System on Chip Age*, 1993.

[34] "Jn5148 ieee802.15.4 wireless microcontroller, http://www.jennic.com."

[35] B. A. Warneke and K. S. J. Pister, "Mems for distributed wireless sensor networks," in *9th Int'l Conf on Electronics, Circuits and Systems*, 2002, pp. 291–294.

[36] *Low stand-by power complementary field effect circuitry*, Dec 1967.

[37] "Cc2420 2.4 ghz ieee 802.15.4 / zigbee rf transceiver, chipcon product datasheet, http://www.chipcon.com/."

[38] R. Cardell-Oliver, K. Smettem, M. Kranz, and K. Mayer, "A reactive soil moisture sensor network: Design and field evaluation," *International journal of distributed sensor networks*, vol. 1, no. 2, pp. 149–162, 2005.

[39] R. Pon, M. Batalin, V. Chen, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, A. Somasundra, Y. Yu, M. Hansen *et al.*, "Coordinated static and mobile sensing for environmental monitoring," in *DCOSS*. Springer, 2005, pp. 403–405.

[40] K. Martinez, R. Ong, and J. Hart, "Glacsweb: a sensor network for hostile environments," in *The First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, 2004, pp. 81–87.

[41] "Corie project, http://www.ccalmr.ogi.edu/corie/."

[42] R. Kremens, J. Faulring, A. Gallagher, A. Seema, and A. Vodacek, "Autonomous field-deployable wildland fire sensors," *International Journal of Wildland Fire*, vol. 12, no. 2, pp. 237–244, 2003.

[43] "Alert systems, automated local evaluation in real time, http://www.alertsystems.org."

[44] A. Cerpa, J. Elson, D. Estrin, and L. Girod, "Habitat monitoring: Application driver for wireless communications technology," *In ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, vol. 31, no. 2, pp. 20–41, April 2001.

[45] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ser. WSNA ́02. New York, NY, USA: ACM, 2002, pp. 88–97.

[46] S. Yang, "Redwoods go high tech: Researchers use wireless sensors to study california's state tree," *University of California at Berkeley News (UCNEWS)*, 2003.

[47] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," in *ACM Sigplan Notices*, vol. 37, no. 10. ACM, 2002, pp. 96–107.

[48] D. Ingraham, R. Beresford, K. Kaluri, M. Ndoh, and K. Srinivasan, "Wireless sensors: Oyster habitat monitoring in the bras d'or lakes," *Distributed Computing in Sensor Systems*, pp. 467–467, 2005.

[49] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis, "Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005, pp. 64–75.

[50] M. L. McKelvin, Jr., M. L. Williams, and N. M. Berry, "Integrated radio frequency identification and wireless sensor network architecture for automated inventory management and tracking applications," in *Proceedings of the 2005 conference on Diversity in computing*, ser. TAPIA '05. New York, NY, USA: ACM, 2005, pp. 44–47.

[51] J. Lynch, "Decentralization of wireless monitoring and control technologies for smart civil structures," in *Ph.D. Thesis, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA*, July 2002.

[52] E. Straser, A. Kiremidjian, and T. Meng, "Modular, wireless damage monitoring system for structures," in *US Patent 6,292,108*, 18 sep 2001.

[53] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 13–24.

[54] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*. ACM, 2004, pp. 270–283.

[55] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. Stankovic *et al.*, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 1–38, 2006.

[56] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob2005), IEEE International Conference on*, vol. 3. IEEE, 2005, pp. 253–259.

[57] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic, "Alarm-net: Wireless sensor networks for assisted-living and residential monitoring," *University of Virginia Computer Science Department Technical Report*, 2006.

[58] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC'06)*. Citeseer, 2006, pp. 640–644.

[59] I. A. Essa, "Ubiquitous sensing for smart and aware environments," *IEEE Personal Communications*, vol. 7, no. 5, pp. 47–49+, 2000.

[60] M. Srivastava, R. Muntz, and M. Potkonjak, "Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 132–138.

