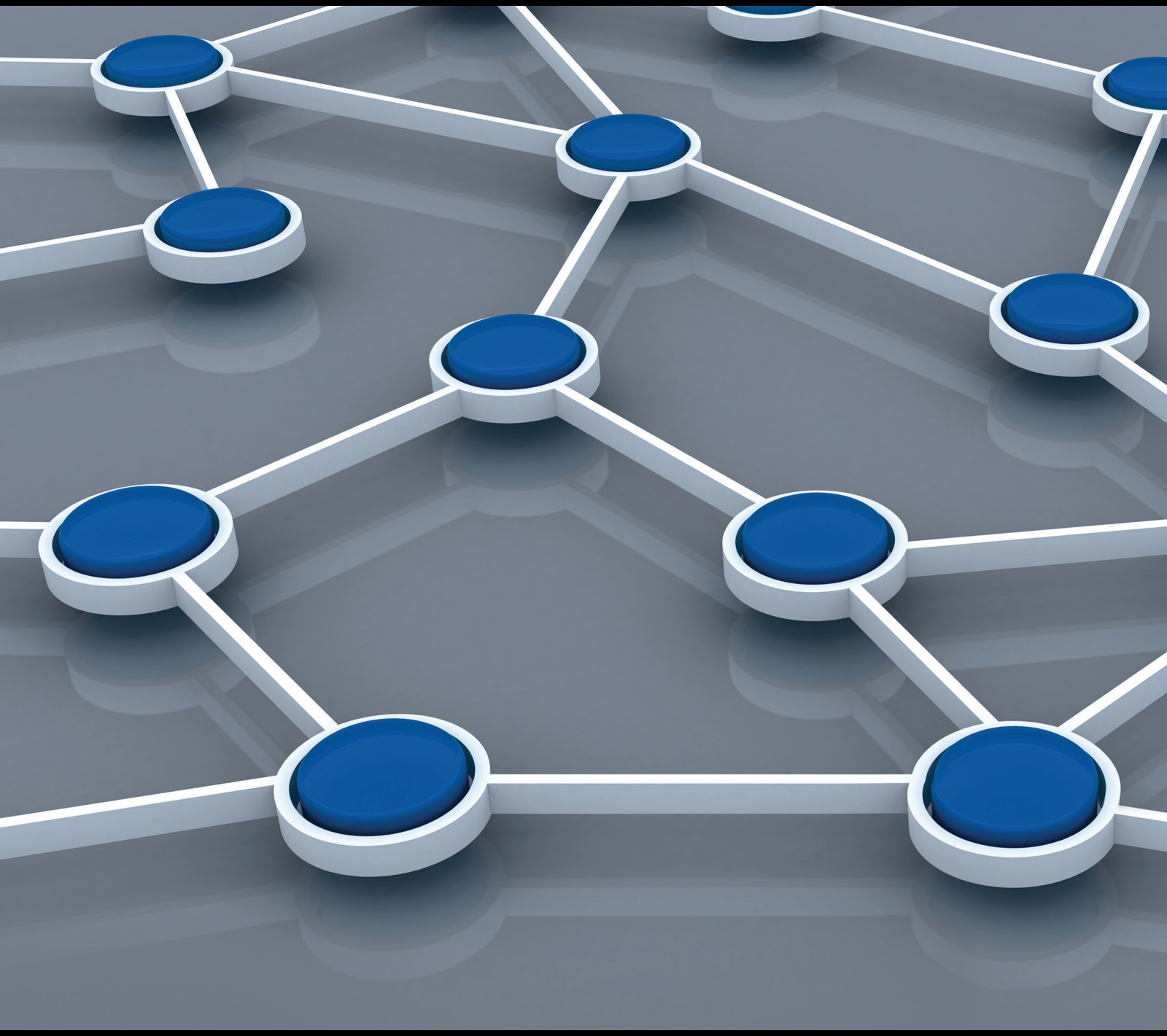
International Journal of Distributed Sensor Networks

Bio-Inspired Mechanisms in Wireless Sensor Networks

Guest Editors: S. Khan, Jaime Lloret, Elsa Macias-López



Bio-Inspired Mechanisms in Wireless Sensor Networks

International Journal of Distributed Sensor Networks

Bio-Inspired Mechanisms in Wireless Sensor Networks

Guest Editors: S. Khan, Jaime Lloret, and Elsa Macias-López

Editorial Board

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 173419, 2 pages
http://dx.doi.org/10.1155/2015/173419

Editorial

Bio-Inspired Mechanisms in Wireless Sensor Networks

S. Khan,[1] Jaime Lloret,[2] and Elsa Macias-López[3]

[1]*Institute of Information Technology, Kohat University of Science and Technology (KUST), Kohat 26000, Pakistan*
[2]*Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universidad Politecnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain*
[3]*Department of Telematics Engineering, Las Palmas de Gran Canaria University, Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain*

Correspondence should be addressed to S. Khan; skkust@hotmail.co.uk

Received 2 December 2014; Accepted 2 December 2014

Wireless sensor networks (WSNs) are gaining significant interest of academia and industry. Multihop wireless networks are self-organizing and self-healing with cost effective deployment and maintenance, yet a lot needs to be done in terms of efficient and robust solutions. WSN has been an interesting area of research in the last years for various disciplines. WSN is the most appropriate choice for different environments monitoring. There are many protocols and algorithms based on natural behaviors that are able to sense data and take decisions.

This special issue is dedicated to consider the technical and theoretical applications and challenges of bio-inspired networking and communication in WSN. The topics in the issue included but were not limited to

(i) adaptive security mechanisms;
(ii) scalable network architecture;
(iii) self-organizing systems;
(iv) self-healing mechanisms;
(v) evolutionary algorithms;
(vi) ant colonization and swarm intelligence techniques;
(vii) artificial neural networks;
(viii) artificial immune methodologies.

We welcomed papers about techniques and applications, awareness, experiences, and best practices as well as future trends and needs related to all aspects of bio-inspired mechanisms in WSN.

The papers have been peer-reviewed and have been selected on the basis of their quality and relevance to the topic of this special issue.

The paper "Systems and Algorithms for Wireless Sensor Networks Based on Animal and Natural Behavior" realizes a survey of the actual systems based on natural behaviors that are currently used for several purposes in WSN. The novelty of this work is that authors recompile systems based on animal behavior and also systems based on other natural behaviors such as plant behavior, bacteria behavior, or immune systems behavior. In this study, authors analyzed the percentage of behaviors in bio-inspired systems, the percentage of bio-inspired main purposes (routing, node location, enhance lifetime, etc.), and even the main purposes of each bio-inspired behavior. Their results show that the bioinspired behaviors which bring more contributions are the animal behaviors (75% of total). Moreover, the ant colony behavior is the most important one between the animal behaviors (35% of the animal based behaviors). The most important function of natural behavior systems is the routing protocol (almost 50% of the total amount), followed by node location (almost 12% of the total amount); meanwhile, coverage optimization or security systems are not so common (less than 3% of the total amount).

Power consumption can increase considerably if data collection is required from all nodes in a WSN. The authors in "In-Network Filtering Schemes for Type-Threshold Function Computation in Wireless Sensor Networks" propose two nature-inspired schemes to forward only relevant data

towards a sink node for processing purposes instead of forwarding all measurements. One scheme is based on the well-known simulated annealing search process and the other one is inspired by the ant colony behavior. Both schemes show significantly communication costs reduction compared to traditional data collection schemes.

The paper "Energy Efficient Routing in Wireless Sensor Networks Based on Fuzzy Ant Colony Optimization" also focuses its contribution on the reduction of power consumption for a WSN but for a different context. Particularly, the authors try to reduce the energy consumption when the nodes dynamically self-organize themselves. They propose a nature-inspired routing protocol for WSN, imitating the foraging behavior of ants. They obtain lower energy consumption amount, lower routing request packets, and a higher network lifetime in comparison with the well-known AODV routing protocol.

WSNs are highly vulnerable to different security attacks. To solve this issue, we have included the paper entitled "Web Spider Defense Technique in Wireless Sensor Networks." In particular, fake wireless sensor nodes are located in the WSN that monitor network and system activities for malicious activities or policy violations. These honeypots attract intruders in order to gather all information about them as possible to report to a central system in order to stop the intrusion.

The authors of the paper "A Framework for Obesity Control Using a Wireless Body Sensor Network" present hardware and software architecture to assist people trying to lose weight. Harnessing the low-cost and low-power consumption small wireless sensor devices to deploy a wireless body area network (WPAN) for the individual, the proposed framework is capable of giving recommendations for the patient by fusing all the data sensed in the WPAN.

We hope that this special issue will be useful for researchers from the academia and the industry, standard developers, policy makers, professionals, and practitioners.

S. Khan
Jaime Lloret
Elsa Macias-López

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 625972, 19 pages
http://dx.doi.org/10.1155/2015/625972

Review Article

Systems and Algorithms for Wireless Sensor Networks Based on Animal and Natural Behavior

Sandra Sendra,[1] Lorena Parra,[1] Jaime Lloret,[1] and Shafiullah Khan[2]

[1] *Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universidad Politécnica de Valencia, C/ Paranimf, n-1, 46730 Grao de Gandia (Valencia), Spain*
[2] *I.I.T., Kohat University of Science and Technology (KUST), Bannu Road, Kohat District, Khyber Pakhtunkhwa, Pakistan*

Correspondence should be addressed to Sandra Sendra; sansenco@posgrado.upv.es

Received 22 April 2014; Accepted 7 July 2014

Academic Editor: Elsa Macias

In last decade, there have been many research works about wireless sensor networks (WSNs) focused on improving the network performance as well as increasing the energy efficiency and communications effectiveness. Many of these new mechanisms have been implemented using the behaviors of certain animals, such as ants, bees, or schools of fish. These systems are called bioinspired systems and are used to improve aspects such as handling large-scale networks, provide dynamic nature, and avoid resource constraints, heterogeneity, unattended operation, or robustness, among many others. Therefore, this paper aims to study bioinspired mechanisms in the field of WSN, providing the concepts of these behavior patterns in which these new approaches are based. The paper will explain existing bioinspired systems in WSNs and analyze their impact on WSNs and their evolution. In addition, we will conduct a comprehensive review of recently proposed bioinspired systems, protocols, and mechanisms. Finally, this paper will try to analyze the applications of each bioinspired mechanism as a function of the imitated animal and the deployed application. Although this research area is considered an area with highly theoretical content, we intend to show the great impact that it is generating from the practical perspective.

1. Introduction

In recent years, researches on wired sensor networks have evolved to wireless infrastructures for implementing wireless sensor networks (WSNs) [1]. A WSN consists of interconnected nodes connected using the air medium to perform distributed sensing tasks. These networks are widely used in agriculture [2], health [3], natural disaster monitoring [4], security and surveillance [5], war ambient [6], and many other fields of interest. WSNs are considered the most appropriate choice in various disciplines for monitoring, sensing, and collaborative decision making. The integration of detection systems, signal processing, and data communication functions converts the WSN in a powerful platform to process the data collected from the environment. The algorithms and protocols for these networks must be able to permit the network operation during the initialization, normal process, and emergency situations. WSNs researchers pursue several requirements, among which we can highlight the following.

(i) Ensure sufficient bandwidth to ensure proper performance.

(ii) Reliability in the network, ensuring a sufficient level of fault tolerance to recover when there is a node failure [7].

(iii) The energy becomes a crucial point because sensor devices are powered by batteries [8, 9].

(iv) The consumption of hardware resources is another of the associated problems especially when it comes to devices intended to be deployed in large spaces monitoring, control, and sensing of intelligent environments, and so forth [10].

(v) The type of used routing protocol is an important aspect to bear in mind because in most cases, it is in

charge of multihop wireless networks in which each node must perform the routing functions of those data from a node to another [11].

(vi) Efficient architectures and topologies to solve any WSNs requirements [12, 13].

(vii) Besides the safety aspects covered by WSNs requirements, it is important to include the availability, integrity, authentication, and confidentiality [14].

These problems and requirements have been widely addressed by the scientific community and there are solutions that have been successfully applied in several application areas. However, recent researches in the WSNs field tend to mimic some biological and animal behaviors which can be easily observed in nature. They are known as bioinspired systems.

Bioinspired systems can be built by hardware or software configurable systems and electronic systems that emulate the way of processing information and problem solving of biological systems. When we look at nature, many types of behavior are observed. Most species of animals show social behaviors. In some species, there is a dominant individual who leads all group members. This is the case of lions, monkeys, and deer. However, there are other types of animals that live in groups with no dominant leaders. In such animals, each individual has an organized behavior that allows them to move through their environment without leader such as birds, fish, bees, ants, or flocks of sheep. In this second type of behavior, animals have no knowledge of the group and the environment in which they move. Instead of this, they move through the environment by sharing information with the closest members. This simple interaction between individuals makes the group behavior more sophisticated [15]. Within imitated bioinspired behavior systems we can also find biological models such as the spread of epidemics or immune systems. The immune system of a human or animal is a complex natural defense mechanism. It has the ability to learn about foreign substances (pathogens) that enter in the body and respond to them by producing antibodies that attack the antigens associated with the disease [16, 17]. These biological behaviors are, in general, the result of millions of years of the nature evolution.

New systems and architectures are tending to include bioinspired systems. The main reasons, among others, are [18] as follows.

(i) They are able to adapt to the medium changes.

(ii) The bioinspired systems exhibit high strength and resistance to failures caused by internal or external factors.

(iii) Allow implementing complex situations and behaviors in a limited set of basic rules.

(iv) These systems are able to learn and evolve as new conditions occur.

(v) They are able to efficiently manage limited resources.

(vi) The set of nodes that implement these systems are able to self-organize in a fully distributed manner achieving an efficient collaboration.

These features give rise to different levels of implementation to the each proposed bioinspired approach, design, and algorithm in every network layer of the WSN, making them more robust, efficient, and resistant to any kind of failure.

We have found in other areas, like MANET, several surveys which include the main bioinspired mechanism [19, 20]. This survey is intended to review the state-of-the-art of the main animal behavior and bioinspired systems used in WSNs. The study will be performed from two points of view. The first one analyzes and explains the main animal and natural behaviors used in the field of WSNs. The second point of view will consider the applications in which these bioinspired mechanisms are employed. Finally, we will statistically analyze the use of each type of application as a function of each mechanism.

The rest of the paper is structured as follows. Section 2 presents and explains the main features of each mechanism as well as the aspects of the imitated natural behavior. Section 3 shows other proposals based on nonanimal behaviors. Using the data presented in Sections 2 and 3, Section 4 will perform an analytical discussion about the use of each mechanism. Finally, conclusion and future works will be shown in Section 5.

2. Animal Behavior Used in WSN

This section presents animal behaviors which are most used in WSNs. They are reviewed from both perspectives, the development of new systems and the improvement of existing systems.

2.1. Ant Colony Optimization (ACO) Algorithm. This section presents the operation of ACO and its principles in which bioinspired systems are based. After that, we will see some examples where ACO has been used with different purposes.

2.1.1. Imitated Natural Behavior Mechanism. In the natural world, initially ants wander randomly. When the food is found, they come back to their colony leaving a trail of pheromones. If other ants find the trail, probably, these ants do not continue walking randomly and follow the pheromone trail. If they eventually find food, ants return and reinforce the trail.

However, over time the pheromone trail starts to evaporate; thus its attraction is reduced. The more time an ant takes to go and come back through the trail, the more time the pheromones will be evaporated. A short path, by comparison, is transited more often, and thus the pheromone density becomes larger in short paths than in longer ones. Pheromone evaporation also has the advantage of avoiding convergence to local optima. If there was no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ants. In that case, the search space of solutions would be limited.

Therefore, when an ant finds a good path between the colony and the food source, the other ants will most probably follow this path and the positive feedback eventually leads all the ants to a single path. The idea of the ant colony algorithm

is to imitate this behavior with "simulated ants" walking through a graph representing the problem. Figure 1 shows the evolution of pheromones that an ant deposits in a way, and how the preferred routes are generated in ant colonies.

The original idea comes from observing the exploitation of food resources among ants, in which ants cognitive abilities are limited individually and together are able to find the shortest path between a food source and their nest or colony [21]. The process runs as follows.

(1) An ant wanders randomly around the colony.

(2) If it locates a food source, it returns to the colony more or less directly, leaving behind a trail of pheromones.

(3) The closest ants will be attracted to these pheromones and new ants adhere to the track more or less directly.

(4) Returning to the colony, ants have strengthened that route.

(5) If there are two paths to reach the same food source, then, in the same given amount of time, the shortest route is traveled by more ants than the longest path.

(6) The shortest path will have increased the amount of pheromones and therefore it will begin to be more attractive.

(7) The longest route will disappear because pheromones are volatile.

(8) Finally, all the ants have determined and chosen the shortest path.

Ants use the environment as a means of communication. They exchanged information indirectly by depositing pheromones on their path, detailing the status of their work. The information exchanged has a local environment. Only an ant located near to the deposited pheromones will know that they are there. This system is called Stigmergy and occurs in many societies of animals. This system is based on the positive feedback (deposition of pheromone attracts other ants and these strengthen such feedback) and negative feedback (route dissipation by evaporation). Theoretically, if the amount of pheromone was the same on all routes at all times, no route was chosen. However, due to feedback, a slight variation on a route will amplify the trail and then, the best path will be chosen.

2.1.2. Works Bon Ant Colony Algorithm. In [22], Camilo et al. presented two ant-based routing algorithms. The first one considerably reduces the size of routing tables, which implies a reduction of the required memory by the node. This proposal also considers the energy levels of the nodes where system gives preference to a longer path with high energy level than a shorter one with lower energy levels. Authors also present the Energy-Efficient Ant based Self-organized Routing (EEABR) which aims to create the best path based on the best pheromone distribution. In this proposal, nodes near the sink present the highest pheromone levels and remote nodes will be forced to find better paths. Their experiments show that EEABR leads to very good results in different WSN scenarios.

T-ANT [23] is a bioinspired approach for data gathering in WSN. This algorithm is based on ant swarm that controls the election of cluster head for obtaining a uniform distribution. Compared with tradition methods such as LEACH, T-ANT needs less memory resource because it can exploit the inherent data correlations in sensed data signals. The results demonstrated that T-ANT reaches significant energy savings for periodic monitoring applications.

An ant-aggregation method for WSN is proposed in [24, 25]. The main aim is to find a solution for the optimal aggregation problem. The method is based on multihop connections and in its operation builds trees with the aim of having the smallest accumulation of cost. In this proposal, the ants have to find the shortest path to the sink or to the nearest aggregation node. Authors compared the simulations of both, the opportunistic aggregation method and the incremental aggregation method. The results show that the energy efficiency in opportunistic aggregation method depends on the number of sources. Simulations show that optimal aggregation method is able to save more than 45% of energy waste in some cases.

The protocol Many-to-One Improved Ant Routing (MO-IAR) is proposed in [25, 26]. MO-IAR is specifically tailored for routing upstream many-to-one sensory data. This protocol operates on two different stages. Firstly, the system finds the shortest path between the nodes and the nest. When those paths are found, it searches the shortest route taking into account the network congestion for minimizing the packet loss. Authors compare their proposal with other related ACO algorithms. MO-IAR presents higher performance in terms of finding the shortest path in the shortest time. The congestion behavior of MO-IAR also presents satisfactory results in comparison with other tested protocols.

Almshreqi et al. [25] and Chen et al. [27] presented other ant system that overcame defects of the conventional routing protocols based on ACO. They adopt a "retry" rule to avoid the deadlock algorithm. The system also adds search ants that are able to reduce the number of "retry." Finally, these proposals introduce a strategy for simulating the pheromone update aimed to accelerate the network convergence. Their simulation results show that this algorithm is able to reduce the total routing cost in WSNs. Their protocols are also more scalable, practicable and efficient than traditional ones.

Saleem et al. [28] and Okdem and Karaboga [29] proposed routing protocols based on ant colony optimization to obtain longer network lifetime, but ensuring, at the same time, that data transmission is efficiently achieved. These proposals discover the shortest paths using an evolutional optimization technique. It provides an effective multipath data transmission. In case that a node fails, the communication inside the WSN remains working. For the implementation, authors used reduced size hardware to solve the space constraints.

Almshreqi et al. [25] and Salehpour et al. [30] show a methodology for cluster-based large scale wireless sensor networks. This proposal works in two stages. Firstly, the system works in intra-cluster level where the sensor nodes transmit their information immediately to the cluster head. Secondly, using the ant-based system, the head clusters discover the

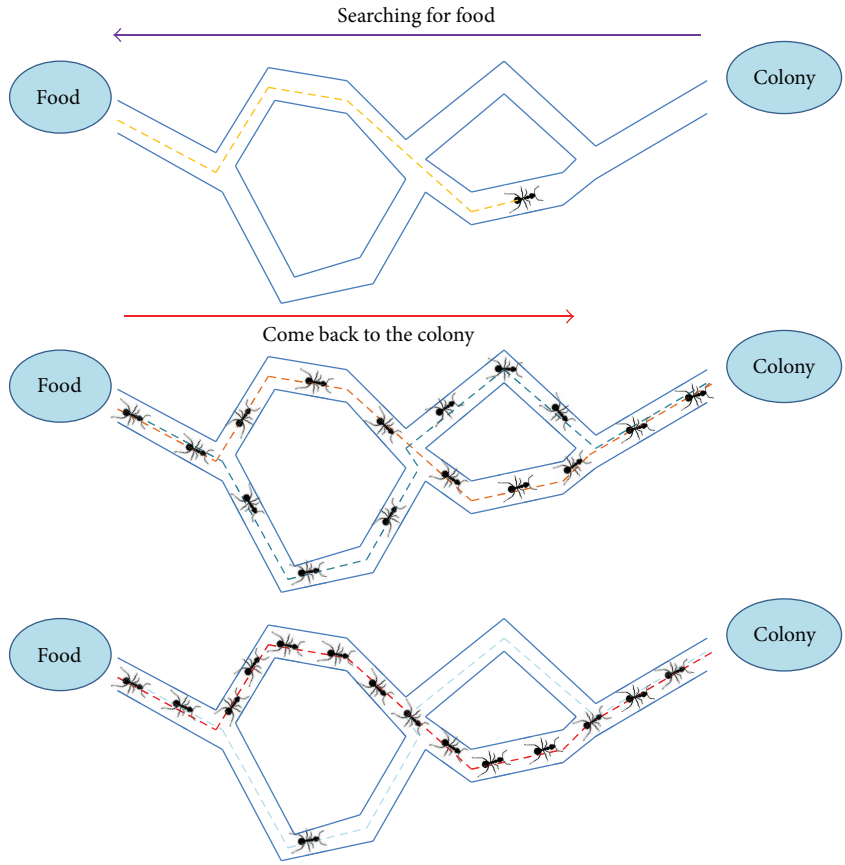


Figure 1: Evolution of pheromones that an ant deposits in a way and how the preferred routes are generated in ant colonies.

route to the sink node. Authors performed a comparison between LEACH and ACO algorithms. Their results show that proposed algorithm present higher network lifetime than the other methodologies. Other simulations between LEACH and the proposal show that proposed algorithm keeps the same level of dissipated energy than LEACH, even changing the number of cluster heads.