[61] *Commercial-Off-The-Shelf (COTS): A Survey*, 2000.

[62] G. E. Moore, "Cramming More Components Onto Integrated Circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, April 1965.

[63] K. Bult, A. Burstein, D. Chang, M. Dong, M. Fielding, E. Kruglick, J. Ho, F. Lin, T. Lin, W. Kaiser *et al.*, "Low power systems for wireless microsensors," in *Proceedings of the 1996 international symposium on Low power electronics and design*. IEEE Press, 1996, pp. 17–21.

[64] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[65] B. Warneke, M. Last, B. Liebowitz, and K. S. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Computer*, vol. 34, no. 1, pp. 44–51, 2001.

[66] K. Fishkin, K. Partidge, and S. Chatterjee, "Wireless user interface components for personal area networks," *Pervasive Computing, IEEE*, vol. 1, no. 4, pp. 49–55, 2002.

[67] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," *Micro, IEEE*, vol. 22, no. 6, pp. 12–24, 2002.

[68] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, pp. 48–es.

[69] "Tmote sky: Low power wireless sensor module datasheet, http://sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf."

[70] B. Crow, I. Widjaja, L. Kim, and P. Sakai, "Ieee 802.11 wireless local area networks," *Communications Magazine, IEEE*, vol. 35, no. 9, pp. 116–126, 1997.

[71] J. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "Ieee 802.15. 4: a developing standard for low-power low-cost wireless personal area networks," *Network, IEEE*, vol. 15, no. 5, pp. 12–19, 2001.

[72] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *Sensor and Ad Hoc*

Communications and Networks, 2006. SECON06. 2006 3rd Annual IEEE Communications Society on, vol. 1. IEEE, 2006, pp. 286–295.

[73] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[74] 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.

[75] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 162–175.

[76] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, 2002.

[77] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[78] Y. Al-Obaisat and R. Braun, "On wireless sensor networks: architectures, protocols, applications, and management," in *International Conference on Wireless Broadband and Ultra Wideband Communications (Auswireless 2006), March*, 2006, pp. 13–16.

[79] L. García Villalba, A. Sandoval Orozco, A. Triviño Cabrera, and C. Barenco Abbas, "Routing protocols in wireless sensor networks," *Sensors*, vol. 9, no. 11, pp. 8399–8421, 2009.

[80] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1999, pp. 174–185.

[81] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 56–67.

[82] D. Braginsky and D. Estrin, "Rumor routing algorthim for sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 22–31.

[83] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD record*, vol. 31, no. 3, pp. 9–18, 2002.

[84] N. Sadagopan, B. Krishnamachari, and A. Helmy, "The acquire mechanism for efficient querying in sensor networks," in *In IEEE International Workshop on Sensor Network Protocols and Applications (SNPA'03)*, 2003, pp. 149–155.

[85] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, ser. HICSS ́00, vol. 8. Washington, DC, USA: IEEE Computer Society, 2000, p. 8020.

[86] *PEGASIS: Power-efficient gathering in sensor information systems*, vol. 3, 2002.

[87] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Parallel and Distributed Processing Symposium., Proceedings 15th International*, 2001, pp. 2009–2015.

[88] L. Subramanian, , L. Subramanian, and Y. H. Katz, "An architecture for building self-configurable systems," in *In MobiHoc*, 2000, pp. 63–73.

[89] *APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks*, 2002.

[90] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for self-organization of a wireless sensor network," *Personal Communications, IEEE*, vol. 7, no. 5, pp. 16–27, 2000.

[91] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "Speed: A stateless protocol for real-time communication in sensor networks," in *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*. IEEE, 2003, pp. 46–55.

[92] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," in *Addison-Wesley*, C. Perkins, Ed., 2001, pp. 139–172.

[93] C. E. Perkins, "Ad-hoc On-Demand Distance Vector Routing," in *IN PROCEEDINGS OF THE 2ND IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, November 1997, pp. 90–100.