SensorAnt [25] uses a new routing scheme to optimize the energy of the sensor nodes. In this scheme, each node compares its information with its neighbors at the beginning of process in order to find the best route to the skin. Authors compare their proposal with EEABR. Their algorithm presents better performance than EEABR in terms of total energy consumption and the energy efficiency. SensorAnt maintains the energy consumption even when WSNs increase the number of sensor nodes. Energy consumption in EEABR increases when the WSN becomes larger.

Wen et al. proposed a dynamic adaptive ant algorithm called E&D ANTS [31]. This algorithm is based on the Energy and Delay metrics for routing operations. Their main objective is to maximize the network lifetime though reducing the propagation delay. In order to achieve these goals, E&D ANTS uses a variation of reinforcement learning. The experimental results show that E&D ANTS presents better performance than AntNet and AndChain. E&D ANTS present communication efficiency up to seven times higher than the AntNet and improve the performance of Ant Chain by more than 150%.

BIOSARP [28] is an Ant Colony inspired Self-Optimized Routing Protocols. It is a specific mechanism that takes into account three variables: speed, link quality and remaining energy in the mechanism. Authors proposed the use of a cross layer design which offers a better delivery radio, maintaining the energy consumption. This issue is especially remarkable in the case of real time traffic. The used algorithm is able to avoid permanent loops. Simulations compared BIOSARP with a real time routing protocol in terms of load distribution. The results show that BIOSARP presents better performance.

Liao et al. used ACO algorithm to solve the deployment problems [32]. The proposed deployment scheme ensures the maximum coverage and reduces the energy consumption. This implies an improvement in the WSN lifetime. The algorithm was tested in five different scenarios with different initial node density and remaining energy. The results show that when node density and remaining energy in nodes increase near to the sink, the system registers higher network lifetime. Furthermore, the proposal presents better performance than the methods previously presented.

Pavai et al. [33] and Juan et al. [34] proposed the use of Minimum Ant-based Data Fusion Tree (MADFT) as a routing algorithm for gathering correlated data in WSN. This algorithm identifies each node as an ant. Firstly, one of these

ants constructs a route. After this, the other ants search the nearest point of this previous discovered route. Using a probabilistic function composed of pheromones and costs, the system can estimate the minimum total cost of the path. MADFT optimizes the transmission and fusion costs. It also adopts an ant colony system to achieve the optimal solution.

Wang and Lin proposed a swarm intelligence optimization based routing algorithm for WSNs [35]. The main goal of this proposal is to accelerate the algorithm convergence rate. The main idea of algorithm is to take less hop numbers into consideration, choosing the nodes with less pheromone as the next hop to avoid prematurely exhaust the energy of some nodes because of having many concentrated routes through those nodes. This will help system to balance the global energy consumption in network. The experiments show that the algorithm proposed in this paper is better than the Directed Diffusion routing protocol in terms of global energy equilibrium.

Jiang et al. [36] proposed a communication protocol called Quantum-inspired Ant-Based Routing (QABR) algorithm for WSNs. QABR combines the operation of quantum-inspired evolutionary algorithms (QEA) and ACO. On the one hand, authors used the concept and principles of quantum computing, such as quantum bit and the superposition of states used in QEA. In QABR algorithm, Q-bit and quantum rotation gate adopted in QEA are introduced into ACO. Their simulations let them conclude that QABR performs better than other conventional routings, such as ACOA and AODV routing. The result is that they obtain a quick algorithm with low convergence time that presents good global search ability.

A new enhanced ant colony inspired self-organized routing mechanism for WSNs is presented in [37]. Saleem et al. focused their efforts on improving the delay, energy and speed in WSNs. The proposal is based on ACO method which is utilized for the optimum route discovery in the WSN. The adopted factors help the WSN in improving the overall data throughput, especially in case of real time traffic, while minimizing the energy consumption. This algorithm is also able to avoid permanent loops. The simulations show that it is an efficient protocol that tries to enhance the sensor network requirements, including energy consumption, success rate and time. Finally its algorithm is improved with reinforcement learning feature to get a superior optimal decision.

Okazaki and Frohlich proposed in [38] a routing protocol based on HOPNET called Ant-based Dynamic Zone Routing Protocol (AD-ZRP). The proposal is a multihop and self-configuring reactive routing approach for WSN. The main aim of this protocol is to reduce the number of control packets sent from the WSN. Simulation results show that AD-ZRP presents better performance than HOPNET. The new methodology has lower routing overhead (because it needs less ants in the network). When studying the performance of AD-ZRP, the authors observed that it presents higher data packet delivery ratio and lower broken routes ratio than HOPNET.

Finally, Hui et al. [39] presented a dynamic and adaptive routing protocol based on ACO. This routing protocol is used to minimize the energy waste thought optimizing the global balance energy in the nodes. For this purpose, the protocol takes into account parameters such as path delay, node energy and the frequency that a node acts as a router to achieve a dynamic and adaptive routing. The tests are performed to check three important aspects: neighbor discovery, routing and data transmission, and route maintenance. Simulations compare this proposal with two popular WSN routing protocols, SPEED protocol and EAR protocol, in order to demonstrate the increased WSN lifetime. The proposal shows better performance in terms of energy consumption levels versus node density and higher node operational time than SPEED and EAR.

2.2. Honey Bee Colony. The Honey Bee Colony behavior is explained in this subsection. This subsection also presents several examples and proposals where this mechanism is used to solve some issues in WSNs.

2.2.1. Imitated Natural Behavior Mechanism. Bees are social insects that live together in large and well-organized family groups. Their high evolution allows them to perform many complex tasks that cannot be performed by solitary insects. Tasks such as communication, construction of the hive, environmental control, defense or the division of tasks are just some of the behaviors that bees perform in social colonies.

A honey bee colony typically consists of three kinds of adult bees: workers, drones, and a queen. Each member has a task to perform, related to its adult age. Surviving and reproducing take the combined efforts of the entire colony. Individual bees cannot survive without the support of the colony. The structure of the colony is maintained by the presence of the queen and workers and depends on an effective communication system. The distribution of chemical pheromones among members and communicative "dances" are responsible for controlling the activities necessary for the colony survival. As the size of the colony increases, so does the efficiency of the colony.

In computer science and operations research, the artificial bee colony algorithm (ABC) is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm, proposed by Karaboga in 2005 [40]. In the ABC model, the colony consists of three groups of bees: employed bees, onlookers and scouts. The algorithm assumes that there is only one employed bee for each food source, that is, the number of employed bees in the colony is equal to the number of food sources around the hive. Figure 2 shows an example of hive.

In this case, employed bees go to a food source in their memory and determine a neighbor source. Then, the bee evaluates its amount nectar and comes back to the hive and dances around the hive's area. The employed bee whose food source has been abandoned becomes a scout and starts to search for finding a new food source. Abandoned food sources are determined and are replaced with the new food sources discovered by the scouts. Onlookers watch the dance of the employed bees and choose one of their sources depending on the dances, and then they go to that source. After choosing a neighbor around that, it evaluates its amount

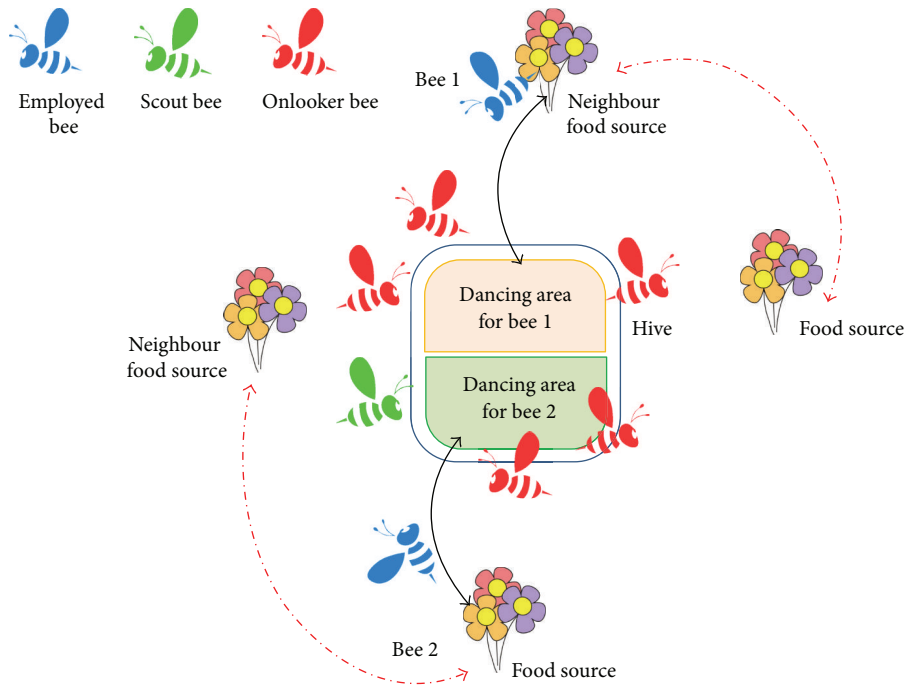


Figure 2: Hive and the kind of bees involved in the process.

nectar. Finally, the best found food source is registered. This process is repeated until the requirements of nectar are met.

2.2.2. Works Based on Bee Honey Colony Algorithm. BeeSensor [39] is a bee-inspired power aware routing protocol for WSN. This protocol has better performance than the traditional ones, like the optimized version of Ad hoc On-demand Distance Vector (AODV). The percentage of delivery packets for BeeSensor is 25% higher than for AODV. In addition, the computational complexity and control overhead are higher in AODV than BeeSensor. Finaly the BeeSensor lifetime is better than the values offered by AODV.

The Pheromone-Signalling-Based Load Balancing Algorithm is presented in [40]. It is based on the process in which the bee queen brings pheromones to the other bees in the hire. When the bees do not feel the pheromone, they assume that the bee queen is death and create a new bee queen. In WSNs, there is a sensor node, or queen node (QN), responsible of managing the execution of all service requests it receives. The other nodes are work nodes (WN). All the nodes (QN and WN) can execute all different tasks, but WN only execute tasks if QN explicitly asks them. Once the QN transmits the "pheromones" to the nearest WN, a WN can transmit it to the other WNs that are far from the QN. The performance of this algorithm allows higher network lifetime (around 10%) than the average time. But, it also presents up to 85% of improvements in service availability in final stages of the system lifetime.

Ismail and Hassan [41] proposed a 6LoWPAN Local Repair Using Bio Inspired Artificial Bee Colony Routing Protocol. The system is aimed to improve the modified AODV for 6LoWPAN to minimize the energy waste and delay with minimum network overhead in a 6LoWPAN mesh network. This proposal is designed with the aim to minimize the new initiation of route request. With all these improvements the network lifetime increases.

BEES [42] is considered as a novel backbone construction protocol for WSN. This protocol is aimed to create bee tiles as identical regular hexagons around the sink node. Its main advantage is that BEES helps to mitigate many of the challenges inherent to WSN such as localization and clustering. It also can simplify many management tasks as data aggregation, leader election, task management, and routing.

BEE-C is a routing algorithm that can reduce the energy consumption in WSN. It was proposed by Da Silva Rego et al. in [43]. BEE-C uses a bionsipered method for clustering the network which is based on the bees' reproduction behavior. It is used to group sensor nodes in order to optimize energy consumption. The experiment results show that the performance of BEE-C presents enhances in comparison to other traditional algorithms like LEACH and LEACH-C. The most important advantages are given in the network lifetime, the low number of sent packets to the base station and in the total network coverage, as a result of using the energy efficiently in the network.

2.3. Bird Flocks and Fish Swarms. This section presents a description of several Bird Flocks and Fish Swarms inspired mechanisms.

2.3.1. Imitated Natural Behavior Mechanism. Swarming behavior is a collective behavior shown by animals of similar size which move together, milling about the same point or

moving in masse. Swarming is usually applied to insects, but this term can also be applied to any other animal that exhibits swarm behavior. The flocking term is often used for referring specifically to swarm behavior in birds, herding refers to swarm behavior in quadrupeds, shoaling or schooling refers to swarm behavior in fish. This subsection explains the generalized swarm behavior for fish swarm and bird flocks. The explanations will be particularized for bird flocks, although these words could be particularized for fish swarms [44, 45].

A flock is a group of birds that present a similar flight or while foraging behavior. The main benefit is the safety of the group that implies an increase of foraging efficiency. The flocks of animals can be formed for specific purposes and the benefit of aggregating is very important in aspects such as defense against predators in closed habitats where predation is often given by ambush. Flocking also has costs to socially subordinate birds which are often bullied by dominant birds. These birds may often be sacrificed in benefit of the rest of the flock.

The basic models of flocking behavior are explained by three simple rules that follow a distributed natural flock behavioral model:

(1) Separation: a bird will turn when another bird gets too close.

(2) Alignment: a bird tends to turn when it is moving in the same direction that nearby birds.

(3) Cohesion: a bird will move towards other nearby birds (unless another bird is too close).

When two birds are too close, the "separation" rule overrides the other two, which are deactivated until the minimum separation is achieved. Each bird always moves forward at the same constant speed. Figure 3 shows these three behaviors.

Each bird acts as an independent object that navigates according to its local perception of the environment, following the physics laws that rule its motion. The bird can perceive other birds at a given distance and at an angle with respect to the direction of its motion, that is, the bird cannot see beyond a certain boundary or beyond a specific angular range. Figure 4 shows this situation.

2.3.2. Works Based on Bird Flock Algorithm. The flock-based congestion control (Flock-CC) is proposed in [46] and with some modifications in [47]. It can be considered as a robust and self-adaptable congestion control mechanism. This mechanism involves a minimal exchange of information and it can increase the WSN lifetime. It also maintains a high quality of service (QoS). The aim of this approach is to guide packets (birds) to create flock and pass (flying) together towards a global actuator, trying to evade the congested regions (obstacles). The performance results show that Flock-CC approach reaches low packet loss, high packet delivery ratio and thus reliability, fault tolerance and low latency. It also outperforms congestion-aware multi-path routing approaches in terms of packet delivery ratio. Antoniou et al. also indicate that Flock-CC has low energy consumption [48].

Ruihua et al. presented a double cluster-heads clustering algorithm based of the particle swarm optimization algorithm [49]. Authors define a new function that takes in account two factors in the cluster-head selection algorithm. The first factor is in regard to the minimum distance between the cluster head and the member node. The second one is the residual energy of the nodes. The system adopts the dual cluster head strategy for balancing the network load. The results indicate that this new algorithm makes the WSN achieve larger network lifetime in comparison with traditional protocols such as LEACH and the combination PSO-LEACH.

In [50], Ma et al. presented the Adaptive Assistant-Aided Clustering protocol for WSN using Niching Particle Swarm Optimization (AAAC-NPSO). Authors define a threshold function to decide the assistant-aided CH in a cluster. The function considers some parameters such as the residual energy, distance between the CH and the base station, and the number of nodes in the cluster. NPSO algorithm is able to reduce the energy consumption and prolong the network lifetime compared to LEACH and PSO-C.

Particle Swarm Optimization is also used to solve the problem of node location after the network deployment [51, 52]. Authors develop a set of simulations to estimate the goodness of this approach. Simulations evaluated some parameters such as the number of localized nodes, computation time, and localization accuracy. The simulation results of PSO, bacterial foraging algorithm, and a traditional method show that PSO is the method that determines the coordinator node in a shorter time.

2.3.3. Works Based on Fish Swarm Algorithm. In [53], authors proposed a hierarchical routing protocol based on Artificial Fish Swarm Optimization (AFSO). This model imitates some fish behaviors such as praying, swarming, following fishes, and so forth. Song et al. used AFSO in the cluster formation phase to solve the NP-hard problem of finding k optimal clusters according to a set of given rules. Traditional protocols, like LEACH, do not offer any guaranties of the position and number of cluster heads. AFSO showed better clusters formation because it dispersed the cluster head nodes throughout the network. Results show that AFSO protocol improves the energy efficiency and prolongs the WSN lifetime thanks to the distribution of the cluster heads.

Neshat et al. [45] and Gao et al. [54] presented a proposal called Improved Artificial Fish Swarm Algorithm (IAFSA). IAFSA included advantages like high convergence speed, flexibility, error tolerance, and high accuracy, while maintaining the behavior of AFSA. The experimental results show that IAFSA presents some advantages such as the fact that it has faster convergence time or higher global search accuracy in respect to the standard AFSA. These enhances make IAFSA advisable for applications in various fields like optimization, control, image processing, data mining, improving neural networks, data networks, scheduling, and signal processing.

Jiang et al. proposed the use of the crossover operation into the Artificial Fish-Swarm (AFA) optimization algorithm

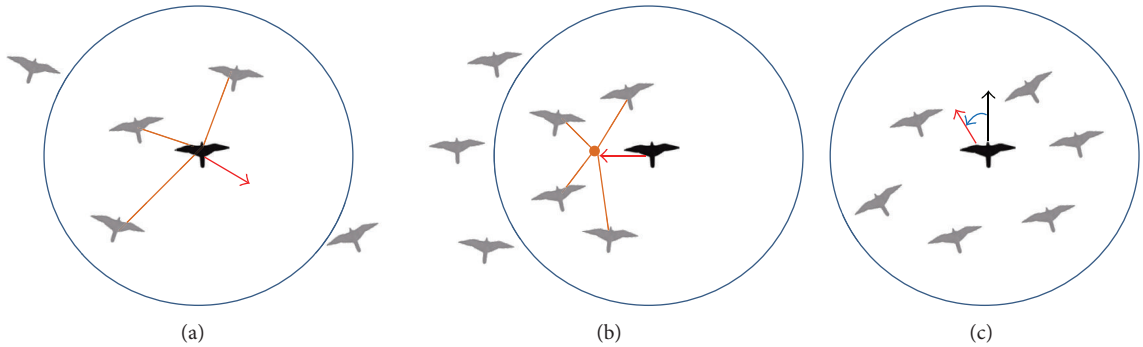


Figure 3: Bird behavior: (a) separation; (b) cohesion; (c) alignment.

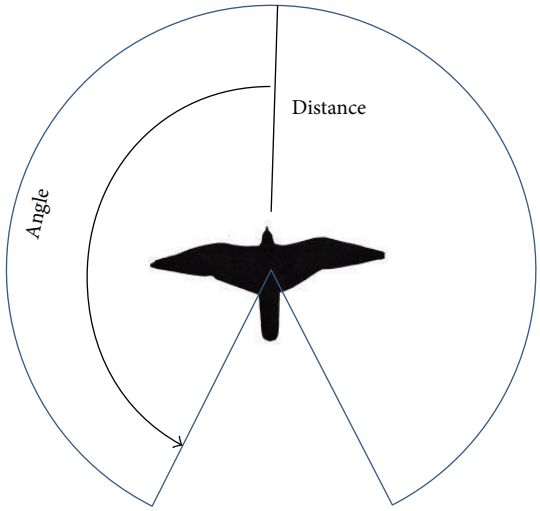


Figure 4: Geometric neighborhood.

and a fusion of the Culture Algorithm (CA) and AFA to enhance the optimization efficiency and combat the blindness of the search of the AFA (CAFAC) [55]. A total of four versions of the CAFAC algorithm are explored. Numerical results show that this new algorithm can outperform the original AFA. The knowledge-based AFA performs much better than the original algorithm. Therefore, the knowledge in the cultural framework can be viewed as an effective and directive term in the evolution of the AFA.

Wang and Ma presented a hybrid artificial fish swarm algorithm, which is combined with CF approximation algorithm and Artificial Fish Swarm Algorithm to solve the packing problem [56]. Experiment results show that when it is compared with GA, the Hybrid Artificial Fish Swarm Algorithm has a good performance with broad and prosperous application, enhancing aspects such as computing speed and accuracy.

The algorithm presented by Fernandes et al. in [57] is a modified version of the artificial fish swarm algorithm for global optimization, denoted by Fish Swarm Intelligent (FSI) algorithm. The main modifications of this algorithm are the following ones: an extension to bound constrained problems meaning that any fish movement will be maintained inside the bounds along the iterative process; modified procedures to translate random, searching, and leaping fish behaviors; the introduction of a selective procedure and different termination conditions. The results show that FSI improves some WSN issues that other existing algorithms are not able to solve.

Yiyue et al. presented a deployment optimization scheme for WSNs which is composed of fixed sensor nodes and some mobile sensor nodes [58]. This proposal is also based on the Optimized Artificial Fish Swarm Algorithm (OAFSA). In this case, authors introduced a modification in OAFSA for including a dynamic threshold to improve the coverage problems of AFSA. Simulations compared OAFSA and traditional ASFA. The results show that the new approach presents better performance. In fact, OASFA is able to increase the network coverage. However, the convergence speed of AFSA is higher than the OAFSA so; there is no improvement in the convergence speed.

2.4. Bat Behavior. Most species of bats rely on echolocation to find their prey. For this reason, they have no problem to find a prey in dark environments. Calls from the bat can reach up to 130 decibels and it is considered as the highest of all flying animals in the world [59].

The echolocation process is very complex and it has been studied in detail by several researchers [60]. Bats are able to differentiate between incoming and outgoing signals and it is the way they can differentiate between sent and received communication. Bats use sonar echoes to detect and avoid obstacles. It is generally known that the sound pulses are transformed to the frequency that is reflected from an obstacle. Bats use delay time from transmission to the reflection for navigation. After hitting and reflecting on the object, bats transform their own pulse to use this information to measure how far away the prey is. Figure 5 shows a bat emitting and receiving a wave to detect an object and the perceived wave by the animal for each ear.

This behavior can be modeled using the following three general rules [61].

(1) All bats use echolocation to sense the distance, and they also guess the difference between food/prey and background barriers in an inexplicable way.

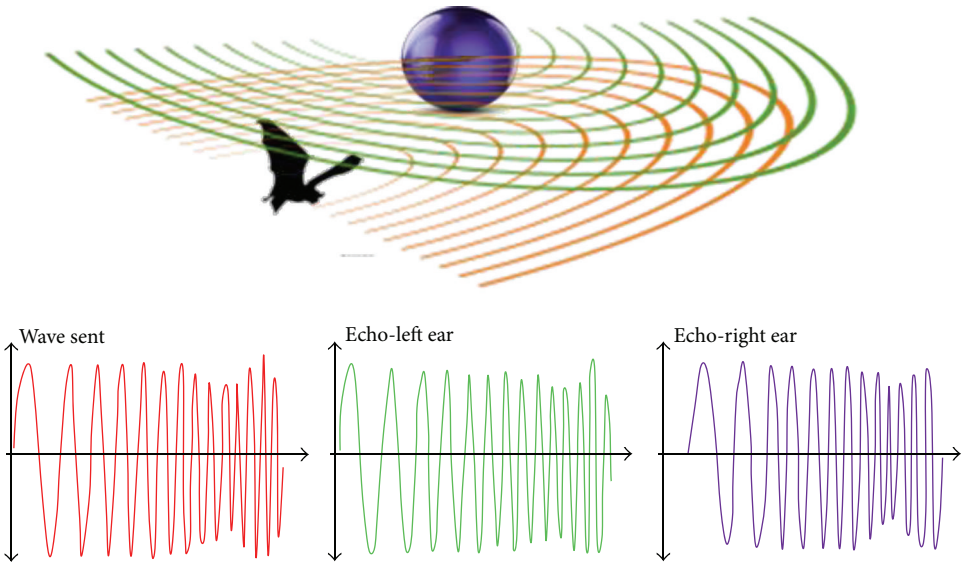


Figure 5: A bat emitting and receiving a wave to detect an object and the perceived wave that the animal has for each ear.

(2) Bats fly randomly and they fix a minimum frequency, varying wavelength, and loudness to search for a prey. They can automatically adjust the frequency of their emitted pulses depending on the proximity of their target.

(3) Although the loudness can vary in many ways, we should assume that the loudness varies from a high to a minimum constant value.

This algorithm is mainly used for node localization in WSNs. There are very few works based on bat algorithm. Some of them are the next ones.

Goyal and Patterh addressed in [62] the problem of nodes localization in WSN. They proposed to use the Bat Algorithm (BA) to solve this problem. Authors performed a comparative simulation between BA and the Biogeography-Based Optimization (BBO) algorithm. Simulations were performed considering 40 target nodes. 20% of these anchor nodes were randomly deployed in a field of 100×100 square units. Each anchor node has a transmission range of 30 units. The results show that BA performs better than the BBO. BA also gives fast convergence time and better accuracy than BBO.

Goyal and Patterh also studied how the mean location error was affected by parameters as field size, transmission range, anchor nodes, and sensor nodes [63]. After a set of simulations, the results show that the localization mean error decreases when the number of anchors and transmission range increase. This value also increases when the number of sensor nodes and field size gets higher.

2.5. Fireflies Behavior. There are about two thousand firefly species and most of them produce short and rhythmic flashes. The pattern of flashes is often unique for a particular species. The flashing light is produced by a process of bioluminescence. There are many theories regarding to the importance of flashing light in firefly's life cycle but many of them converge to mating phase [64], that is, to communicate, to attract a prey, and as protective warning mechanism [65].

The light intensity at a given distance from the light source follows the inverse square law. Air also acts as absorbent and light gets weaker when the distance increases [66]. Combining these two factors the visibility of fireflies is reduced to a limited distance. But, it is sufficient for fireflies to communicate between them at distances of few hundred meters at night.

The flashing light of fireflies can be modeled to formulate new optimization algorithms for WSNs. Firefly algorithm is based on the flashing characteristic of fireflies. It follows three idealized rules.

(1) All fireflies are considered as unisex and a firefly will be attracted to other fireflies regardless of their sex.

(2) Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness, that is, for a couple of individuals, the less bright firefly will move towards the brighter one. But if no one is brighter than other, they will fly randomly. In addition, the brightness of both will decrease as their distance increases.

(3) The brightness of a firefly depends on the aim function [67].

The main application of this mechanism is to find the optimized solution for a given problem. Because of its operation principles, this animal behavior is not widely used. Now some examples are presented.

Liao et al. proposed a sensor deployment scheme based on glowworm swarm optimization (GSO) [68]. This proposal can increase the coverage of the sensor with limited movement after the initial position. Its advantage is that this system does not need a centralized control, so it is easy to scale it for large WSNs. The simulation results show that

GSO algorithm (compared to nonbioinspired mechanism) has higher coverage rate with limited sensor movement.

Sun et al. proposed a clustering scheme based on firefly behavior [69]. The scheme is called Clustering Firefly Synchronization Algorithm (CFSA) and it is an improvement of the Reach back Firefly Algorithm (RFA). The proposed scheme works in different steps. Firstly, each cluster achieves its own intracluster synchronization. Secondly, all clusters achieve the intercluster synchronization. After some comparative simulations, they saw that CFSA outperforms the RFA.

2.6. Other Less Used Animal Behaviors. There are some other animals' behaviors which are less used such as the ones presented in this subsection. However, they propose very interesting approaches.

2.6.1. Termites. Termites are beings of small size and small number of neurons. They are not able to perform complex tasks individually. However, termite colony is seen as an intelligent entity because of their great level of self-organization. This allows them to perform very complex tasks. Based on the termites' behavior, Zungeru et al. [70] performed a comparative study between three routing algorithms SC, FF, AODV, and Termite-hill algorithm. The analogy in regards to termite behavior, each node serves as a router and as a source, and the hill is a specialized node called sink which can be one or more nodes as a function of the network size. In addition, each network node can also serve as a termite-hill. The basic operation principles are the following ones. Termite-hill discovers routes only when they are required. When a node has some events or data to be relayed to a sink node, and it does not have the valid routing table entry, it generates a forward soldier and broadcasts it to all its neighbors. Authors tested the performance of the algorithm on static and dynamic sink scenarios. The results show that Termite-hill algorithm is scalable and presents lower network energy consumption.

2.6.2. Elephants. Elephants are the biggest land mammals. They live in groups. These groups can raise more than 50 individuals, so they need a well-organized structure and good communication. Some of the quotidian actions developed by the group are teamwork, offspring care, group defense, and resource acquisition. All these decisions are made by the oldest female of the group [71].

Desai et al. presented a new approach to enhance the WSN lifetime [72]. This proposal is based on the Elephant Based Swarm Optimization model. Authors present a cross layer (routing, MAC, and radio layers) approach which is compared with LEACH protocol in terms of efficiency. Simulation results show that the proposed elephant swarm optimization technique presents better performance than LEACH. The proposed model improves the network lifetime, reduces the sensor node energy decay rate, and presents lower communication overheads and higher active node ratios.

Other proposals based on Elephant Swarm Optimization model is presented in [71, 73]. Their aim also is to increase the network lifetime. As in [72], these proposals use the elephant Swarm Optimization mode to create clusters. In both proposals, there is collaborative processing to enhance the data reliability during aggregation. Authors performed several simulations comparing their proposal with the ACO model. Their results show that the data reliability of the WSN increases when Elephant Swarm Optimization model is used.

2.6.3. Monkeys. Rhesus macaques live in large multimale and female group. Females in a group very rarely move away from their groups, but the males often wander away in search of mating opportunities after attaining their puberty. Females are arranged according to their matrilineal family relationship. If that monkey dies, automatically the rank is passed to the next monkey in the hierarchy. The less dominant monkeys' are involved in intergroup communication for giving alert calls. Following this model, Kumar and Kusuma [74] proposed a hybrid methodology based on the monkeys' behavior to improve the traditional LEACH protocol. Simulations were performed in terms of number of dead nodes and energy consumption. The results show that this proposal presents lower energy waste than LEACH regarding the set-up phase of LEACH. The new protocol presents good scalability and it can be easily adapted for its use in WSNs.

2.6.4. Hybrid Algorithms. Breza and McCann [75] proposed the combination of two different bioinspired algorithms. They tested two combinations: flocking-fireflies and firefly-gossip. The results show that in both cases the performance of the hybrid method is the same or even better than each method alone. In flocking-fireflies combination, the synchronization time was shorter than the firefly algorithm. This is because the flocking algorithm increases the number of neighbor population of any given agent. Authors concluded that there could be certain combinations of bioinspired algorithms that may have negative effects, generating lower performance.

The hybrid algorithm based on fish and particle swarm algorithm is presented in [76]. This combination is used to solve the coverage problem. The system tries to organize the sensor nodes in order to obtain the maximum coverage for improving the performance. The combination of AFSA and particle swarm optimization (PSO) allowed it to acquire the global search capacity (from AFSA) and the rapid search ability (from PSO). The simulation results show that this new algorithm can optimize the deployment of the sensors and improve the coverage.

A combination of PSO with Voronoi Diagram is presented by Aziz et al. in [77]. This method can be used to optimize the coverage problem in WSN. Authors implemented the algorithm using MATLAB with different specifications of the WSN. They studied the effects on the algorithm due to changes in the number of sensors and the size of the region. Their results show that the proposed algorithm presents better performance with high number of sensors and small region of interest. Results are also promising for low number of sensors and large region of interest.

Sun and Tian proposed in [78] a new hybrid method for route optimization in WSNs. This method is based on

a combination of modified ACO algorithm with GA. Hybrid method is used to reduce the complexity and enhances greatly its efficiency. Moreover, authors added a multipath route for transmitting data. After simulations, the results display that the new method is effective and has a better performance than ACO and GA working separately.

Falcon et al. proposed a kind of cooperative networking system in which a small team of robotic agents are placed in a base station [79]. The mission of those robotic agents is to serve an already-deployed WSN. Robotic agents perform a periodic replacing of all the damaged sensors in the field to preserve the existing WSN coverage. Authors called this novel application as multiple-carrier coverage repair (MC2R). The replacement trajectories followed by the robotic fleet are originated by a hybrid algorithm in a short running time. That hybrid algorithm uses the swarm of artificial firefly algorithm and the exploratory principles featured by Harmony search. Results show that promising solutions can be obtained in a limited time span.

2.6.5. Predator-Prey Behavior. The Lodka-Volterra model is based on nonlinear equations that predict the behavior of a predator-prey system. The variation of population size as a function of the time can be modeled as a simple balance equation. This equation predicts the changes on the population size and it depends on the interaction with resources, competitors, mutualisms, and natural enemies. The Lodka-Volterra model is based on a deterministic competition model which involves two coexisting species. In this case, the fitness of one individual is negatively influenced by the presence of the individual specie. Those individuals can be of the same specie (intraspecific competition) or from different species (interspecific competition) [80]. According to [80, 81], a WSN can be compared to an ecosystem where the nodes are grouped in different clusters. Authors proposed to use this behavior to create a congestion control mechanism. Lodka-Volterra model is able to preserve the global properties of biological processes, that is, stability, self-adaptation, scalability and fairness. Lodka-Volterra model shows that it can overcome traditional protocols like Additive Increase Multiplicative Decrease.

2.6.6. Spiders. The social spider of Congo presents a very impacting behavior when hunting its prey. These spiders live in groups which can be greater than 5000 individuals in the same spiderweb. They start the predatory behavior "dancing" on the spiderweb altogether. After certain time, all of them stop moving and detect the vibrations in the spiderweb. If a vibration is detected, it means that there is a prey in the spiderweb and all spiders move together to hunt it. The preys do not usually have any opportunity to avoid the attack. In this behavior, there is no direct communication between individuals. The silk actuates as a vector of information. Vibration goes through the silk and transmits the information from the prey. From this information, the number of spiders required for that prey is determined [82].

This behavior was used by Benahmed et al. [82] to propose a mechanism which is able to detect and delete misbehaving sensor nodes in WSNs. Like spiders, monitor nodes at each time period enter on a very short listening time. If any other node transmits in this period, it is considered as an abnormal behavior and the identity of this node is archived. If this pattern is detected in this node more times, the communications with this node will be isolated form the rest of the WSN and it will not participate in more network activity. Their simulation results show that in this new methodology, monitor nodes can detect the presence of intruder nodes. This detection probability is higher in the simulations with higher monitor nodes density.

Table 1: Resume of animal bioinspired mechanisms.

Bioinspired mechanism	Quantity of works based on them
Ant colony	18
Bee honey colony	5
Bird Flock	5
Fish Swarm	6
Bat	2
Fireflies	2
Termites	1
Elephant	3
Monkey	1
Hybrid animal behavior	5
Predator-prey behavior	2
Spider	1

2.7. Summary. In this section we summarize the number of works found of each revised bio-inspired mechanism. They are shown in Table 1. The most used animal inspired mechanism is the ant colony. It was the first to appear and is more widespread than the others. In the lasts years new ones like monkey behavior or elephant behavior have started to be in use.

3. Other Biological Behaviors Used in WSN

3.1. Genetic Algorithm (GA). In the 1970s, John Henry Holland proposed a new idea within artificial intelligence, genetic algorithms (GA). They are so named because they are inspired by biological evolution and molecular-genetic basis. A GA is a directed search method based on probability. Under a very weak condition (that the algorithm keeps elitism, that is, always save the best element of the population without making any changes) it can be shown that the algorithm converges in probability to the optimum. That is, by increasing the number of iterations, the probability of the optimum in the population tends to 1. In recent years, GAs are being used in many areas, including advanced security mechanisms [83]. GA is based on the process that drives biological evolution, the natural selection. It is based on a series of individual solutions of a population that is modified as a function of the time. At each step, one of the individuals is selected randomly from the current population. This individual becomes a parent that produces the children

for the next population. After some steps, the population starts to evolve towards an optimal solution [84].

Hussain et al. proposed in [85, 86] the use of a genetic algorithm to create energy efficient clusters for data dissemination in WSNs. After simulations, authors concluded that GA has better performance than traditional clustering protocols as LEACH or hierarchical cluster-based routing (HCR). GA also uses a cross layer optimization, so the energy consumption during the reconfiguration is minimal. Furthermore, this algorithm works trying to adapt itself to the energy levels in nodes; other cluster-based protocols do not use this technique. The proposed algorithm is able to increase the network lifetime.

Ferentinos and Tsiligiridis presented an algorithm for multiobjective optimization such as optimal design, dynamic adaptation, and energy management [87]. This proposal is based on the evolutionary optimizations of GA. The algorithm has sophisticated characteristics that make it able to decide about the sensors' activity/inactivity schedule as well as the rotation of the CH. It also manages different sensors roles according to their wireless signal. All these improvements contribute to increase the network lifetime in the WSN.

Other approaches based on GA are presented in [84]. In this case, Bhondekar et al. use a GA for node placement. Sensors are spread and the GA decides which sensors must be active or inactive and which one will be the CH. It also decides if the normal nodes (no CH) should have medium or low transmission range. Results show that GA-generated design presents better performance than random designs. The uniformity of sensing points of optimal designs was satisfactory, operational communication, and energy consumption were minimized while maintaining the connectivity constraints.

The energy-efficient Coverage Control Algorithm (ECCA) is presented by Jia et al. in [88]. This proposal is inspired in multiobjective genetic algorithms (MOGAs). The mechanism is used in the data gathering inside the WSN. Its goal is to activate the lowest possible number of nodes in a densely deployed environment. The protocol presents two restrictions. The first one is in regards to coverage rate of the WSN. The other restriction is the number of the chosen nodes from the whole network. ECCA offers several important advantages such as negligible computation time and one-time resetting of the working state of the sensor nodes. It can also use the desired sensor field coverage and model parameters as inputs, so it has great flexibility.

In [89] authors presented a solution, based on GA paradigm, for the problem of placing multiple sinks in a time-sensitive WSN. It is called Genetic Algorithm-based heuristic for sink placement (GASP). Authors developed a series of tests aimed to compare the solution based on GA with nonbioinspired methods. Their results show that the performance of GASP is better than pure random search. GASP has more favorable behavior in respect to the quality of the solutions found and the computational effort invested, especially in large-scale networks.

Another application based on GA is presented in [90]. In this case, the algorithm is used to solve the problem of knowing the location of the sensor nodes in WSN. This information is essential in many tasks such as routing, service delivery, or cooperative sensing. Traditional methods do not bring an accurate solution. The presented algorithms were used in a simulation on a WSN with fixed number of nodes whose distance measurements were corrupted by Gaussian noise. Their results show that the proposal was able to give an accurate location of the nodes.

3.2. Immune Systems. The immune system that protects the body form the attacks of pathogens is a complex system that self-adapts to different situations. Artificial immune systems are based on functions and algorithms that mimic the behavior and properties of immunologic cells. There are two main procedures, that is, pathogens detection and pathogens elimination. In order to detect pathogens, the system must be able to distinguish between self and nonself cells. In human body, this task is performed by lymphocyte cells. When a detector finds a nonself-cell with a different mechanism, depending on the harmfulness of this cell, it can be triggered to delete that cell [91].

In [92], authors proposed a biological inspired secure autonomous routing mechanism named BIOSARP. This routing mechanism is based on ACO and a self-security mechanism based on artificial immunity system (AIS). It is able to detect the nonself-antigens (most commonly known attacks). If the system finds an abnormality, it will immediately generate a decision agent. Once the intruder alarm is generated by the monitoring agents, the security management will implement the authentication process. The system provides the security with no additional cost (control, energy waste, and computational cost), so it has a higher performance.

A new bioinspired technique for autonomous plausibility checking and data processing for WSNs was proposed by Jabbari and Lang in [93]. This methodology consists of two stages. In the first one a Neuroimmune system is introduced and developed, which is used to predict the sensor records. The second algorithm is used to evaluate the sensor records to check the plausibility of the records in the WSN. The performance of the developed technique presents a more correct data approximation than the sliding back propagation technique.

Teng et al. presented in [91] an immune inspired system which is used to locally discover and recover from losses of query messages. Authors propose the use of a cluster of loss detectors for each sensor node which cooperate in a distributed and scalable way. Detectors cooperate to locally recover from query losses, similar to antibodies in an immune system. The results show that this proposal is energy-efficient and scalable to operate in a fully distributed manner.

3.3. Bacteria. Bacterial foraging algorithm is based on the behavior of *Escherichia coli* (*E. coli*). It is a bacterium that lives in some mammals' intestine. *E. coli* uses a pattern of two different moving types: tumbling and swimming and uses them to search the rich-nutrient areas. When *E. coli* is in neutral medium, it alternates both types of moving. When *E. coli* is moving to the direction of a rich-nutrient area, it uses

a swimming direction. However, when swimming movement raises a low-nutrient area, a tumble movement appears and the swimming direction changes. After this tumble and few swimming steps, called chemotactic round, the bacterium finally raises a rich-nutrient location. The bacterium that raises these areas can be split into two new bacteria and the others that do not reach that area die. The bacteria reproduction process is explained in [51].

In [51, 52], a bacterial foraging algorithm is used to solve the problem of node location after the deployment. Simulations were performed for estimating the accuracy of this methodology. Simulations evaluated the number of nodes localized, the computation time, and the localization accuracy. During the PSO simulations, the bacterial foraging algorithm and a traditional method were compared. Bacterial foraging algorithm was the method that locates the sensor nodes with more accuracy.

Dhiman presented a new routing protocol in [94]. It is a hybrid protocol, partially based on bacteria foraging optimization. The Bioinspired Hybrid Routing Protocol (BIHP) uses de BFO technique in the cluster head selection phase and the hybrid protocol for achieving the improvements. Simulations compared BIHP and LEACH. BIHP is able to improve the energy efficiency of the WSN up to 35% better than LEACH. BIHP also helps to balance the energy consumption and improves the stability period for the WSN.

On the other hand, Sribala and Virudhunagar presented a modification of BFA [95] called Modified Bacterial Foraging Algorithm (MBFA). BFA is used in routing tasks to enhance the nodes' lifetime. Meanwhile MBFA can be used for large scale optimization problems. Authors performed comparative simulations with BFA, MBFA, and LEACH. The controlled parameters were data transmission, energy waste, and the number of alive nodes. The results show that MBFA presents better performance than LEACH and BFA. BFSA is able to increase more nodes' lifetime than the other analyzed protocols.

3.4. Artificial Plant Optimization Algorithm (APOA). In natural environments, plants survivals depend on their capacity to sense the environment and data storage. Plants use this information to decide their next movements, that is, the new areas that are going to be colonized. Plants are continuously sensing biotic environment, but they can also do it actively. Some of the sensed parameters are light and gravity [96].

In [96] authors propose a biologically inspired (Botany) mobile agent based self-healing wireless sensor network (BIMAS). It was inspired on the concept of adopting the mobility of a wireless sensor node as a potential solution for handling the energy utilization of the wireless sensor network. The use of those mobile agents for sending sensed data to the base station can help to maintain the energy levels at sensor nodes, especially in the nodes close to the base station. If those nodes lose their energy the network may fail but BIMAS can efficiently handle this problem.

Artificial plant optimization algorithm (APOA) is a new evolutionary computation inspired by plant growing process presented by Li et al. [97]. APOA is used to solve the coverage optimization problem. APOA defines three operators: a photosynthesis operator, a phototropism operator, and apical dominance operator. Phototropism operator has the most important role for the grow direction of the branch. The phototropism operator is developed to increase the search efficiency. After simulations, the results show that APOA presents better node distribution than other algorithms.

4. Analytical Study about the Use of Bioinspired Systems

After reviewing the main models of animal and natural behavior used in the development of new improved systems for WSNs, we can perform a statistical analysis. This study can be performed from different viewpoints. On the one hand, we can analyze the type of mechanism used, that is, animals or other natural behaviors and the type of behavior that they imitate. On the other hand, it is important to know the type of application in which these systems and proposals are intended. Of course, we cannot speak in absolute values on the use of each type of mechanism. However, we can see these data as relative values to show us the tendency of researchers within this field. This section discusses and compares these data.

The first classification shows the trend of using each type of system. A relevant fact is that most bioinspired systems proposals for WSNs are based on the animal behaviors (75.00%). This is because each animal (its analogy in WSN is the sensor node) is considered as an individual which is able to perform a very limited series of tasks. However, the set of all the animals can develop very complex actions (in our case it would be the entire network of nodes) for the benefit of the whole group. In the WSN world, this benefit is translated into an improvement on energy consumption and improvement on the routing information, and so forth. The second most used method (with a value of 16.18%) is the imitation of physiological functions. This group is based on the interaction of the basic elements of a living being with its environment systems. To perform such functions, it is necessary to know both the particles as a whole organism and the environment.