[94] K. Xu, G. Takahara, and H. Hassanein, "On the robustness of grid-based deployment in wireless sensor networks," in *Proceedings of the 2006 international conference on Wireless communications and mobile computing*. ACM, 2006, pp. 1183–1188.

[95] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *Personal Communications, IEEE*, vol. 7, no. 5, pp. 28–34, 2000.

[96] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," in *MobiCom 01́: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2001, pp. 166–179.

[97] S. Čapkun, M. Hamdi, and J. Hubaux, "Gps-free positioning in mobile ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 157–167, 2002.

[98] H. Akcan, V. Kriakov, H. Brönnimann, and A. Delis, "Gps-free node localization in mobile wireless sensor networks," in *MobiDE '06: Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access*. New York, NY, USA: ACM, 2006, pp. 35–42.

[99] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *IN GLOBECOM*, 2001, pp. 2926–2931.

[100] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," *Network Protocols, IEEE International Conference on*, vol. 0, p. 0035, 2001.

[101] H. Oliveira, E. Nakamura, and A. Loureiro, "Directed position estimation: A recursive localization approach for wireless sensor networks," in *In Proceedings of the 14th IEEE International Conference on Computer Communications and Networks*, San Diego, USA, 2005, p. 557.

[102] R. Friedman and G. Kliot, "Location services in wireless ad hoc and hybrid networks: A survey," *Techanical Report, Technion Computer Science*, 2006.

[103] K. Seada and A. Helmy, "Geographic protocols in sensor networks," *Encyclopedia of Sensors, American Scientific Publishers (ASP)*, 2004.

[104] C. Lemmon, S. Lui, and I. Lee, "Review of location-aware routing protocols," *Advances in Information Sciences and Service Sciences*, vol. 2, pp. 132–143, 2010.

[105] S. Ruehrup, "Theory and practice of geographic routing," *Ad Hoc and Sensor Wireless Networks: Architectures, Algorithms and Protocols, Bentham Science Publishers*, 2009.

[106] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Math.*, vol. 86, no. 1-3, pp. 165–177, January 1991.

[107] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, pp. 70–84.

[108] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wächli, "BLR: Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks," *Elsevier's Computer Communications Journal (ECC)*, vol. 27, no. 11, pp. 1076–1086, July 2004.

[109] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-Based Forwarding for Mobile Ad Hoc Networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351 – 369, 2003.

[110] B. Blum, T. He, S. Son, and J. Stankovic, "IGF: A state-free robust communication protocol for wireless sensor networks," in *Technical Report in Department of Computer Science, University of Virginia, USA*, 2003.

[111] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *Communications, IEEE Transactions on*, vol. 32, no. 3, pp. 246–257, 1984.

[112] G. G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," in *Technical Report in Information Sciences Institute*, no. ISI/RR-87-180, 1987.

[113] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," in *Proc. 11 th Canadian Conference on Computational Geometry*, Vancouver, August 1999, pp. 51–54.

[114] P. Bose and P. Morin, "Online routing in triangulations," *Algorithms and Computation*, pp. 113–122, 1999.

[115] P. Bose, A. Brodnik, S. Carlsson, E. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, and J. Munro, "Online routing in convex subdivisions," *Algorithms and Computation*, pp. 1–90, 2000.

[116] T. Hou and V. Li, "Transmission range control in multihop packet radio networks," *Communications, IEEE Transactions on*, vol. 34, no. 1, pp. 38–44, 1986.

[117] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 12, no. 11, pp. 1122–1133, 2001.

[118] I. Stojmenovic, "Localized network layer protocols in wireless sensor networks based on optimizing cost over progress ratio," *Network, IEEE*, vol. 20, no. 1, pp. 21–27, 2006.

[119] J. Sanchez and P. Ruiz, "Locally optimal source routing for energy-efficient geographic routing," *Wireless Networks*, vol. 15, no. 4, pp. 513–523, 2009.

[120] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[121] K. Gabriel and R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology*, vol. 18, pp. 259–278, 1969.

[122] G. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition*, vol. 12, pp. 261–268, 1980.