The behavior of plants is the third most common mechanism (with a value of 5.88%) and, finally, bacteria behavior is the fewest employed (nearly to 3%). Figure 6 shows the measurements obtained about the percentage of mechanisms used in bioinspired proposals for WSNs.

Regarding animal behavior, bioinspired systems can imitate tens of animals. However, only some of them are the most used systems (see Figure 7). The most used mechanisms are those which are based on a colony, flock, or swarm of animals. Ant colony is the most widely imitated mechanism (more than 35%). After it, bird flock, fish swarm, bee honey colony, and hybrid animal behavior present similar percentages of use (around 9–12%). With a percentage of use lower than 5%, we can find some animal behavior which is in natural habitat, although they live in society, their behavior does not imply the collaboration of several individuals. Finally, we highlight the predator-prey behavior because it is not a behavior between animals of the same society.

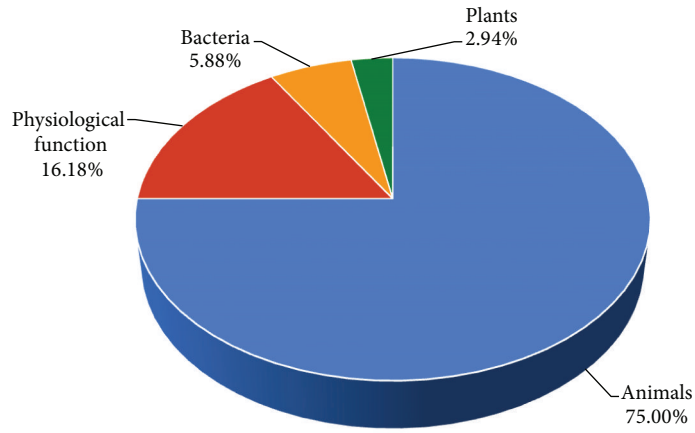


Figure 6: Percentage of mechanisms used in bioinspired proposals for WSNs.

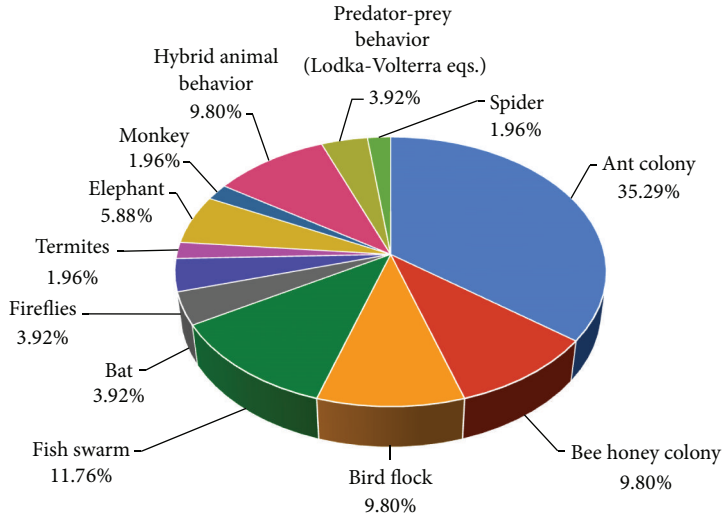


Figure 7: Bioinspired mechanism based on animals' behavior.

Within the bioinspired systems, we have also found proposals which are not based on animals. This group includes the imitation of immune systems (17.65%), bacteria behavior (23.53%), and genetic algorithms (47.06%). Plant based bioinspired systems are used in 11.76% of cases. Figure 8 shows the percentage of use of nonanimals bioinspired systems.

It is important to analyze the main utilities developed and proposed for improving the efficiency of several issues in regards to WSN which use these animal and natural behaviors. As Figure 9 shows, almost 50% of new proposals are related to routing protocols. We can see that researchers are also interested in improving energy consumption (around 16%) and node localization in WSN (almost 12%). Other important issues that bioinspired systems solve are congestion control, fault tolerance, and coverage optimization, among others. Finally, we can see that there are very few bioinspired systems useful for improving security, coverage, and data collection systems in WSN. Each one represents a percentage of 2.94%.

Finally, we are going to analyze the uses of each bioinspired mechanism and whether a behavior can be used in different applications. As Figure 10 shows, most behaviors are used to solve more than one problem. However, bat, termites, elephant, spider, and predatory-prey algorithms are used to solve only one problem. The bioinspired mechanism with more utilities is the genetic algorithm with 5 different applications (routing protocol, node location, enhance energy efficient, data collection systems, and security systems). Honey bee colony, bird flock, and fish swarm behaviors have 4 different uses. Ant colony, immune systems, and bacteria behavior have 3 different purposes. The rest of bioinspired mechanisms have two different applications. Approaches in routing protocols are present in almost all behaviors (except in bat, elephant, spider, and predatory-prey behaviours). On the other hand, uses like fault tolerance,

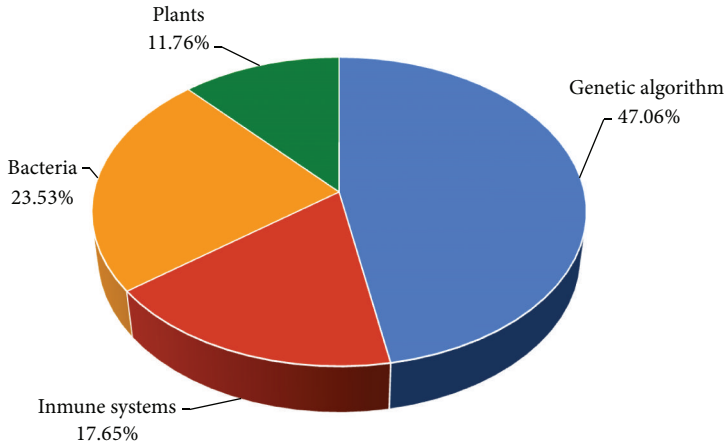


Figure 8: Bioinspired mechanism based on nonanimals behavior.

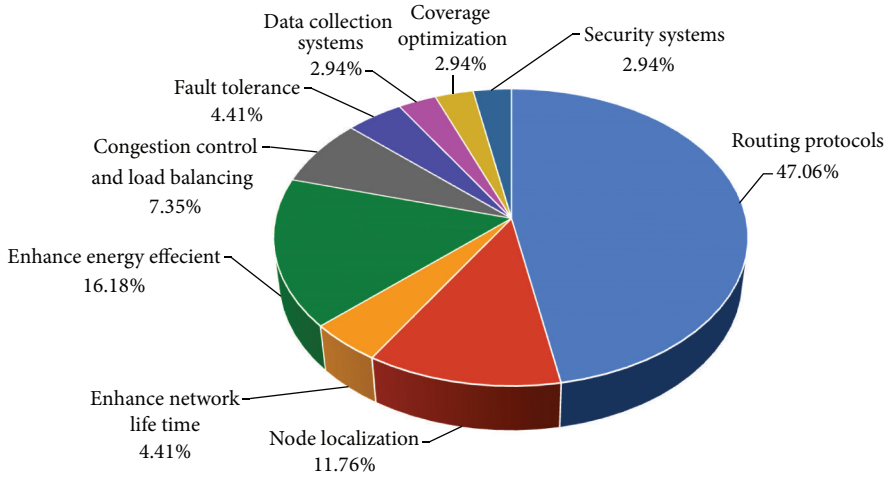


Figure 9: Main purposes of bi-inspired systems in WSN.

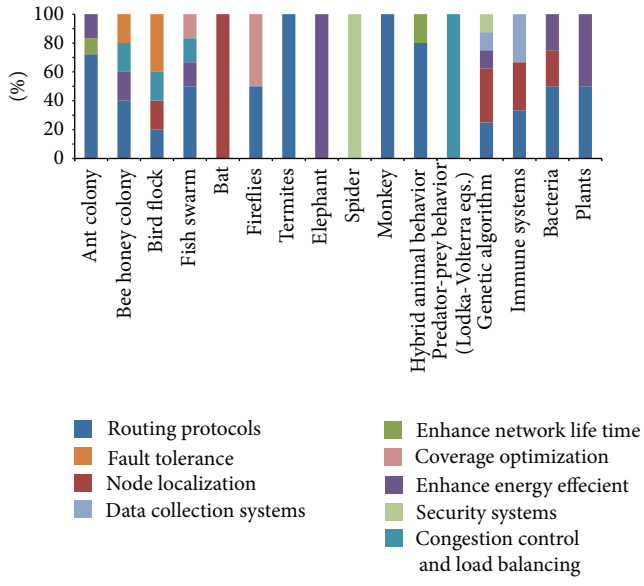


Figure 10: Different purposes for each bioinspired mechanism.

coverage optimization, or security systems are implemented by less behaviors (only two each one).

5. Conclusion and Future Works

Systems based on animals and natural behaviors are being used to improve and solve several issues in WSNs. Because of the interest of researchers in this topic, in this paper, we have analyzed the state of the art of bioinspired systems focused on WSNs' issues. We have explained the most well-known animals behaviors used in bioinspired mechanisms, paying special attention to ant colony, bee honey colony, bird flocks, bats, and fireflies.

We have also analyzed several works based on nonanimal behaviors such as GA, immune systems, bacteria, or artificial plant optimization algorithm. As we can see, these behaviors are less used than animal behaviors. In fact, almost 75% of bioinspired proposed systems are based on animals' behavior. Within nonanimals behavior, the most used is GA and the less used are the systems based on plants' behavior.

Regarding imitated animals, the most used mechanism is the ant colony technique. Although, behaviors based on swarms or flocks are often used.

Finally, we have seen that almost 50% of proposals are focused on improving routing tasks aspects. The issue of enhancing the energy efficiency in WSN is a hot topic within bioinspired systems. A striking fact is that very few of these proposals are focused on enhancing the network security.

As we have seen, there exists a huge variety of bioinspired mechanisms. Due to the apparition of new systems in the last years, such as monkey or bat behavior, in last two years, we expect that this knowledge area will continue growing. We think that these systems will offer new contributions to the different areas where they can be applied, not only in WSN, but also in MANET, Ad-hoc network, Mathematics, Robotics, and other application fields.

As future works, we would like to analyze in depth some of these mechanisms to improve several of our proposals within WSN such as indoor location [98], environmental monitoring [2, 4, 99, 100], tracking animals [101], disabled and elderly people monitoring [102], underwater communications [103, 104], and some other systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] R. Iram, M. I. Sheikh, S. Jabbar, and A. A. Minhas, "Computational intelligence based optimization in wireless sensor network," in *Proceedings of the 4th International Conference on Information and Communication Technologies (ICICT '11)*, pp. 1–6, Karachi, Pakistan, July 2011.

[2] J. Lloret, I. Bosch, S. Sendra, and A. Serrano, "A wireless sensor network for vineyard monitoring that uses image processing," *Sensors*, vol. 11, no. 6, pp. 6165–6196, 2011.

[3] P. A. C. S. Neves, J. F. P. Fonseca, and J. J. P. C. Rodrigues, "Simulation tools for wireless sensor networks in medicine: a comparative Study," in *Proceedings of the 1st International Conference on Biomedical Electronics and Devices (BIODEVICES '08)*, pp. 111–114, Funchal, Portugal, January 2008.

[4] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A wireless sensor network deployment for rural and forest fire detection and verification," *Sensors*, vol. 9, no. 11, pp. 8722–8747, 2009.

[5] F. Viani, G. Oliveri, M. Donelli, L. Lizzi, P. Rocca, and A. Massa, "WSN-based solutions for security and surveillance," in *Proceedings of the European Microwave Conference (EuMC '10)*, pp. 1762–1765, Paris, France, September 2010.

[6] P. Dasgupta, "A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 38, no. 3, pp. 549–563, 2008.

[7] M. Quwaider and S. Biswas, "Delay tolerant routing protocol modeling for low power wearable wireless sensor networks," *Network Protocols and Algorithms*, vol. 4, no. 3, pp. 15–34, 2012.

[8] S. Sendra, J. Lloret, M. García, and J. F. Toledo, "Power saving and energy optimization techniques for wireless sensor networks," *Journal of Communications*, vol. 6, no. 6, pp. 439–459, 2011.

[9] M. Liu and C. Song, "Ant-based transmission range assignment scheme for energy hole problem in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 290717, 12 pages, 2012.

[10] M. Garcia, D. Bri, S. Sendra, and J. Lloret, "Practical deployments of wireless sensor networks: a survey," *International Journal on Advances in Networks and Services*, vol. 3, no. 1, pp. 170–185, 2010.

[11] G. Riva and J. M. Finochietto, "Pheromone-based in-network processing for wireless sensor network monitoring systems," *Network Protocols and Algorithms*, vol. 4, no. 4, pp. 156–173, 2012.

[12] M. Garcia, S. Sendra, J. Lloret, and A. Canovas, "Saving energy and improving communications using cooperative group-based wireless sensor networks," *Telecommunication Systems*, vol. 52, no. 4, pp. 2489–2502, 2013.

[13] J.-Y. Kim, T. Sharma, B. Kumar, G. S. Tomar, K. Berry, and W.-H. Lee, "Intercluster ant colony optimization algorithm for wireless sensor network in dense environment," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 457402, 10 pages, 2014.

[14] Z. S. Bojkovic, B. M. Bakmaz abd, and M. R. Bakmaz, "Security issues in wireless sensor networks," *International Journal of Communications*, vol. 2, pp. 106–115, 2008.

[15] F. Dressler and O. B. Akan, "A survey on bio-inspired networking," *Computer Networks*, vol. 54, no. 6, pp. 881–900, 2010.

[16] B. Atakan and Ö. B. Akan, "Immune system based distributed node and rate selection in wireless sensor networks," in *Proceeding of the 1st Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS '06)*, pp. 1–8, Madonna di Campiglio, Italy, December 2006.

[17] R. Di Pietro and N. V. Verde, "Introducing epidemic models for data survivability in unattended wireless sensor networks," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM '11)*, pp. 1–6, Lucca, Italy, June 2011.

[18] F. Dressler and Ö. B. Akan, "A survey on bio-inspired networking," *Computer Networks*, vol. 54, no. 6, pp. 881–900, 2010.

[19] S. Marwaha, J. Indulska, and M. Portmann, "Biologically inspired ant-based routing in mobile ad hoc networks (MANET): a survey," in *Proceedings of the 6th Ubiquitous, Autonomic and Trusted Computing (UIC-ATC '09)*, Brisbane, Australia, July 2009.

[20] V. Jha, K. Khetarpal, and M. Sharma, "A survey of nature inspired routing algorithms for MANETs," in *Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT '11)*, vol. 6, pp. 16–24, Kanyakumari, India, April 2011.

[21] J. L. Fernandez-Marquez, G. Di Marzo Serugendo, S. Montagna, M. Viroli, and J. L. Arcos, "Description and composition of bio-inspired design patterns: a complete overview," *Natural Computing*, vol. 12, no. 1, pp. 43–67, 2013.

[22] T. Camilo, C. Carreto, J. S. Silva, and F. Boavida, "An energy-efficient ant-based routing algorithm for wireless sensor networks," in *Ant Colony Optimization and Swarm Intelligence*, pp. 49–59, Springer, Berlin, Germany, 2006.

[23] S. Selvakennedy, S. Sinnappan, and Y. Shang, "T-ANT: a nature-inspired data gathering protocol for wireless sensor networks," *Journal of Communications*, vol. 1, no. 2, pp. 22–29, 2006.

[24] R. Misra and C. Mandal, "Ant-aggregation: ant colony algorithm for optimal data aggregation in wireless sensor networks," in *Proceedings of the IFIP International Conference on Wireless and Optical Communications Networks*, pp. 1–5, Bangalore, India, April 2006.

[25] A. M. S. Almshreqi, B. M. Ali, M. F. A. Rasid, A. Ismail, and P. Varahram, "An improved routing mechanism using bio-inspired for energy balancing in wireless sensor networks," in *Proceeding of the 26th International Conference on Information Networking (ICOIN '12)*, pp. 150–153, Bali, India, February 2012.

[26] R. GhasemAghaei, A. M. Rahman, M. A. Rahman, and W. Gueaieb, "Ant colony-based many-to-one sensory data routing in wireless sensor networks," in *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications*, pp. 1005–1010, Doha, Qatar, April 2008.

[27] G. Chen, T. Guo, W. Yang, and T. Zhao, "An improved ant-based routing protocol in wireless sensor networks," in *Proceedings of the 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom '06)*, pp. 1–7, Atlanta, Ga, USA, November 2006.

[28] K. Saleem, N. Fisal, M. A. Baharudin, A. A. Ahmed, S. Hafizah, and S. Kamilah, "Ant colony inspired self-optimized routing protocol based on cross layer architecture for wireless sensor networks," *WSEAS Transactions on Communications*, vol. 9, no. 10, pp. 669–678, 2010.

[29] S. Okdem and D. Karaboga, "Routing in wireless sensor networks using ant colony optimization," in *Proceedings of the 1st NASA/ESA Conference on Adaptive Hardware and Systems (AHS '06)*, pp. 401–404, Istanbul, Turkey, June 2006.

[30] A. Salehpour, B. Mirmobin, A. Afzali-Kusha, and S. Mohammadi, "An energy efficient routing protocol for cluster-based wireless sensor networks using ant colony optimization," in *Proceeding of the International Conference on Innovations in Information Technology (IIT '08)*, pp. 455–459, Al Ain, United Arab Emirates, December 2008.

[31] Y. F. Wen, Y. Q. Chen, and M. Pan, "Adaptive ant-based routing in wireless sensor networks using energy delay metrics," *Journal of Zhejiang University: Science A*, vol. 9, no. 4, pp. 531–538, 2008.

[32] W. H. Liao, Y. Kao, and R. T. Wu, "Ant colony optimization based sensor deployment protocol for wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6599–6605, 2011.

[33] K. Pavai, A. Sivagami, and D. Sridharan, "Study of routing protocols in wireless sensor networks," in *Proceedings of the International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT '09)*, pp. 522–525, Trivandrum, India, December 2009.

[34] L. Juan, S. Chen, and Z. Chao, "Ant system based anycast routing in wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 2420–2423, Shanghai, China, September 2007.

[35] C. Wang and Q. Lin, "Swarm intelligence optimization based routing algorithm for wireless sensor networks," in *Proceedings of the IEEE International Conference Neural Networks and Signal Processing (ICNNSP '08)*, pp. 136–141, Zhenjiang, China, June 2008.

[36] H. Jiang, M. R. Wang, M. Liu, and J. W. Yan, "A quantum-inspired ant-based routing algorithm for WSNs," in *Proceeding of the 16th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD '12)*, pp. 609–615, Wuhan, China, May 2012.

[37] K. Saleem, N. Fisal, S. Hafizah, S. Kamilah, and R. A. Rashid, "Ant based self-organized routing protocol for wireless sensor networks," *International Journal of Communication Networks & Information Security*, vol. 1, no. 2, pp. 42–46, 2009.

[38] A. M. Okazaki and A. A. Frohlich, "Ant-based dynamic hop optimization protocol: a routing algorithm for Mobile Wireless Sensor Networks," in *Proceedings of the IEEE GLOBECOM Workshops (GC Wkshps '11)*, pp. 1139–1143, Houston, Tex, USA, December 2011.

[39] X. Hui, Z. Zhigang, and Z. Xueguang, "A novel routing protocol in wireless sensor networks based on ant colony optimization," in *proceedings of the International Conference on Environmental Science and Information Application Technology (ESIAT '09)*, pp. 646–649, Wuhan, China, July 2009.

[40] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[41] N. H. A. Ismail and R. Hassan, "6LoWPAN local repair using bio inspired artificial bee colony routing protocol," in *Proceedings of the 4th International Conference on Electrical Engineering and Informatics (ICEEI '13)*, vol. 11, pp. 281–287, 2013.

[42] H. S. Abdelsalam and S. Olariu, "BEES: bioinspired backbone selection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 44–51, 2012.

[43] A. Da Silva Rego, J. Celestino, A. Dos Santos, E. C. Cerqueira, A. Patel, and M. Taghavi, "BEE-C: a bio-inspired energy efficient cluster-based algorithm for data continuous dissemination in wireless sensor networks," in *Proceedings of the 18th IEEE International Conference on Networks (ICON '12)*, pp. 405–410, Singapore, December 2012.

[44] J. L. Fernandez-Marquez, G. di Marzo Serugendo, S. Montagna, M. Viroli, and J. L. Arcos, "Description and composition of bio-inspired design patterns: a complete overview," *Natural Computing*, vol. 12, no. 1, pp. 43–67, 2013.

[45] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artificial Intelligence Review*, pp. 1–33, 2012.

[46] P. Antoniou, A. Pitsillides, T. Blackwell, and A. Engelbrecht, "Employing the flocking behavior of birds for controlling congestion in autonomous decentralized networks," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 1753–1761, Trondheim, Norway, May 2009.

[47] P. Antoniou, A. Pitsillides, A. Engelbrecht, and T. Blackwell, "A swarm intelligence congestion control approach for autonomous decentralized communication networks," in *Applied Swarm Intelligence*, A. Engelbrecht and M. Middendorf, Eds., Springer Series in Studies in Computational Intelligence, 2008.

[48] P. Antoniou, A. Pitsillides, A. Engelbrecht, T. Blackwell, and L. Michael, "Congestion control in wireless sensor networks based on the bird flocking behavior," in *Proceedings of the 4th IFIP TC 6 International Workshop: Self-Organizing Systems (IWSOS '09)*, pp. 220–225, Zurich, Switzerland, December 2009.

[49] Z. Ruihua, J. Zhiping, L. Xin, and H. Dongxue, "Double cluster-heads clustering algorithm for wireless sensor networks using PSO," in *Proceeding of the 6th IEEE Conference on Industrial Electronics and Applications (ICIEA '11)*, pp. 763–766, Beijing, China, June 2011.

[50] D. Ma, J. Ma, and P. Xu, "An adaptive assistant-aided clustering protocol for WSNs using niching particle swarm optimization,"

in *Proceedings of the 4th IEEE International Conference on Software Engineering and Service Science (ICSESS '13)*, pp. 648–651, Beijing, China, May 2013.

[51] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 205–210, San Antonio, Tex, USA, October 2009.

[52] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 40, no. 6, pp. 663–675, 2010.

[53] X. Song, C. Wang, J. Wang, and B. Zhang, "A hierarchical routing protocol based on AFSO algorithm for WSN," in *Proceedings of the International Conference on Computer Design and Applications (ICCDA '10)*, pp. 635–639, Qinhuangdao, China, June 2010.

[54] X. Z. Gao, Y. Wu, K. Zenger, and X. Huang, "A knowledge-based artificial fish-swarm algorithm," in *Proceedings of the 13th IEEE International Conference on Computational Science and Engineering (CSE '10)*, pp. 327–332, Hong Kong, December 2010.

[55] M. Jiang, D. Yuan, and Y. Cheng, "Improved artificial fish swarm algorithm," in *Proceedings of the 5th International Conference on Natural Computation (ICNC '09)*, pp. 281–285, Tianjian, China, August 2009.

[56] L. Wang and L. Ma, "A hybrid artificial fish swarm algorithm for Bin-packing problem," in *Proceedings of the International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT '11)*, pp. 27–29, Heilongjiang, China, August 2011.

[57] E. M. D. G. Fernandes, T. F. Martins, and A. M. A. Rocha, "Fish swarm intelligent algorithm for bound constrained global optimization," in *Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE '09)*, Murcia, Spain, June-July 2009.

[58] W. Yiyue, L. Hongmei, and H. Hengyang, "Wireless sensor network deployment using an optimized artificial fish swarm algorithm," in *Proceedings of the International Conference on Computer Science and Electronics Engineering (ICCSEE '12)*, pp. 90–94, Hangzhou, China, March 2012.

[59] R. M. Brigham, E. K. V. Kalko, G. Jones, S. Parsons, and H. J. G. A. Limpens, *Bat Echolocation Research: Tools, Techniques and Analysis*, Bat Conservation International, Austin, Tex, USA, 2004.

[60] I. Fister Jr., D. Fister, and X.-S. Yang, "A hybrid bat algorithm," *Elektrotehniški Vestnik*, vol. 80, no. 1-2, pp. 1–7, 2013.

[61] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284 of *Studies in Computational Intelligence*, pp. 65–74, Springer, Berlin, Germany, 2010.

[62] S. Goyal and M. S. Patterh, "Wireless sensor network localization based on BAT algorithm," *International Journal of Emerging Technologies in Computational and Applied Sciences*, vol. 13, no. 192, pp. 507–512, 2013.

[63] S. Goyal and M. S. Patterh, "Performance of BAT algorithm on localization of wireless sensor network," *International Journal of Computers & Technology*, vol. 6, no. 3, pp. 351–358, 2013.

[64] S. Łukasik and S. Żak, "Firefly algorithm for continuous constrained optimization tasks," in *Proceedings of the 1st International Conference (ICCCI '09)*, Wrocław, Poland, October 2009.

[65] K. N. Krishnanand and D. Ghose, "Glowworms swarm based optimization algorithm for multimodal functions with collective robotics applications," *International Journal of Multiagent and Grid Systems*, vol. 2, no. 3, pp. 209–222, 2006.

[66] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Proceedings of the 5th International Symposium (SAGA '09)*, pp. 169–178, Sapporo, Japan, October 2009.

[67] A. Apostolopoulos and A. Vlachos, "Application of the firefly algorithm for solving the economic emissions load dispatch problem," *International Journal of Combinatorics*, vol. 2011, Article ID 523806, 23 pages, 2011.

[68] W. Liao, Y. Kao, and Y. Li, "A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12180–12188, 2011.

[69] Y. Sun, Q. Jiang, and K. Zhang, "A clustering scheme for Reachback firefly synchronicity in wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC '12)*, pp. 27–31, Beijing, China, September 2012.

[70] A. M. Zungeru, L.-M. Ang, and K. P. Seng, "Termite-hill: from natural to artificial termites in sensor networks," *International Journal of Swarm Intelligence Research*, vol. 3, no. 4, pp. 1–23, 2013.

[71] M. A. Bharathi, B. P. Vijayakumar, and D. H. Manjaiah, "Cluster based data aggregation in WSN using swarm optimization technique," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 12, 2013.

[72] S. C. Desai, K. S. Jagadeesh, and K. D. Dhruve, "Enhancing network lifetime in wireless sensor networks adopting elephant swarm optimization," *Global Journal of Computer Science and Technology: (E) Network, Web & Security*, vol. 13, no. 4, 2013.

[73] M. A. Bharathi, B. P. Vijayakumar, and M. Mallikarjuna, "A novel approach for energy effecient data aggregation using elephant behavior as a swarm intelligence," *Lecture Notes on Software Engineering*, vol. 1, no. 2, pp. 153–155, 2013.

[74] S. Kumar and S. M. Kusuma, "Clustering protocol for wireless sensor networks based on Rhesus Macaque (*Macaca mulatta*) animal's social behavior," *International Journal of Computer Applications*, vol. 87, no. 8, 2014.

[75] M. Breza and J. A. McCann, "Lessons in implementing bio-inspired algorithms on wireless sensor networks," in *Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems (AHS '08)*, pp. 271–276, Noordwijk , The Netherlands, June 2008.

[76] C. Huadong, W. Shuzong, L. Jingxi, and L. Yunfan, "A hybrid of artificial fish swarm algorithm and particle swarm optimization for feed forward neural network training," in *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering (ISKE '07)*, Chengdu, China, October 2007.

[77] N. A. B. A. Aziz, A. W. Mohemmed, and B. S. Daya Sagar, "Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization," in *Proceedings of the International Conference on Intelligent and Advanced Systems (ICIAS '07)*, pp. 961–965, Kuala Lumpur, Malaysia, November 2007.

[78] Y. Sun and J. Tian, "WSN path optimization based on fusion of improved ant colony algorithm and genetic algorithm," *Journal of Computational Information Systems*, vol. 6, no. 5, pp. 1591–1599, 2010.

[79] R. Falcon, X. Li, A. Nayak, and I. Stojmenovic, "A harmony-seeking firefly swarm to the periodic replacement of damaged

sensors by a team of mobile robots," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 4914–4918, Ottawa, Canada, June 2012.

[80] P. Antoniou, A. Pitsillides, and P. Koullourou, "Congestion control in wireless sensor networks based on the lotka volterra competition model," *Biologically Inspired Networking and Sensing: Algorithms and Architectures*, pp. 158–181, 2010.

[81] P. Antoniou and A. Pitsillides, "A bio-inspired approach for streaming applications in wireless sensor networks based on the Lotka-Volterra competition model," *Computer Communications*, vol. 33, no. 17, pp. 2039–2047, 2010.

[82] K. Benahmed, M. Merabti, and H. Haffaf, "Inspired social spider behavior for secure wireless sensor networks," *International Journal of Mobile Computing and Multimedia Communications*, vol. 4, no. 4, pp. 1–10, 2012.

[83] N. A. Alrajeh and J. Lloret, "Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 351047, 6 pages, 2013.

[84] A. P. Bhondekar, R. Vig, M. L. Singla, C. Ghanshyam, and P. Kapur, "Genetic algorithm based node placement methodology for wireless sensor networks," in *Proceedings of the International Multiconference of Engineers and Computer Scientists*, vol. 1, pp. 18–20, Hong Kong, China, March 2009.

[85] S. Hussain, A. W. Matin, and O. Islam, "Genetic algorithm for hierarchical wireless sensor networks," *Journal of Networks*, vol. 2, no. 5, pp. 87–97, 2007.

[86] S. Hussain, A. W. Matin, and O. Islam, "Genetic algorithm for energy efficient clusters in wireless sensor networks," in *Proceedings of the 4th International Conference on Information Technology: New Generations (ITNG '07)*, pp. 147–151, Las Vegas, Nev, USA, April 2007.

[87] K. P. Ferentinos and T. A. Tsiligiridis, "Adaptive design optimization of wireless sensor networks using genetic algorithms," *Computer Networks*, vol. 51, no. 4, pp. 1031–1051, 2007.

[88] J. Jia, J. Chen, G. Chang, and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm," *Computers and Mathematics with Applications*, vol. 57, no. 11-12, pp. 1756–1766, 2009.

[89] W. Y. Poe and J. B. Schmitt, "Placing multiple sinks in time-sensitive wireless sensor networks using a genetic algorithm," in *Proceedings of the 14th GI/ITG Conference Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB '08)*, pp. 1–15, Dortmund, Germany, March-April 2008.

[90] G. Nan, M. Li, and J. Li, "Estimation of node localization with a real-coded genetic algorithm in WSNS," in *Proceedings of the 6th International Conference on Machine Learning and Cybernetics (ICMLC '07)*, pp. 873–878, Hong Kong, China, August 2007.

[91] R. Teng, K. Leibnitz, and B. Zhang, "Immune system inspired reliable query dissemination in wireless sensor networks," in *Proceedings of the 10th International Conference on Artificial Immune Systems (ICARIS '11)*, pp. 282–293, Cambridge, UK, July 2011.

[92] K. Saleem, N. Fisal, M. S. Abdullah, A. B. Zulkarmwan, S. Hafizah, and S. Kamilah, "Proposed nature inspired self-organized secure autonomous mechanism for WSNs," in *proceedings of the 1st Asian Conference on Intelligent Information and Database Systems (ACIIDS '09)*, pp. 277–282, Dong Hoi, Vietnam, April 2009.

[93] A. Jabbari and W. Lang, "Advanced bio-inspired plausibility checking in a wireless sensor network using Neuro-immune systems: autonomous fault diagnosis in an intelligent transportation system," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM '10)*, pp. 108–114, Venice, Italy, July 2010.

[94] V. Dhiman, "BIO inspired hybrid routing protocol for wireless sensor networks," *International Journal for Advance Research in Engineering and Technology*, vol. 1, no. 4, pp. 33–36, 2013.

[95] S. Sribala and T. Virudhunagar, "Energy efficient routing in wireless sensor networks using modified bacterial foraging algorithm," *International Journal of Research in Engineering & Advanced Technology*, vol. 1, no. 1, pp. 1–5, 2013.

[96] V. Ponnusamy and A. Abdullah, "Biologically-inspired (botany) mobile agent based self-healing wireless sensor network," in *Proceeding of the 6th International Conference on Intelligent Environments (IE '10)*, pp. 215–219, Kuala Lumpur, Malaysia, July 2010.

[97] J. Li, Z. Cui, and Z. Shi, "An improved artificial plant optimization algorithm for coverage problem in WSN," *Sensor Letters*, vol. 10, no. 8, pp. 1874–1878, 2012.

[98] S. Sendra, J. Lloret, C. Turro, and J. Aguiar, "IEEE 802.11a/b/g/n short scale indoor wireless sensor placement," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1–3, 2014.

[99] S. Sendra, A. T. Lloret, J. Lloret, and J. J. P. C. Rodrigues, "A wireless sensor network deployment to detect the degeneration of cement used in construction," *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, vol. 15, no. 1/2/3, 2014.

[100] S. Sendra, F. Llario, L. Parra, and J. Lloret, "Smart wireless sensor network to detect and protect sheep and goats to wolf attacks," *Recent Advances in Communications and Networking Technology*, vol. 2, no. 2, pp. 91–101, 2013.

[101] S. Sendra, E. Granell, J. Lloret, and J. J. P. C. Rodrigues, "Smart collaborative mobile system for taking care of disabled and elderly people," *Mobile Networks and Applications*, vol. 19, no. 3, pp. 287–302, 2014.

[102] M. Garcia, S. Sendra, G. Lloret, and J. Lloret, "Monitoring and control sensor system for fish feeding in marine fish farms," *IET Communications*, vol. 5, no. 12, pp. 1682–1690, 2011.

[103] S. Sendra, J. Lloret, J. J. P. C. Rodrigues, and J. M. Aguiar, "Underwater Wireless Communications in Freshwater at 2.4 GHz," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1794–1797, 2013.

[104] J. Lloret, S. Sendra, M. Ardid, and J. J. P. C. Rodrigues, "Underwater wireless sensor communications in the 2.4 GHz ISM frequency band," *Sensors*, vol. 12, no. 4, pp. 4237–4264, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 245924, 20 pages
http://dx.doi.org/10.1155/2014/245924

Research Article

In-Network Filtering Schemes for Type-Threshold Function Computation in Wireless Sensor Networks

Guillermo G. Riva[1,2] and Jorge M. Finochietto[2,3]

[1] *Universidad Tecnológica Nacional, Facultad Regional Córdoba, Maestro Lopez y Cruz Roja Argentina, X5016ZAA Córdoba, Argentina*
[2] *CONICET, Haya de la Torre S/N, Ciudad Universitaria, 5016 Córdoba, Argentina*
[3] *Universidad Nacional de Córdoba, Velez Sarsfield 1611, Ciudad Universitaria, X5016GCA Córdoba, Argentina*

Correspondence should be addressed to Guillermo G. Riva; griva@scdt.frc.utn.edu.ar

Received 2 February 2014; Revised 2 July 2014; Accepted 2 July 2014; Published 14 August 2014

Academic Editor: Jaime Lloret

Data collection in wireless sensor networks (WSNs) can become extremely expensive in terms of power consumption if all measurements have to be fetched. However, since multiple applications do not require data from all nodes but to compute a function over a smaller data set, much of the available data on the network can be considered irrelevant and not worthy of spending energy. In this context, in-network filtering schemes can be used to forward only relevant data towards a sink node for processing purposes. In this work, we propose and evaluate two schemes that can drive this filtering process. Both of them are based on the integration of metaheuristics and learning algorithms inspired by nature. In particular, we consider the computation of the maximum function as case study for these schemes. We investigate the trade-off between communications costs, which are directly associated with power consumption, and error costs due to fetching not all relevant data. We show by simulation that communication costs can be significantly reduced with respect to traditional schemes while keeping the computation error bounded.

1. Introduction

A wireless sensor network (WSN) consists of a set of small and low cost sensor nodes connected by wireless links, which can be deployed to obtain information about physical phenomena such as heat, light, noise, radiation, and chemical emissions. Typically, sensor readings are sent to a collector node (i.e., sink) using multihop forwarding, according to a data delivery model [1–3]. However, sensor nodes can have limited sensing, processing, and communication capabilities due to energy constraints resulting from battery operation. Hence, energy efficiency is a key issue in the design of systems based on this technology, where data communication can typically be considered the most energy demanding task.

Besides forwarding packets, WSN nodes are typically allowed to process in transit data available on the payload of these packets. Data can be aggregated or filtered (i.e., discarded) in order to decrease the amount of packets traveling through the network; thus, saving energy by reducing communication costs [4]. While data aggregation, which aims at fusing redundant data, has been thoroughly investigated [5], data filtering has received much less attention. In particular, the computation of *type-sensitive* and *type-threshold* functions [6, 7] can significantly benefit from in-network filtering schemes. The former requires to know a minimum fraction of arguments for the function value to be determined. Examples include average, median, histogram, and majority. The latter depends only on element-wise maximum (minimum) of the histogram and a threshold vector. Instances of these functions are maximum, minimum, range, bottom-n, and top-n. Intuitively, the value of a type-sensitive function can be accurately determined if a large fraction of the arguments are known, whereas the value of a type-threshold function can be determined by a smaller amount of arguments. Consequently, type-threshold function can potentially be computed with low communication costs.

A major issue in type-threshold function computation is *how to select a set of arguments* (i.e., *node readings) for the function value to be accurately determined*. The most naive solution is to consider all arguments, thus all network data without any kind of filtering. This guarantees the accuracy of the computed function value but involves querying all nodes and forwarding all readings to a sink for computation, which can demand a huge communication effort as the network size increases. Ideally, there exist a minimal set of nodes which can return the actual function value, but this set is unknown a priori. Since functions are typically computed over time, a learning scheme could be used to select a set of arguments that can provide with high probability the function value. If so, only a limited number of nodes can actually be queried for data, which can reduce the message count required to compute the given function. In this way, it is possible to implement low-latency *one-shot* computation schemes by issuing a single query at regular intervals that fetches relevant data only.

In this work, we consider the problem of filtering inside the network, those arguments not actually required to compute a type-threshold function. To this end, only the set of arguments which can be used to determine the function value at the sink node are selected. Indeed, this selection constitutes a precomputation of the function value for type-threshold functions, which can be performed inside the network in order to save energy. The network can provide the best argument candidates for a given function to a sink, which can then process these data to obtain the function value. For the maximum (minimum) function computation, just the argument with maximum (minimum) value could be provided, while, for the range case, all arguments belonging to the range. However, in general, we may provide (i) a larger set, which can result in a high communication cost but no computation error, or (ii) a smaller set, which results in a computation error. A trade-off between cost and error is thus present in solutions addressing this problem.

Type-threshold functions can be divided into two categories based on how the threshold level is defined. *Fixed-threshold functions* define a threshold which is known a priori and can be embedded on the query as a constant value. Nodes only read this threshold to determine whether their readings are relevant or not. The range and isocontour functions are good example of this category. On the other hand, *dynamic-threshold* defines a threshold which is embedded on the query as a variable value. Nodes can not only read but also update this threshold. Among the functions belonging to this category, the maximum function is the most interesting example as it can be used to implement other functions such as minimum, top-n, and bottom-n. Besides, the maximum function is typically required in several WSN applications. In this work, we focus on filtering schemes which can be used for computing the maximum function.

Dynamic-threshold functions require filtering schemes that can adapt and learn from the network. To this end, we propose to integrate computational intelligence techniques (CI) on nodes. CI is a set of nature-inspired computational methodologies to address complex problems of the real world to which traditional methodologies are ineffective or infeasible. In this work, these techniques are used to provide nodes with *two filtering rules*, as shown in Figure 1. The first one, namely, *self-filtering rule*, determines whether a reading constitutes a relevant argument to the function. The second rule, namely, *neighbor-filtering rule*, determines if neighbor nodes can have relevant arguments or not. In other words, if neighbor data are assumed irrelevant, the query message (containing the function) is only forwarded with some (low) probability. These rules are not static (i.e., preconfigured) but *built on the fly* by means of learning mechanisms.

Nodes implement these simple rules iteratively resulting in a complex network behavior, where the network evolves from a nonfiltering state to a filtering one by means of independent decisions made by its nodes. In particular, our work considers two different approaches to implement the neighbor-filtering rule. The first scheme is based on the simulated annealing (SA) concept and centralizes network learning at the sink node to update a global query forwarding policy which fetches data from relevant areas. Instead, the second strategy distributes learning on network nodes in order to locally update the query dissemination policy at each node. This is achieved by means of the ant colony optimization (ACO) algorithm which is a bioinspired scheme based on the behaviour of ants seeking a path between their colony and a source of food. In our context, relevant data seeks a path to the sink node. When implemented over time, these schemes can learn from the network and dynamically provide a set of arguments that offers good performance in terms of communication cost and computation error.

This paper significantly extends our previous analysis reported in [8, 9], providing a much deeper discussion of the proposed schemes as well as new results. Besides, both schemes are compared to identify benefits and limitations, which can determine the best application scenario for each proposal. The main contributions of this work are as follows.

(i) A detailed description of how filtering schemes can be used in the context of in-network computation is presented. General approaches as well as particular solutions are discussed.

(ii) Proposed schemes are described in terms of forwarding and learning processes, which helps to analyze its behavior and compare their performance.

(iii) Different simulation scenarios are evaluated, thus enhancing result analysis and strengthening conclusions.

The rest of the paper is organized as follows. Section 2 briefly discusses related work on in-network computation with special focus on query mechanisms. Section 3 formalizes the problem we consider in this work and introduces the concept of filtering as an efficient solution. Section 4 describes the general network model and the behavior of nodes. Forwarding and learning algorithms used to implement neighbor-filtering rules are explained in Section 5. Section 6 discusses main results obtained by simulation. Finally, Section 7 concludes the work.

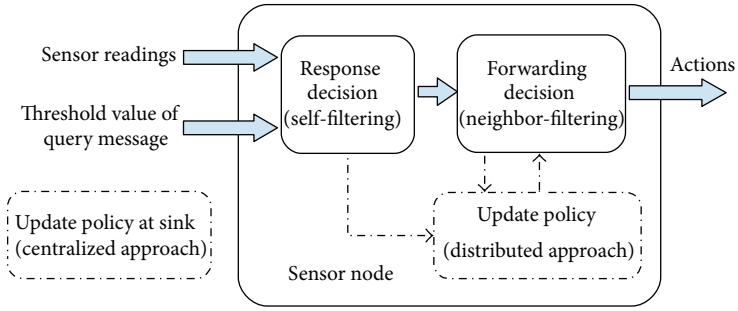


Figure 1: Decision rules at each sensor node.

2. Related Work

In-network computation of symmetric functions in multihop random WSNs has been discussed broadly in [6, 7]. In particular, the use of gossip-based algorithms has been identified as suitable solutions when both robustness and fault tolerance are required. These schemes have mainly considered type-sensitive functions (i.e., average) [10, 11] which are computed through data aggregation. However, a different approach can be considered for type-threshold functions, which actually require data searching rather than aggregation. In this sense, data aggregation techniques can be used to reduce the communication costs in WSN applications where a large amount of data needs to be sent from source nodes to a sink one (i.e, in many-to-one data flows) [2]. However, in case of type-threshold functions, where often only few nodes can possess arguments for computing a given function, these techniques can be inefficient. For this reason and because nodes with relevant readings are unknown a priori, our problem is more related to *data search/discovery problems* than to *data collection* ones.

Several query processing schemes for WSNs have been proposed in the literature, which can be classified from the point of view of both the required *infrastructure* and the *query dissemination* mechanism. The former can be infrastructure-based (such as tree, clustering, etc.) [12, 13] or infrastructure-free (i.e., unstructured) [13–15]. The latter can be walk-based [13–15], random walk-based [16], flooding-based [17, 18], and gradient-based [17]. In this sense, unstructured- and gradient-based schemes are the best option in case of type-threshold function computation due to the low overhead of unstructured mechanisms and the information gain concept of gradient-based strategies for query dissemination. The concept of using fingerprint gradients to direct query dissemination to nodes detecting events was proposed in [17]. This strategy is based on the fact that every physical event produces a fingerprint in the environment which results in a natural information gradient in the proximity of the phenomenon.

Most query processing schemes used in WSNs are based on *one-phase pull diffusion* [19] and rely on the network infrastructure for query propagation and data collection, which enables in-network processing for data reduction [14] (i.e., data aggregation). First, the sink node disseminates once an interest described as a function computation task. Besides, this query sets up backward paths from each sensor node to the sink. Afterwards, sensor nodes with relevant data for this function can either send data to the sink for computing the function value in a centralized way or distributively compute the function inside the network to deliver to the sink the final result. This last process can be repeated after a period of time to update the function value at the sink. However, since WSNs can be considered unreliable networks where nodes are prone to failure, these schemes can suffer from topology changes which can break up backward paths.