[123] D. Lee and A. Lin, "Generalized delaunay triangulation for planar graphs," *Discrete & Computational Geometry*, vol. 1, no. 1, pp. 201–217, 1986.

[124] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanners for routing in mobile networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 1, pp. 174–185, 2005.

[125] X. Li, I. Stojmenovic, and Y. Wang, "Partial delaunay triangulation and degree limited localized bluetooth scatternet formation," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, no. 4, pp. 350–361, 2004.

[126] H. Frey and S. Rührup, "Paving the way towards reactive planar spanner construction in wireless networks," in *Kommunikation in Verteilten Systemen (KiVS)*. Springer, 2009, pp. 17–28.

[127] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "On the pitfalls of geographic face routing," in *Proceedings of the 2005 joint workshop on Foundations of mobile computing (DIALM-POMC '05)*. New York, NY, USA: ACM, 2005, pp. 34–43.

[128] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 2006, pp. 390–401.

[129] Y.-J. K. R. Govindan, B. Karp, and S. Shenker, "Lazy cross-link removal for geographic routing," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys ́06. ACM, 2006, pp. 112–124.

[130] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.

[131] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically Optimal Geometric Mobile Ad-Hoc Routing," in *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, (Dial-M)*, A. Press, Ed., 2002, pp. 24–33.

[132] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. 6th annual ACM/IEEE International*

Conference on Mobile Computing and Networking (MobiCom '00). New York, NY, USA: ACM Press, 2000, pp. 243–254.

[133] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc '03)*, ACM New York, NY, USA, 2003, pp. 267–278.

[134] H. Kalosha, A. Najak, S. Ruhrup, and I. Stojmenovic, "Select-and-protest-based beaconless georouting with guaranteed delivery in wireless sensor networks," in *Proc. 27th Conference on IEEE Computer Communications (INFOCOM 08)*, April 2008, pp. 346–350.

[135] M. Zorzi and R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," *IEEE Transaction on Mobile Computing*, vol. 2, no. 4, pp. 349–365, 2003.

[136] A. Cerpa, J. L. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical model of lossy links in wireless sensor networks," in *Proc. 4th international symposium on Information processing in sensor networks, (IPSN '05)*. Piscataway, NJ, USA: IEEE Press, 2005, p. 11.

[137] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks," in *Technical Report in UCLA*, February 2002, CSD-TR 02-0013.

[138] M. Grossglauser and M. Vetterli, "Locating nodes with ease: Last encounter routing for ad hoc networks through mobility diffusion," in *In Proceedings of the IEEE Conference on Computer Communications (INFOCOM), San Francisco, CA, April*, 2003.

[139] I. Abraham, D. Dolev, and D. Malkhi, "Lls: a locality aware location service for mobile ad hoc networks," in *DIALM-POMC '04: Proceedings*

of the 2004 joint workshop on Foundations of mobile computing. New York, NY, USA: ACM Press, 2004, pp. 75–84.

[140] R. Flury and R. Wattenhofer, "Mls: an efficient location service for mobile ad hoc networks," in *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2006, pp. 226–237.

[141] "Nslu2 product data: Network storage link; http://www.linksys.com."

[142] S. Kwon and N. B. Shroff, "Geographic routing in the presence of location errors," *Comput. Networks*, vol. 50, no. 15, pp. 2902–2917, 2006.

[143] A. Mohammed, M. Ould-Khaoua, L. M. Mackenzie, and J. Abdulai, "An adjusted counter-based broadcast scheme for mobile ad hoc networks," in *UKSIM '08: Proceedings of the Tenth International Conference on Computer Modeling and Simulation*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 441–446.

[144] B. Leong, B. Liskov, and R. Morris, "Greedy virtual coordinates for geographic routing," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, October 2007, pp. 71–80.

[145] J. Lian, K. Naik, Y. Liu, and L. Chen, "Virtual surrounding face geocasting with guaranteed message delivery for ad hoc and sensor networks," *Network Protocols, IEEE International Conference on*, vol. 0, pp. 198–207, 2006.