In this work, we focus on type-threshold function computation problems using *one-shot on-demand query-based schemes*. These types of schemes are much more robust to network changes since queries are disseminated each time the function is to be computed; thus backward paths are set up on-demand after each query. Note that this scheme, at first glance, might seem to demand high communication costs; however, if query dissemination can be combined with both *gradient-based routing strategies* for data searching and *in-network filtering schemes* query communication costs can be reduced. In this sense, each node, based on simple rules, participates in this process by filtering irrelevant readings as well as neighbor nodes not necessary for the function computation. As a consequence, the query area can be reduced iteratively, and the query dissemination can be directed only to those nodes which can give the best arguments for the function computation.

Our first proposed scheme drives the query dissemination process using a simulated annealing (SA) metaheuristic. Kirkpatrick et al. [20] proposed the SA algorithm to deal with traveling salesman and component placement problems. SA is a nature-based probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. Due to the distributed nature of WSNs, algorithms such as SA can be implemented in parallel fashion [17, 21], in order to compute type-threshold functions in the network. Our second proposed scheme is based on computational intelligence, in particular, bioinspired mechanisms. Kulkarni et al. [22] proposed reinforcement learning (RL) and swarm intelligence (SI) as the best options in WSNs from the point of view of computational and memory requirements, flexibility, and optimality. RL is biologically inspired and acquires its

knowledge by actively exploring its environment [23]. In the last years, communication and networking technologies are increasingly considered to integrate bioinspired strategies as robust and efficient solutions [24].

A current branch of swarm intelligence focuses on *ant colony optimization (ACO)* and pheromone-based mechanisms. In this sense, pheromone-based routing strategies mimic the behavior of ant colonies when searching for food. Upon finding food sources, ants return to their colony laying down pheromone trails. Other ants tend to follow these paths and to reinforce them releasing more pheromone. Over time, pheromone tends to evaporate to erase unused paths. Most pheromone-based strategies for WSNs address routing and aggregation problems [25, 26]. To the best of our knowledge, the problem of in-network filtering for computing type-threshold functions using bioinspired schemes in WSNs has not yet been treated.

3. Problem Formulation

Most WSN applications are required to compute a function over sensed data (i.e., measurements). We can formalize this as computing a function $f(X)$ where X is the set of readings from x_1 to x_n, with n being the number of sensor nodes in the network. In general, all x readings could be made available to the sink node in order to compute the function value in a centralized manner. In this case, sink incurs no computation error at the expense of a high communication cost (i.e., energy consumption), especially in large scale WSNs. However, type-threshold functions can typically be computed over a subset $X' \subseteq X$ of readings such that $f(X') = f(X)$. Thus, the problem of finding the set X' becomes relevant as it has the potential to compute the function value at lower communication costs.

Let $\widehat{X}' = g(X)$ be the set of arguments provided by a given $g(\cdot)$ in-network filtering function. If $\widehat{X}' \supseteq X'$, $g(\cdot)$ does not introduce any computation error. However, if $\widehat{X}' \not\supset X'$, the filtering adds some computation error $e = |f(X') - f(\widehat{X}')|/f(X')$. The communication cost c is harder to estimate but it is expected to be inversely proportional to $|\widehat{X}'|$ (i.e., the size of the reported set). This is due to the fact that as $\widehat{X}' \to X'$, the more message exchange is required by the $g(\cdot)$ filtering scheme. Consider, for example, flooding the network with a query related to computing a given range of values. Each node whose reading belongs to the range can report its value to the sink node. Since all nodes receive the query, then $\widehat{X}' = X'$ and $|\widehat{X}'| = |X'|$. However, if the query scheme is based on random walk, not all relevant nodes may receive the query message. Thus, we expect $\widehat{X}' \not\supset X'$ with $|\widehat{X}'| < |X'|$, which has some computation error but a lower communication cost since not all the network was explored.

The computation of a function $f(\cdot)$ can then be performed in two parts: a first one, given by $\widehat{X}' = g(X)$, which is computed inside the network, and a second one, given by $f(\widehat{X}')$, which is done at the sink node. Both processes are described in Figure 2. Since $g(\cdot)$ is to be implemented in a distributed fashion, it can be governed by two filtering rules: a *self-filtering* one, which decides whether the node's reading is relevant or not for the computation of the function $f(\cdot)$, and a *neighbor-filtering rule*, which determines if it is worth forwarding the query to neighbor nodes. Note that if the query is not forwarded, these nodes can be potentially excluded from the in-network computation process as their data are actually not even considered for reporting to the sink node.

The self-filtering rule is straightforward as it only requires evaluating the local data against a threshold available on the query message. However, as discussed on Section 1, fixed and dynamic thresholds can be considered. Fixed thresholds [27, 28] are used for searching readings within a specific range of values, while dynamic ones, for readings without a predefined range. Fixed thresholds can be implemented locally by each node despite the actual readings available at other nodes, while dynamic ones depend on these readings to update the threshold values [29]. The canonical example for a dynamic threshold is the maximum function where, given an initial threshold, all nodes whose readings are above the threshold update it on the query message. In this way, the threshold tends to increase as the query message is disseminated through the network. In general, we will consider for our analysis the case of the maximum (minimum) function since it makes use of dynamic thresholds; however, the analysis can also be extended to fixed ones.

Among both rules, the second one (neighbor-filtering) is the most challenging as it requires learning towards which directions relevant data are present. Since data can change over time, this process needs to be robust enough to track changes that can lead to considering new nodes and/or discarding existing ones. Thus, solutions to the problem of computing $\widehat{X}'$ need to tackle these challenges.

4. Network and Data Models

We consider a WSN composed of sensor nodes uniformly distributed over a square area, where communication range r and node density ρ are constant. In this sense, the more the sensor nodes in the network, the greater the area covered by the network. We assume the broadcast protocol model for wireless communication, where a node transmits information to all nodes in its communication range. In this sense, a node i can successfully transmit a packet to another node j if $d(i, j) < r$, where $d(\cdot)$ is the distance between these nodes. The communication range r is defined in order to assure a network connectivity near 99%. In this sense, a network with n nodes uniformly distributed (random network) can be considered as asymptotically connected with probability approaching one if each node is connected to more than $5.1774 \log(n)$ nearest neighbors [30].

Information sources (i.e., events) are modeled as uniformly distributed functions following a diffusion law with the distance; that is, $f(d) \propto 1/d^{\alpha}$, where α is the diffusion parameter (e.g., for light $\alpha \approx 2$, and for heat $\alpha \approx 1$) [17]. In this sense, most of the physical phenomena (e.g., light, heat, noise, radiation, etc.) follow this law. As a consequence, the reading

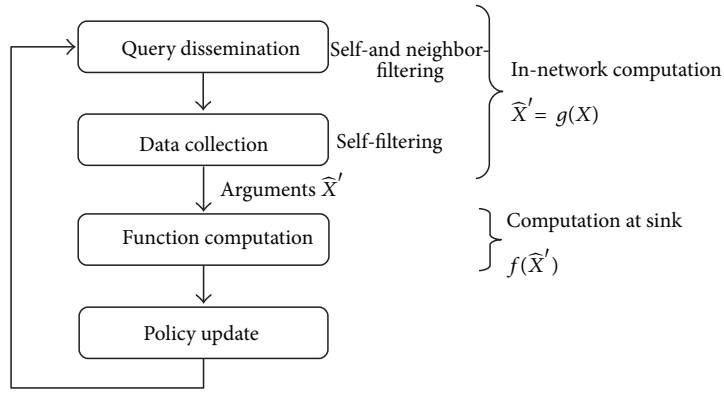


Figure 2: Function computation processes.

x_i obtained by the sensor node i is defined as the contribution of all events in the network, such as

$$x_i \propto \sum_{k=1}^{m} \frac{1}{d(i,k)^{\alpha}}, \tag{1}$$

where m is the number of events in the network, $d(\cdot)$ is the distance between sensor node i and event k, and α is the diffusion coefficient. In this work, we consider that all information sources emit the same amount of their physical magnitude. These sources could be objects such as light bulbs, air conditioners, heaters, and noise sources. For simplicity, we suppose that nodes are synchronized and sensors have infinite resolution.

In this context, a sink node, located at the center of the network, requires computing a specific function over time by injecting queries to the network. These queries are routed through the network to fetch relevant data $\widehat{X}'$. Thus, data flow in both directions: *downstream* (from sink to sensor nodes) and *upstream* (from sensor nodes to the sink). In the downstream direction, nodes can receive query messages, while, in the upstream direction, reply ones. In both directions, upon receiving any (query or reply) message, nodes decide both (i) if their readings are still worth to be reported to the sink and (ii) whether or not to broadcast the message to its neighbors.

In the *downstream direction*, the query message is first broadcasted by the sink node to all its neighbors who, after processing the query, can decide to forward it to their neighbors, and so on. Even if nodes can receive the same query multiple times, they can at most forward it once. The query message contains an identification (*msg-id*), a querier node identification (*querier-node-id*), a threshold value (*th*), and a hop counter (*hc*). Nodes keep track of the *ids* of already forwarded queries using local tables to avoid retransmissions. Initially, each node has no information of its distance in hops to the sink. This information is learned by each sensor node on each query iteration. In this sense, the distance to the sink (i.e., hop level) is calculated as the minimal hop count inside of received messages. Query messages are always processed even if received several times. Each time a node receives a query, the threshold value inside the message is read to determine if the node has relevant data (i.e., self-filtering). In *static filtering*, in which the query is routed through the whole network to select those nodes with readings above (below) a given (fixed) threshold value, each node can determine the relevance of its reading based on the first query message. Instead, in *dynamic filtering*, the threshold can be updated by nodes to implement maximum (minimum) function computation. Thus, even if a node may consider its data relevant on a first query arrival, it may learn on a successive query arrival that its data are not relevant at all as the threshold value has been updated. After processing the query, nodes must decide whether to forward or not the message to their neighbors (i.e., downstream neighbor-filtering). Even if a detailed discussion of the proposed schemes is presented in the next sections, we introduce its common principles: if a node believes that relevant data are present nearby, it will always relay the query message; otherwise, it will randomly decide whether to broadcast or not the query message. Since the function is computed over time, queries are periodically injected in the network by the sink node to calculate its value. Once the query process ends, only those nodes with relevant data (selected in downstream direction) report their readings ($\widehat{X}'$) to the sink after a short time interval Δt which is inversely proportional to the hop level of the selected node plus a small random time. In this sense, far selected nodes begin the response process first, filtering intermediate preselected nodes. As we will discuss later, each iteration feeds a learning process which helps to route next queries in a more efficient way. The query message conveys also a hop counter *hc*. This field is initialized to zero by the sink node and incremented by each receiving node to enable a *downstream learning* process. Since multiple messages from the sink may be received through different paths, nodes can discover their minimum distance (in hops) to the sink by this learning process.

In the *upstream direction*, selected nodes (i.e., not filtered) sent a reply message back to the sink containing their readings. This enhances the self-filtering process when

considering dynamic thresholds as preselected nodes in the downstream direction could be filtered in the upstream one. Even if each reply message is broadcasted, it is delivered to only one upstream node; thus, replies are routed back to the sink node through a single path. The broadcast mechanism helps to disable preselected nodes near the response path.

5. In-Network Filtering Schemes

Upon receiving either a query message or a reply one, nodes implement self-filtering in both directions. This is done comparing their readings with either thresholds in (downstream) query messages or arguments in (upstream) reply ones. Note that self-filtering does not introduce computation errors but can increase the communication cost due to the required message exchange to filter nodes with actually no relevant data. On the contrary, neighbor-filtering, which decides on whether it is worth broadcasting a query to neighbors, can introduce computation errors but tends to decrease the communication cost by avoiding querying nodes which may have useless data.

In this section, we describe two schemes for implementing the neighbor-filtering process. The first one considers that sensors nodes do not keep any state of previous filtering decisions; thus, it is referred to as *stateless filtering*. However, the sink node can learn from the information obtained from the network and modify the filtering rule over time by embedding some state information in the query messages that can be used to fetch only relevant data at a lower communication cost. Instead, in the second scheme, named *stateful filtering*, nodes maintain information about previous decisions while the sink node always keeps the same filtering rule. Both schemes can be described by (i) a downstream forwarding algorithm which implements the neighbor-filtering rule to route the query message through the network and (ii) an upstream learning process which actually adapts the dynamics of the neighbor-filtering rule in time.

5.1. Stateless Filtering. This scheme implements a version of the simulated annealing (SA) metaheuristic [20]. SA has the ability to explore beyond those areas where no relevant data are available due to its stochastic behavior. Since each query can be broadcasted to more than one node, this results in a natural parallelization of the algorithm as it can generate multiple forks of the same algorithm. The resulting algorithm is also known as *Parallel Adaptive Simulated Annealing (PASA)* [8]. Because PASA scheme is query-driven, it is conformed by two sequential phases, downstream forwarding and upstream learning. Both phases conform a computation iteration. The first one is used to forward the query to the sensor nodes following a one-to-many data flow model. The second one, instead, is implemented to send the arguments from selected nodes with relevant readings to the sink, following a one- or many-to-one data flow model, depending on the number of events in the network. These phases are detailed in the following subsections.

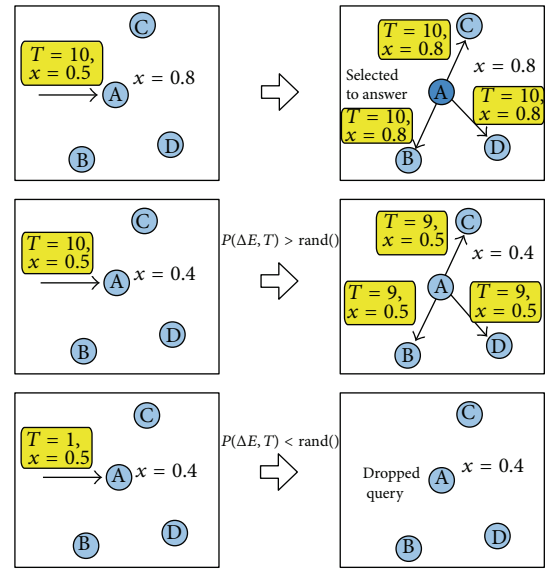


Figure 3: Downstream forwarding of the stateless scheme.

Downstream Forwarding. Each node can forward each query at most once in each computation iteration, in order to save as much energy as possible. The query message includes a T value, which represents the temperature of the algorithm, and the best found value (threshold) in the path followed by the query. Both parameters can be modified by each forwarding node. As a consequence, queries in the same iteration can have different T and threshold values depending on the sequence of broadcasts. A node having relevant data is configured to answer the query after a period of time and always broadcasts the query with the updated threshold to all its neighbors (Figure 3 top). Given that ΔE is the difference between the threshold value carried in the query message and the sensor reading, the condition $\Delta E \leq 0$ determines that relevant data are present, in case of maximum function computation. A node with no useful data forwards the query message to its neighbors with probability P computed as

$$P(\Delta E, T) = e^{-\Delta E/T}, \tag{2}$$

where T is the temperature parameter of the SA algorithm which is carried inside the query (Figure 3 center). Either the lower T value or the larger ΔE value, the lower probability P of forwarding the query to neighbors. In other words, the query is probably discarded by the forwarding node if the temperature of SA inside the query is low or nearby data are assumed irrelevant (Figure 3 bottom). A large T value encourages data exploration despite data relevance, thus increasing the probability P of forwarding queries. In our proposal, a large value of $T = T_0$ is initially used and then, in the query dissemination, linearly decreased by a D factor by intermediate nodes assuming irrelevant data, which is known as adaptive cooling. In the following iterations, a logarithmic decrement of T_0 is applied to accelerate the convergence of the algorithm. Instead, in the original SA algorithm, the cooling process is executed only at the beginning of each iteration. The value of D is computed by the sink node as $T_0/maxDist$,

```
(1)  Input Received query message
(2)  Output Response and forwarding decisions
(3)  id ← Msg.getId()
(4)  D ← Msg.getDec()
(5)  T ← Msg.getTemperature()
(6)  thresholdValue ← Msg.getThreshold()
(7)  ΔE ← thresholdValue − sensedValue
(8)  // Discard already forwarded queries
(9)  if idTable[id] == true then
(10)    delete(Msg)
(11) else
(12)    updateIdTable(id)
(13) end if
(14) // Query forwarding decision
(15) if ΔE ≤ 0 then
(16)    Msg.setValue(sensedValue)
(17)    Msg.setTemperature(T)
(18)    send(Msg)
(19) else
(20)    if P = e^{−ΔE/T} > rand() then
(21)       Msg.setTemperature(T − D)
(22)       send(Msg)
(23)    else
(24)       delete(Msg)
(25)    end if
(26) end if
```

Algorithm 1: Stateless Forwarding Scheme (PASA).

where *maxDist* is the maximum estimated distance from the sink in hops. This is due to the fact that there should exist a nonnull probability of exploring network borders even if irrelevant data are assumed; thus, it is required that $T > 0$ after traversing *maxDist* hops. Note that typically D will decrease with the network size. However, other cooling strategies could be considered. A detailed description of the forwarding algorithm of PASA is shown in Algorithm 1.

In the response process, single-path routing is implemented in order to reduce the communication cost, such as in [31]. Each selected node in PASA scheme uses the path of the first query arrival (lower latency path) to report arguments to the sink. Since PASA iteratively implements query and response phases, it can maintain an updated view of the network state. This allows dealing with nodes that can fail previous to an iteration. In case of a failure within an iteration, a simple *self-healing strategy* has been included to deal with this problem, which is discussed in detail in Section 6.4.

Upstream Learning. For a large value of T_0, downstream forwarding can behave as flooding as every node would always forward the query despite data relevance; thus, there is certainty in always finding most relevant data but at highest communication cost. At low T_0 values, it is equivalent to the gradient descent algorithm, where the query message is forwarded only through nodes with some relevant data. As a result, the probability of finding significant data is low, which can introduce computation errors. This feature is depicted in Figure 4 for the case of 10 events in the network. Different values for T_0 are considered and their impact is illustrated in terms of the computation error, the probability of success in computing the actual function value, and the query cost, which is defined by the average number of forwarded packets by each node in the network. From this analysis, we can conclude that there exists some T_0 value which can provide low or no computation error as well as low communication costs.

Therefore, it is necessary to find an optimal T_0 value to initialize T so that when linearly decreased it offers a good trade-off between communication costs and computation errors. For this purpose, a reinforcement learning (RL) algorithm is implemented in the sink node. The main idea behind this algorithm is that the sink can learn about the data distribution in the network based on its experience when receiving reply messages. It is implemented based on the *id* of response nodes. The sink node can apply a temperature update policy to adapt $T_0(i)$ value at each query iteration i to improve the search process, hence reducing energy consumption. The objective of the sink is to reduce the search depth iteratively by updating the $T_0(i)$ in order to reach only sensor nodes with relevant arguments.

This learning process is sketched in Figure 5. Even if nodes are scattered over an area A as illustrated, a first query iteration reaches nodes on area B. Nodes on (A-B) area were not queried since relevant data was not found nearby and the value of P became too small. On the second iteration, the initial $T_0(i)$ value is decreased to a $T_0(i+1)$ value, which reduces the search space to area C. This process is repeated till the algorithm finds out the value of $T_0(i)$, at some iteration i, that enables the query of all nodes with relevant data at the lowest cost, which in Figure 5 is represented by area E. At each query iteration, a decision on whether to decrease the previous $T_0(i-1)$ value or not is made by the sink. After sending the query for the first time with a high value of T_0, the sink records the number of node responses (i.e., the number of nodes which reported relevant data). As long as the sink gets data from the same number of responses, it is assumed that the previous $T_0(i-1)$ value can be reduced, so the sink decreases the injected $T_0(i)$ value in the next iteration i. A lower T_0 tends to narrow the search space, thus saving energy on nodes. If this number of responses decreases, the $T_0(i)$ value in iteration i is increased to an intermediate value between the $T_0(i-2)$, the last time without loss of information, and $T_0(i-1)$, the last iteration with loss of information. For example, $T_0(i-2) = 100$, $T_0(i-1) = 10$, and $T_0(i) = 45$ show this concept.

A description of logarithmic and linear cooling processes is sketched in Figure 6 in case of *max* function computation. The sink node is indicated as s node and a node selected to report its argument to the sink as n node. Query messages embed both the current T and threshold values. After i iterations, T_0 converges to a fixed $T_0(i)$ value which is used in successive queries, as shown in Figure 7. Note that if T_0 is decreased beyond a given value, the computation error increases as search space is significantly reduced and query forwarding becomes limited to areas with data gradients (i.e., $\Delta E \leq 0$). This last effect can be noticed by the number of wrong hops, which is defined as the average number of

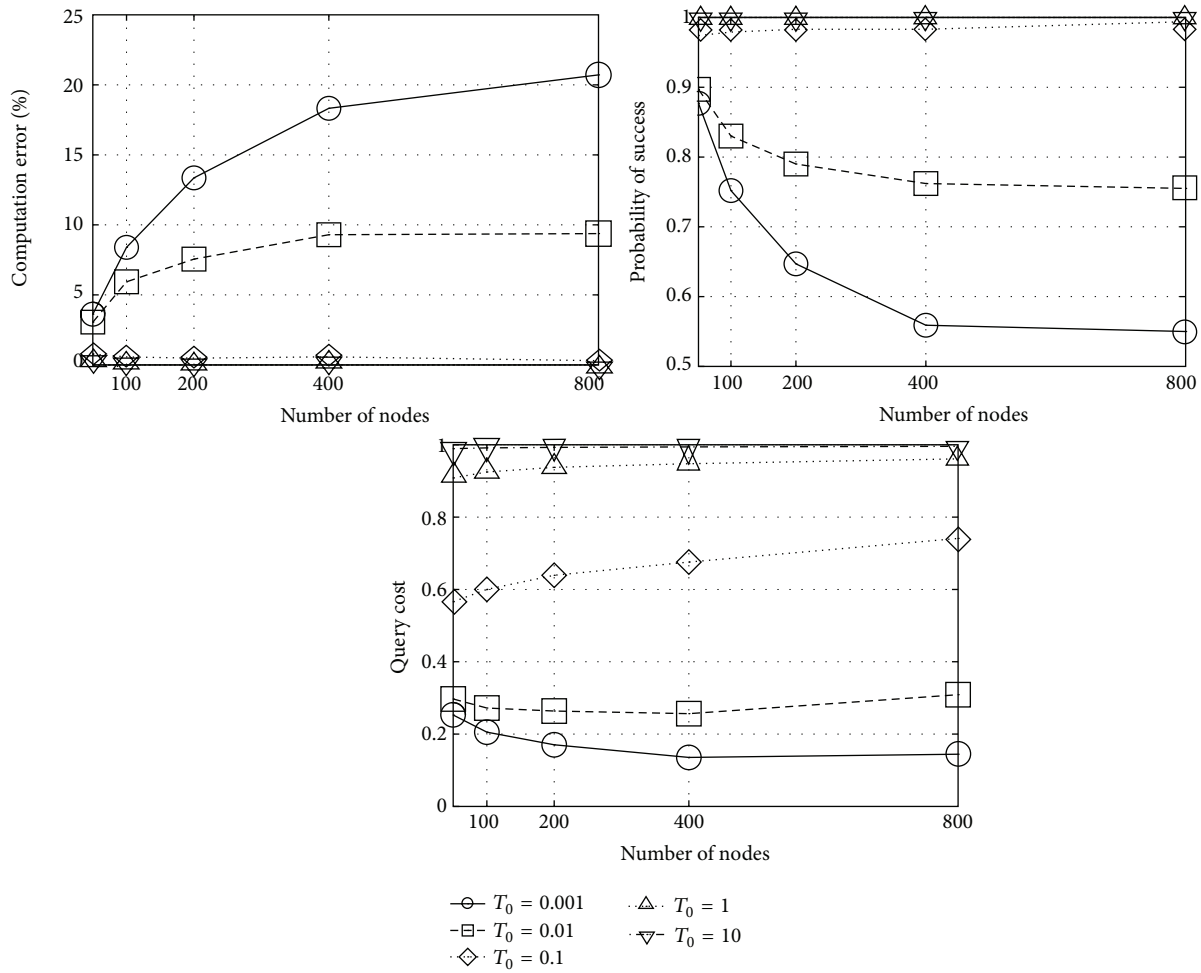


Figure 4: Performance of stateless scheme for different T_0 values.

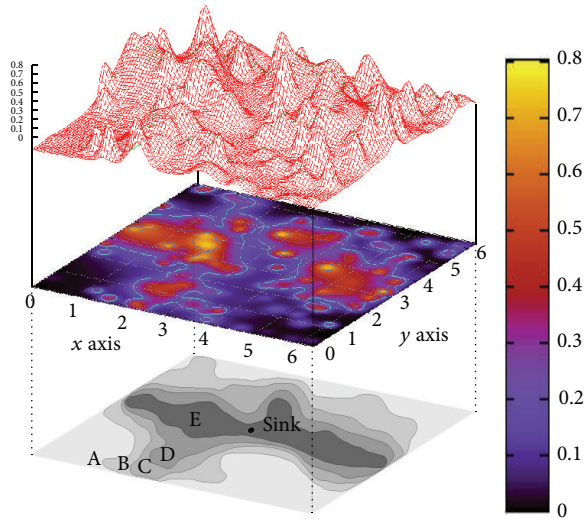


Figure 5: Iterative filtering with space reduction from A to E.

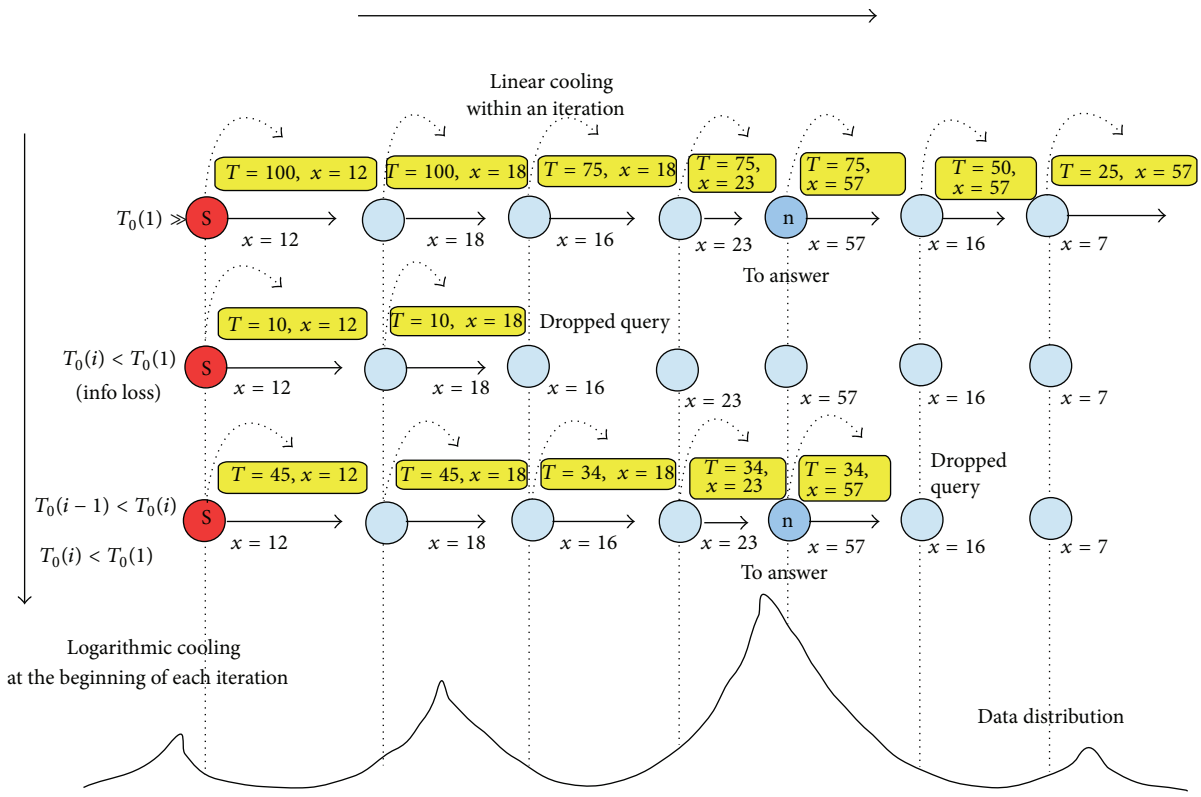


FIGURE 6: Cooling strategies used in stateless scheme.

queries forwarded by each node when data are considered not relevant ($\Delta E > 0$). Unlike traditional implementations of SA, we do not define a *stop condition* in PASA. Either a low T_0 value or a limit on the number of iterations is normally used as stop conditions in proposed SA-based solutions. In our scheme, it is possible under certain conditions that $T_0(i) \to 0$ for high i values, for example, in case of one event in the network. Since both network topology and data may change in time, this learning process can be run periodically to adapt the query dissemination to these variations.

In this work, we apply a more robust decision rule of the temperature update policy at the sink node with respect to our previous work [8]. In our previous version on PASA, we considered that the sink node takes advantage of the number of responses $\widehat{X}'$ obtained in the first iteration to apply the T adaptation policy. Instead, in this improved version, sink node applies a temperature update policy based on the *id* of those sensor nodes which report arguments in the first iteration. In this sense, sink decrements the temperature while those nodes report their readings and increments it otherwise, in order to avoid losing information.

5.2. Stateful Filtering. As for the stateless case, the goal of stateful scheme is to route queries towards nodes with relevant data to obtain the arguments for the computation of a given function. However, instead of centralizing learning on the sink node as just discussed, we consider a different scheme where sensor nodes can learn in a distributed fashion from previous decisions. The proposed scheme, also known as *Pheromone-based In-Network Processing (PhINP)* [9], implements an iterative procedure based on path reinforcement which results in search space reduction. Path reinforcement is achieved following a pheromone-based strategy similar to that used in ant colonies when searching for food [32], thus resulting in a probabilistic query routing towards nodes with relevant readings. PhINP, like PASA scheme, is a query-driven scheme conformed by two sequential phases, downstream forwarding and upstream learning, which are described in the following subsections.

Downstream Forwarding. Like in PASA, the query message is first broadcasted by the sink node to all its neighbors who, after processing the query, can decide to forward it to their neighbors, and so on. Even if nodes can receive the same query multiple times, they can at most forward it once. The message content was detailed in Section 4. In this case, each node maintains a state referred to as *pheromone level* λ whose value can range between 0 and 1. Nodes periodically decrement the value of λ by a factor λ_{dec} and can increase this value by a factor λ_{inc} under certain conditions but only after the upstream data collection process is over. The rate of the pheromone update policy is set up by the query process and in general it is assumed equal to the computation frequency (i.e., the frequency at which queries are disseminated). After each query iteration, some nodes tend to keep high pheromone levels, while others to decrease it. Queries are forwarded based on the pheromone level in a stochastic fashion such that

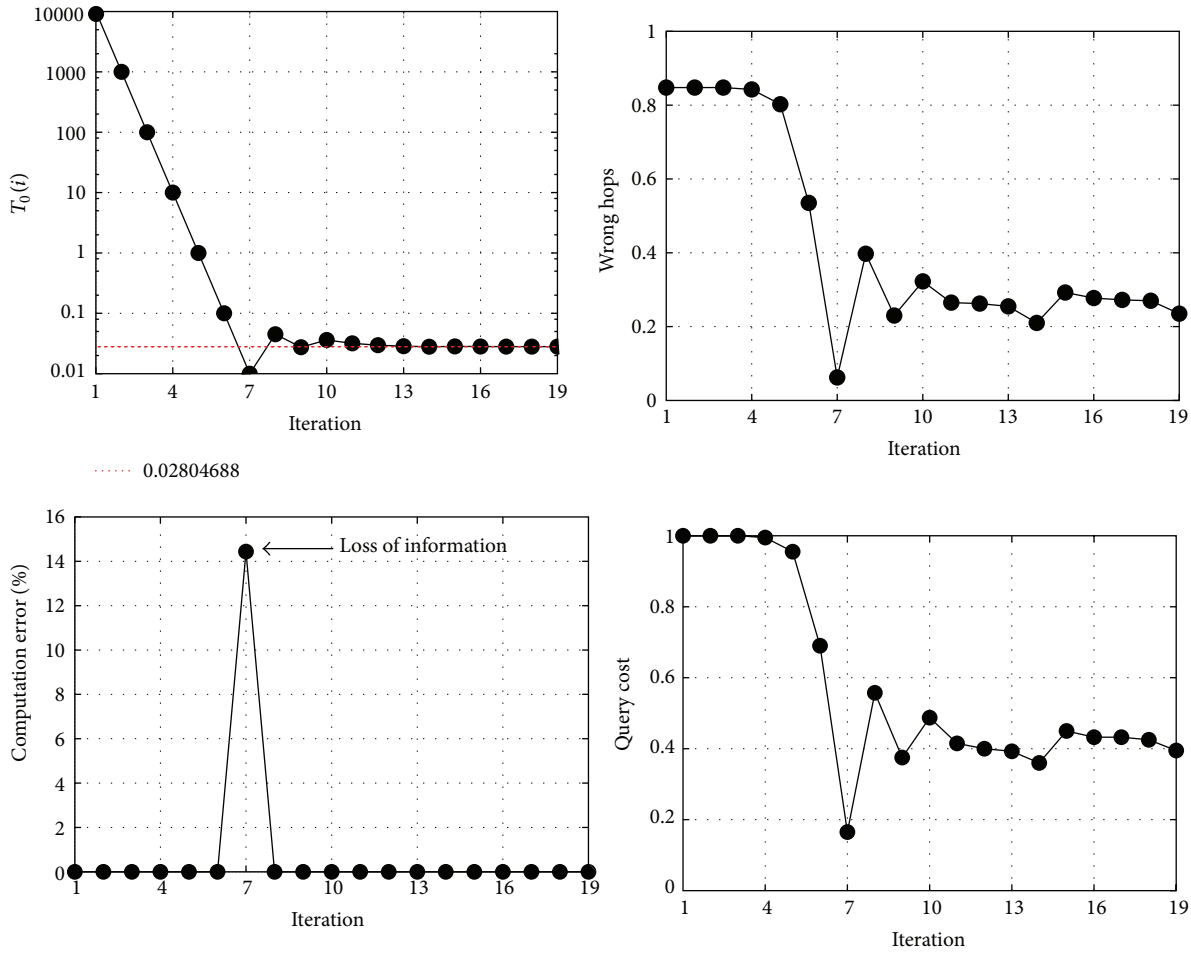


Figure 7: Stateless scheme convergence (400 nodes, 10 events).

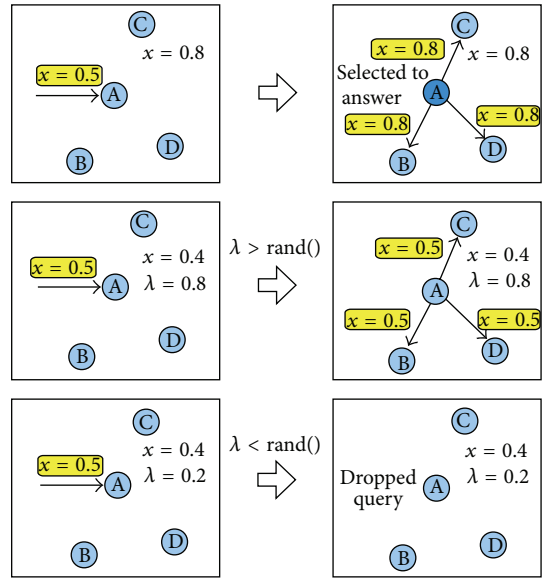


Figure 8: Downstream forwarding of the stateful scheme.

```
(1)  Input Received query message
(2)  Output Response and forwarding decisions
(3)  id ← Msg.getId()
(4)  thresholdValue ← Msg.getThreshold()
(5)  ΔE ← thresholdValue − sensedValue
(6)  // Discard already forwarded queries
(7)  if idTable[id] == true then
(8)      delete(Msg)
(9)  else
(10)     updateIdTable(id)
(11) end if
(12) // Query forwarding decision
(13) If ΔE ≤ 0 then
(14)     Msg ← setValue(sensedValue)
(15)     send(Msg)
(16) else
(17)     if λ > rand() then
(18)         Msg ← setValue(thresholdValue)
(19)         send(Msg)
(20)     end if
(21) else
(22)     delete(Msg)
(23) end if
```

Algorithm 2: Stateful Forwarding Scheme (PhINP).

```
(1) if λ > λmin then
(2)     λ = λ − λdec
(3) end if
(4) if answerQuery == true or forwardResp == true then
(5)     if λ < λmax then
(6)         λ = λ + λinc
(7)     end if
(8) end if
```

Algorithm 3: Pheromone update policy.

the more the pheromone is available, the more likely the message is forwarded. This process is described in Algorithm 2. The resulting node behaviors are shown in Figure 8, in which quite extreme conditions are considered (e.g., $\lambda = 0.2$ and $\lambda = 0.8$). The first query message sent by the sink encourages all nodes to participate in the forwarding phase by setting their pheromone level to its maximum value (i.e., $\lambda = 1$). This guarantees that all nodes receive the query and that all relevant data are properly selected during startup. Nodes can decrease their pheromone level up to a lower bound given by $\lambda_{\min}$ which determines the minimum exploration probability of searching for new data. Moreover, nodes can increase their pheromone level up to a maximal value of $\lambda_{\max}$ normally set to 1.

Upstream Learning. Only nodes which have been either selected to answer the query or have forwarded a response message to the sink node increase their pheromone level by λ_{inc} up to an upper bound given by $\lambda_{\max}$, as is described in Algorithm 3. In this way, paths from the sink node towards nodes with relevant data can be reinforced. On the following iterations, queries will be routed mostly over these paths, thus avoiding exploration of those areas where irrelevant data are assumed. To simplify the notation, we define a configuration vector $[\lambda_{\max}\ \ \lambda_{\text{inc}}\ \ \lambda_{\text{dec}}\ \ \lambda_{\min}]$ which is set in each sensor node and whose values are a function of the nature of the physical phenomena to be monitored. In this work, a reference configuration vector [1 0.2 0.1 0.1] is in general assumed.

In this work, we improve further the downstream forwarding decision with respect to our previous versions [9, 33, 34] to increase its robustness to changes in the sensed field. This is implemented by always forwarding queries if relevant data are present (i.e., $\Delta E \leq 0$), despite the pheromone level, which enhances the detection of new events.

Table 1: Simulation parameter setting.

Parameter	Value
Deployment area	200×200, 283×283, 400×400 m
Network size	100, 200, 400 nodes
Node density	2.5×10^{-3} nodes/sq m
Communication range	Circular, 50 m
Average neighbors per node	19.635
Node failure probability	0, 1, 5, 10%
Number of data sources	1 to 400
Movement of data sources	20 to 200% of comm. range
Number of simulations	2000
Random generator	Mersenne Twister

6. Simulation Results

The proposed schemes were evaluated in simulation environments developed in Omnet++ [35], where three metrics were analyzed to assess their performance under different scenarios: computation error, probability of success, and communication cost. Computation error represents the relative error resulting from the computed function at the sink and the actual optimal one. On the other hand, the probability of success represents the probability of computing the optimal value (i.e., zero relative error). In this sense, in case of computing the *max* function, this metric represents the probability of finding the node with the maximal reading in the network. Communication cost considers average number of nodes involved in forwarding both the query message (query cost) and the relevant readings to the sink (response cost). Note that the maximum query cost is equal to 1, which means that all nodes forwarded the query once (i.e., flooding). As each selected node sends its reading to the sink by a single path, this response cost is negligible with respect to the query cost. For this reason, we focus on the query communication cost only. Under the assumption that each sensor node can forward once the query in each iteration, the query cost can be also defined as the number of packets transmitted at each iteration. Note that this metric is closely related to the consumed energy in the network. The parameter set used in the simulations is shown in Table 1. We evaluate several aspects of the proposed schemes, in order to compare their performance and to determine the best

application fields of each scheme. Our analysis includes the algorithm convergence, event analysis, the robustness to node failure, packet loss and dynamic events, and the capability of readaptation to event changes in the sensor network. Finally, a comparison between the proposed schemes is addressed. These aspects are described in detail in the following sections.

6.1. Convergence. Since the stateless and stateful schemes learn in time, it is expected that their metrics will experience some variations during first iterations. In case of the stateless scheme, it needs to converge to a T_0 value, while for stateful one, the pheromone level needs to get stabilized on nodes. To simplify the analysis, we consider a network where both nodes and links are ideal; thus, nodes cannot fail and links are loss free. Besides, we assume that the sensed field, formed by 10 random data sources or events with the same amplitude following the diffusion law of heat ($\alpha = 1$), does not change neither their amplitudes nor their positions while the scheme converges.

In the stateless scheme, the sink node sets a high T_0 value to the query for the first iteration ($T_0(1) = 10000$ is in general considered in this work), in order to reach each sensor node in the network. Based on the set nodes which report data in the first iteration, the sink node can estimate if there is any information loss in the following iteration and adjust the $T_0(i)$ value in query to avoid it. As discussed in Section 5.1, this is based on reinforcement learning. The objective is to reduce the query dissemination cost, while there is no information loss. Due to the probabilistic nature of the proposed stateless scheme, it can incur in very low computation errors once the algorithm has converged, near 3% in case of this scenario. The computation error and query cost metrics obtained by simulation in this scenario are shown in Figures 9(a) and 9(b), respectively.

On the other hand, since the stateful scheme also floods the network in the first iteration, the computation error remains null in the following iterations, considering no major data changes in the sensed field. Under these conditions, the convergence of the algorithm can be analyzed as it tries to decrease the query cost while keeping the error null (i.e., retrieving the same relevant data) as shown in Figures 9(c) and 9(d). Indeed, the first iteration has always a cost equal to 1 as every node forwards the query once. After each iteration, paths to nodes with relevant data are reinforced based on local pheromone levels, and the probability P of using other paths is reduced, minimizing the communication cost. Each node's behaviour is described by 4 fixed parameters (λ_{max}, λ_{min}, λ_{inc}, and λ_{dec}) and a variable one λ, all of them ranging from 0 to 1. The collaborative work of the whole network defines in a decentralized way the performance of this scheme. In this analysis, different values of λ_{dec}, λ_{inc}, and λ_{min} are considered in order to understand the operation of this scheme. The larger these values, the faster the convergence to a minimum cost configuration; however, this cost tends to be higher for larger λ_{dec} and λ_{inc} values. A trade-off does exist between convergence time and this minimum cost. The value of λ_{max} is always set to 1 in order to avoid errors in the computation. The behavior of this scheme for several configuration values and network sizes is shown in Figures 9(c) and 9(d). Note that the performance of this distributed algorithm is independent of the network size.

From simulations we can see that the query cost for the stateless scheme can be decreased to more than 50% with respect to flooding while still keeping the error bounded to less than 2.5% with the considered scenario. Note that the stateful scheme can improve the communication cost with respect to stateless one by using lower λ_{min} values, at expense of lower robustness to dynamic events, as a consequence of the lower probability to escape from local minima, as we will discuss in following subsections. If $\lambda_{min} \to 0$, then the communication query cost and the response cost tend to have the same value, since the same path is used to forward both queries and responses.

Moreover, the stateless scheme tends to introduce more errors than the stateful one. However, the latter requires good synchronization of the setting parameters due to its distributed nature. In the stateless scheme, this condition is relaxed since the sink node configures and sends the cooling policy into the query.

6.2. Event Analysis. In this case, we evaluate the performance of the stateless and stateful schemes for different fields, that is, from the point of view of data distribution in the network. In this sense, we define the *event-sensor-rate* metric (esr), which is the relationship between data sources and sensor nodes presented in the network. In this sense, if we have n sensor nodes and e events (i.e., sources of information), the esr factor can be defined as $esr = e/n$. Normally, events and nodes are not at the same position, thus, we assume an independent random distribution for each of them. Moreover, to simplify the analysis, we consider that all events have the same amplitude and diffusion coefficient. Simulations with esr values ranging from 0.01 to 1 and configurations of $T_0(1) = 10000$ for stateless and [1 0.2 0.1 0.1] for stateful scheme are used for this purpose. Results obtained by simulation are shown in Figure 10. We can see that the stateless scheme has a very low computation error, which is normally the lower possible when $esr \to 0$, that is, the case of one or few events. On the other side, an $esr \approx 0.1$ introduces the higher error, in case of stateless scheme, with the higher communication cost in both schemes. As a consequence, we defined a scenario with 10 events as the case study in this work. Also, we can see that communication cost is rapidly reduced when $esr \to 1$ due to the filtering process in the network.