[146] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[147] K. Seada, A. Helmy, and R. Govindan, "Modeling and analyzing the correctness of geographic face routing under realistic conditions," *Ad Hoc Netw.*, vol. 5, no. 6, pp. 855–871, 2007.

[148] H. Tejeda, E. Chávez, J. Sanchez, and P. Ruiz, "A virtual spanner for efficient face routing in multihop wireless networks," in *Personal Wireless Communications*, ser. Lecture Notes in Computer Science, P. Cuenca and L. Orozco-Barbosa, Eds., vol. 4217. Springer Berlin / Heidelberg, 2006, pp. 459–470.

[149] L. Lazos and R. Poovendran, "Serloc: secure range-independent localization for wireless sensor networks," in *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*. New York, NY, USA: ACM, 2004, pp. 21–30.

[150] E. Shi and A. Perrig, "Designing secure sensor networks," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 38–43, 2004.

[151] A. D. Wood, L. Fang, J. A. Stankovic, and T. He, "Sigf: a family of configurable, secure routing protocols for wireless sensor networks," in *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2006, pp. 35–48.

[152] G. Ács, L. Buttyán, and I. Vajda, "Modelling adversaries and security objectives for routing protocols in wireless sensor networks," in *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2006, pp. 49–58.

[153] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *WiSe '03: Proceedings of the 2nd ACM workshop on Wireless security*. New York, NY, USA: ACM, 2003, pp. 1–10.

[154] S. Capkun and J. pierre Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *In Proceedings of INFOCOM*, 2005, pp. 1917–1928.

[155] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *IPSN '05:*

Proceedings of the 4th international symposium on Information processing in sensor networks. Piscataway, NJ, USA: IEEE Press, 2005, p. 12.

[156] A. Boukerche and X. Li, "Atrm: An agent-based trust and reputation management scheme for wireless sensor networks," in *In Proceedings of IEEE Global Telecommunications Conference*, 2005, pp. 1857–1861.

[157] M. Heissenbüttel and T. Braun, "A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks," in *Proc. of the 3rd IEEE Workshop on Applications and Services in Wireless Networks, (ASWN' 03)*, Bern, Switzerland, July 2003, pp. 197–210.

[158] J.-H. Song, V. W. Wong, and V. C. Leung, "A framework of secure location service for position-based ad hoc routing," in *PE-WASUN '04: Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. New York, NY, USA: ACM, 2004, pp. 99–106.

[159] D. Son, A. Helmy, and B. Krishnamachari, "The effect of mobility-induced location errors on geographic routing in mobile ad hoc and sensor networks: Analysis and improvement using mobility prediction," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 233–245, 2004.

[160] P. Ruiz, V. Cabrera, J. Martinez, and F. Ros, "Brave: Beacon-less routing algorithm for vehicular environments," in *Proc. Of Second IEEE International Workshop on Intelligent Vehicular Networks (InVeNet 2010)*, 2010.

[161] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 85–96.

[162] C. Schurgers, V. Tsiatsis, and M. Srivastava, "Stem: Topology management for energy efficient sensor networks," in *Aerospace Conference Proceedings*, vol. 3. IEEE, 2002, pp. 3–1099.

[163] E. Lin, J. Rabaey, and A. Wolisz, "Power-efficient rendez-vous schemes for dense wireless sensor networks," in *Communications, 2004 IEEE International Conference on*, vol. 7. IEEE, 2004, pp. 3769–3776.

[164] M. Mauve, H. Füßler, J. Widmer, and T. Lang, "Position-based multicast routing for mobile ad-hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 53–55, 2003.

[165] J. Sanchez, P. Ruiz, and I. Stojmnenovic, "Gmr: Geographic multicast routing for wireless sensor networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, vol. 1. IEEE, 2006, pp. 20–29.

[166] J. Sanchez, R. Marin-Perez, and P. Ruiz, "Bruma: Beacon-less geographic routing for multicast applications," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*. IEEE, 2009, pp. 522–529.