6.3. One-Event Detection. In this case, the capacity to detect an event randomly located in the network is analyzed. Note that this analysis is a particular case ($esr \to 0$) of the previous one. However, the goal is to compare the performance of the stateless and stateful schemes with respect to two traditional mechanisms such as *flooding* and *gossip*. The gossip mechanism is set with a *gossip probability = 0.25* in order to have the same query cost as the proposed schemes. The performance of these schemes is shown in Figure 11. We can see that flooding can always find those nodes with relevant arguments to compute the function, but it incurs in

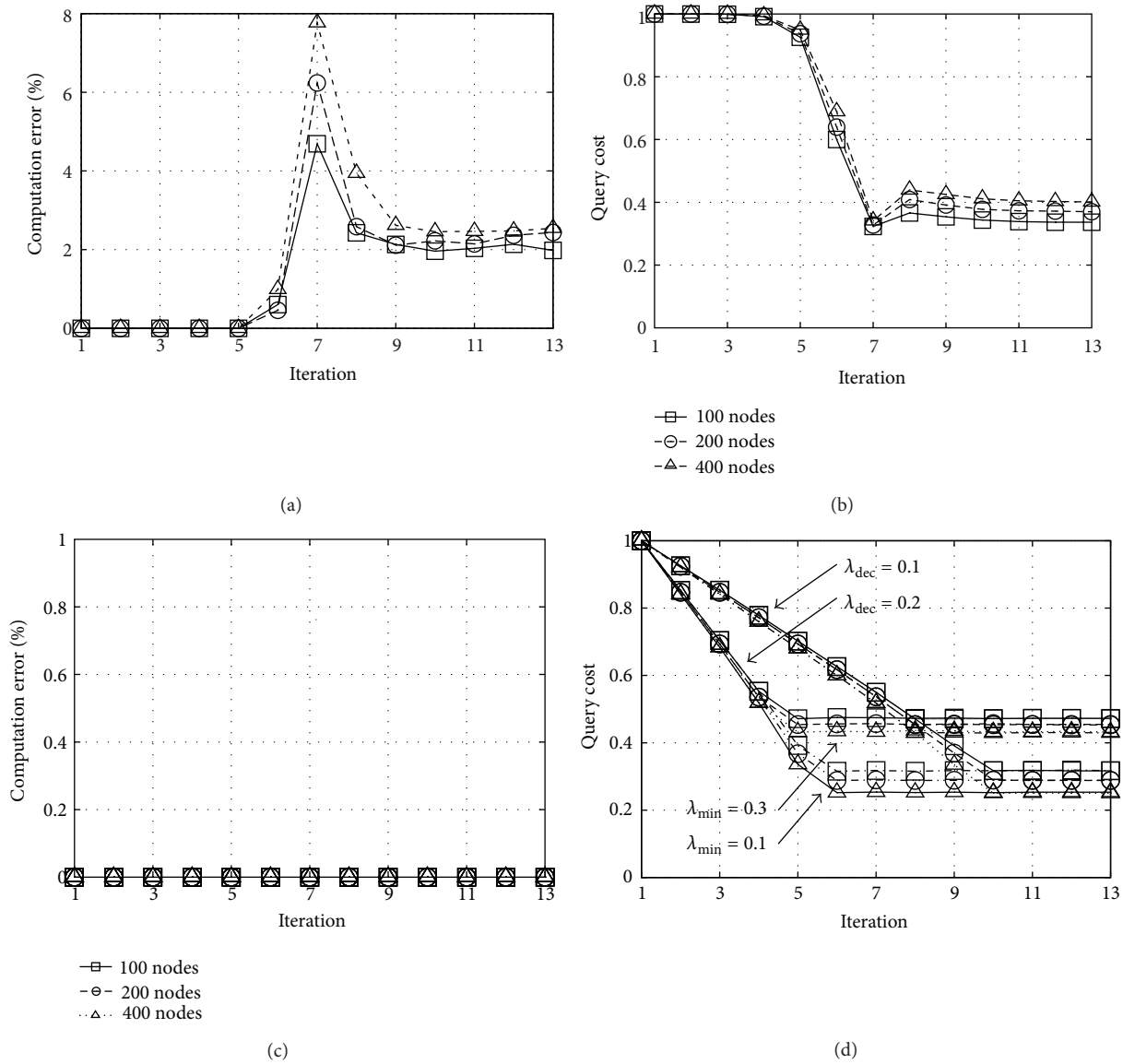


Figure 9: Convergence: stateless ((a), (b)) and stateful ((c), (d)) schemes.

a high communication cost, because each node has to forward the query (i.e., query cost = 1). Note that the performance of stateless and stateful schemes falls between the performance obtained by flooding and gossip mechanisms. On one hand, the proposed schemes incur in lower errors than gossip and, on the other hand, can reduce the communication cost greatly, in this case a reduction of 80%.

6.4. Robustness to Node Failure. Once the stateless and stateful schemes have converged after few iterations, it becomes critical for them to still work under faulty conditions. According to this, we consider the case of nodes with some failure probability. Figure 12 shows the performance of both schemes for different probabilities of node failure as the network size increases. To simplify the analysis, we suppose that the sensed field is static, thus, the sensed values do not change over time.

As expected, the stateful scheme, although has lower computation error in the network size range analyzed, is more susceptible to failure of nodes than the stateless scheme. The reason behind this is that the stateful one tends to maintain a single path (i.e., pheromone trail) for the query dissemination between sink and each of the nodes that provide relevant arguments to compute the function at the sink. In this sense, the failure of a forwarding node is critical, since this can affect the computation of a function. As we will see, this behavior of the stateful scheme to distributively form paths can be relaxed, as the width of paths is a function of the $\lambda_{\min}$ value. With the term *width*, we refer to the multipath characteristic of query dissemination. With a low value of $\lambda_{\min}$, that is, $\lambda_{\min} \rightarrow 0$, a unique single path is formed between sink and a selected sensor node. As $\lambda_{\min}$ increases, each disseminated query arrives to a node through more incoming paths (i.e., multiple paths). This effect is depicted in Figure 13 for a better

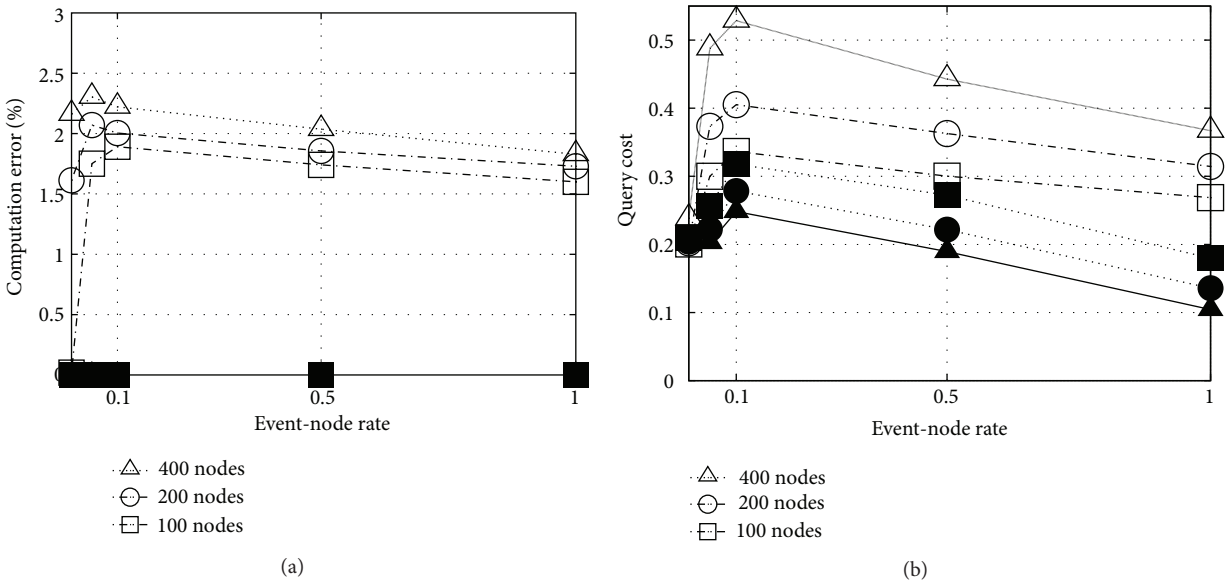


Figure 10: Event detection. Stateless (white) and stateful (black).

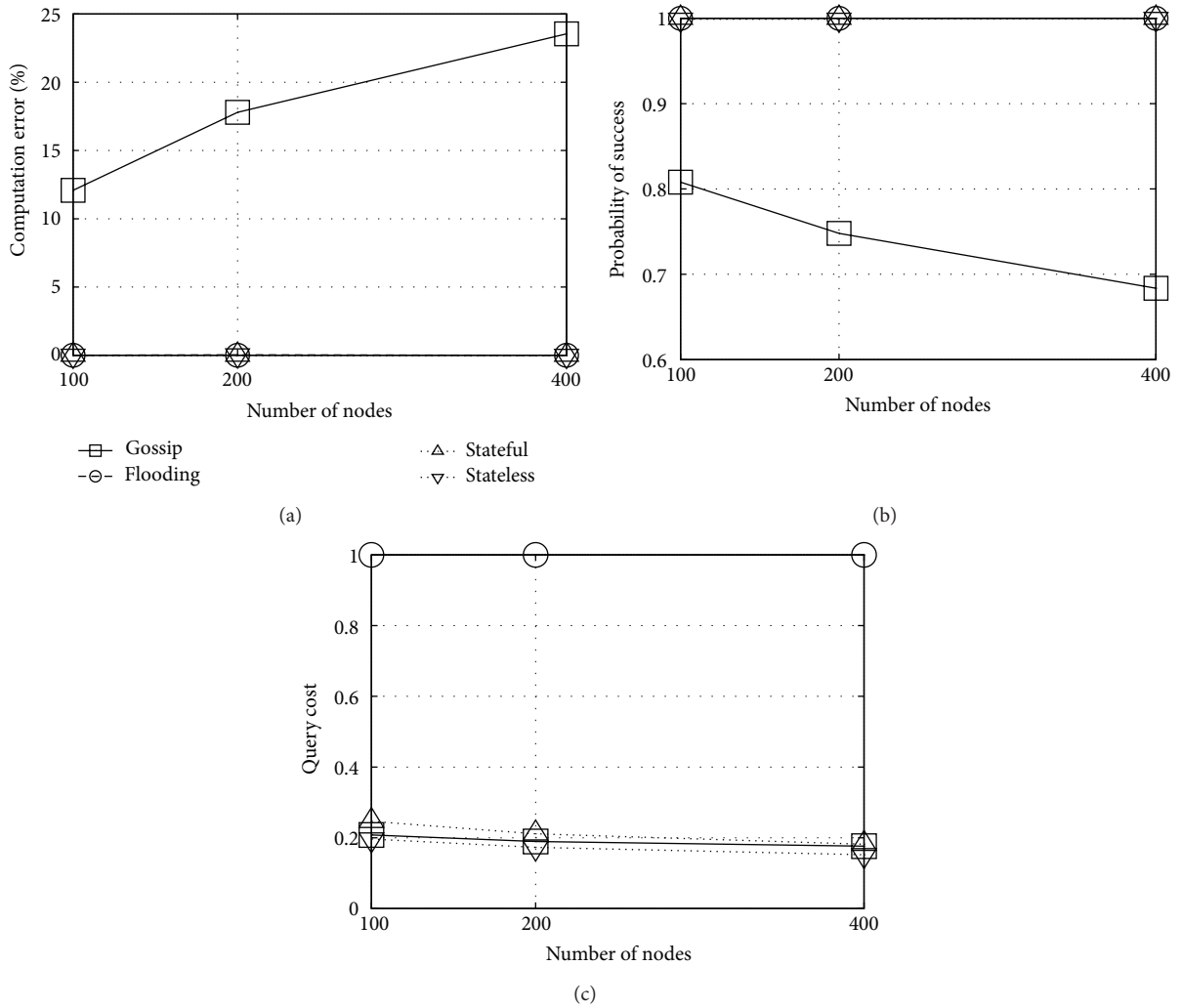


Figure 11: Capacity to detect a randomly deployed event.

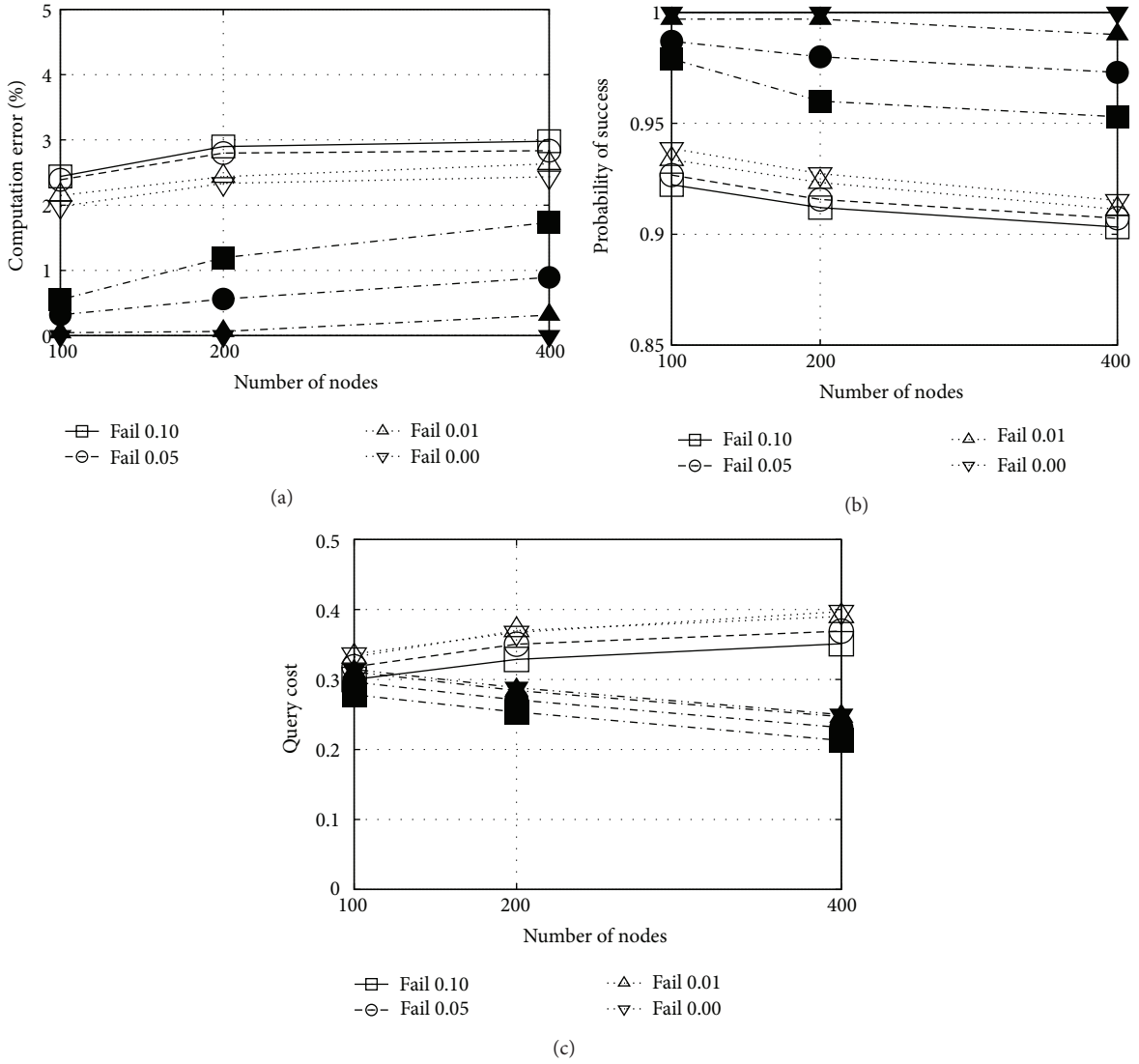


Figure 12: Lossy network. Stateless (white) and stateful (black).

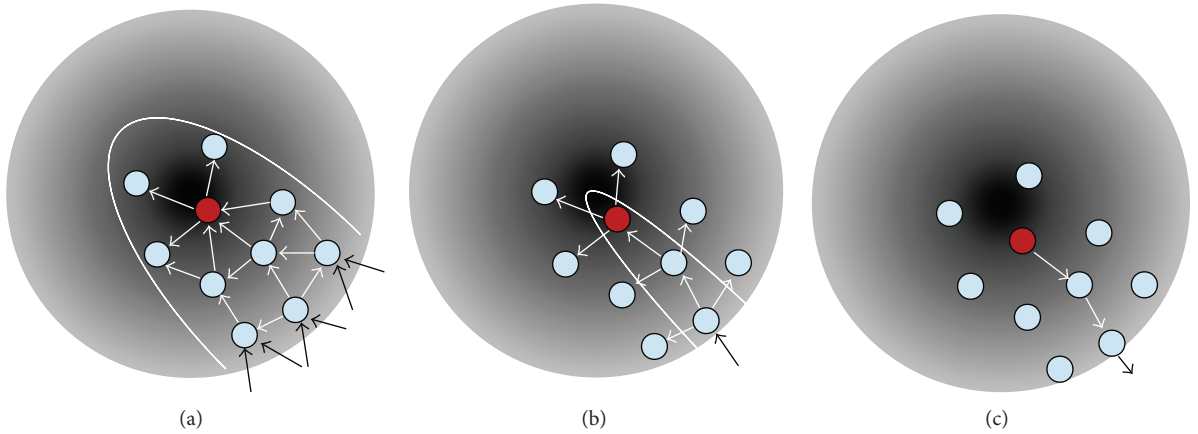


Figure 13: Downstream forwarding: stateless (a) and stateful (b) schemes and upstream forwarding (c) after convergence.

understanding. In case of the stateless scheme, the query arrives through multiple incoming paths. So, a failure in an forwarding node is not critical (see Figure 13(a)). Instead, in case of stateful scheme, if a low $\lambda_{\min}$ value is used, the scheme after convergence forms narrow paths which are more affected by failures in nodes as depicted in Figure 13(b). For higher $\lambda_{\min}$ values, the stateful scheme tends to have similar behavior from the point of view of incoming query messages as the stateless one, at the expense of a higher communication cost. A more detailed description is given in Section 6.8. As explained above, both schemes report the arguments of selected nodes by using the inverse path of the first arrival path as shown in Figure 13(c).

In order to increase the robustness of the proposed schemes to node failures, a simple *self-healing mechanism* is implemented to avoid loss of information. In this sense, each node sending information in upstream forwarding, based on reinforcement learning by active hearing, can notice if the transmitted packet was forwarded by its one-hop neighbor towards the sink. The same mechanism is followed by intermediate forwarding nodes in the path. When a sending node realizes that its one-hop destination neighbor does not forward the response packet (see Figure 14(a)), it sends the same response to the following neighbor of its neighbor's vector as depicted in Figure 14(b). In this respect, as seen above, the neighbor's vector is constructed by each sensor node in each query iteration. This is an ordered array in which the *ids* of one-hop neighbor nodes sending the query are stored. Note that the stateless scheme can also benefit from this improvement, not so for the case of stateful schemes set with low $\lambda_{\min}$ values.

6.5. Robustness to Loss of Packets. In this case, the robustness of the stateless and stateful schemes to packet loss is analyzed. The same scenario and parameters setting as in the case of node failure analysis are considered. Results are shown in Figure 15. As expected, the stateful scheme is more sensible to packet loss than the stateless one, since it is able to conform paths using fewer query forwarding nodes.

6.6. Robustness to Dynamic Events. Even if the proposed schemes have shown to be robust enough to lossy networks, it becomes also crucial to analyze their behavior as the sensed field changes in time. For this purpose, we consider now a loss free network where all field sources (events) may change their positions. We are interested in evaluating if, after the schemes have converged to a given sensed field, they can adapt to a different one which has still some correlation with the original one. In this sense, a random position change is inserted to each event in the network, which is defined as a percentage of the communication range. The simulation considers the case of 10 events in the network, and the statistics are reported just after the change, that is, in the following iteration without considering readaptation as will be analyzed above for the stateful scheme. Results obtained for several position change values are shown in Figure 16. More clearly, if the communication range r of sensor nodes is fixed to 50 meters, a position change of 100% implies that each event in the network is randomly moved at most the same value of the communication range from its initial position. In this sense, a change of 20% has the effect that the node with the best reading to report to the sink (i.e., near to the event) could be the same; instead with a change of 200% the node with the best reading is usually another. Based on the simulation results, we can see that both schemes have similar tendencies as the percentage of position change of events is increased.

6.7. Readaptation Capability. As an extension of the previous analysis, in which we only report the metrics after applying the changes, we analyze the readaptation capability of the stateful scheme in several iterations after those changes. Recall that the stateless scheme does not have this readaptation feature due to the learning process followed by the centralized control of sink. As seen above, the configuration parameters λ_{inc} and λ_{dec} define the dynamic of this scheme. In this case, a scenario with 50 randomly deployed events was considered. The scheme was configured as $[1 \;\; X \;\; 0.2 \;\; 0.1]$ and the results obtained through simulations are shown in Figure 17. Intuitively, a user would try to set the λ_{inc} value as high as possible, but this depends on the application.

6.8. Schemes Comparison. Based on the discussed results, we can notice that each scheme has its own scope; that is, each scheme is more efficient for a given scenario. We make this analysis considering that both schemes have already converged to a low-energy state. In this sense, the stateful scheme is more appropriate than the stateless one in scenarios with dynamic events. However, the stateless scheme is more robust to failure of sensor nodes due to the fact that the same query message more probably arrives to a node through multiple paths. In this sense, in the stateful scheme there is a trade-off between the communication cost and the robustness to node failure. If a low $\lambda_{\min}$ value is set, the scheme tends to a low-energy state, with a minimal communication cost, at the expense of a low robustness to node failure, and vice versa. Figure 18 shows the queried areas for both schemes after the convergence in case of max function computation. In this case, a low $\lambda_{\min}$ value was set in the stateful scheme in order to obtain well-defined paths between sink (red node) and selected nodes (white nodes) to provide arguments to compute the function. In case of setting a large $\lambda_{\min}$ value, these paths are widened, and, as consequence, the queried area tends to resemble that of the stateless scheme.

A substantial difference between both schemes is the probability of exploration of sensor nodes (i.e., query forwarding) related to the neighbor-filtering rule. In the stateful scheme this probability is limited to a $\lambda_{\min}$ value which is independent of the position of the node. Instead, in the stateless scheme, this probability is a function of the distance to the sink, so the farther the node, the lower the probability of forwarding the query (i.e., exploration). This is due to the degradation of the temperature T of a query when it is forwarded through the network using a centralized control. In this sense, the temperature $T_0(i)$ inside the query, which is set by the sink node when the query is injected to

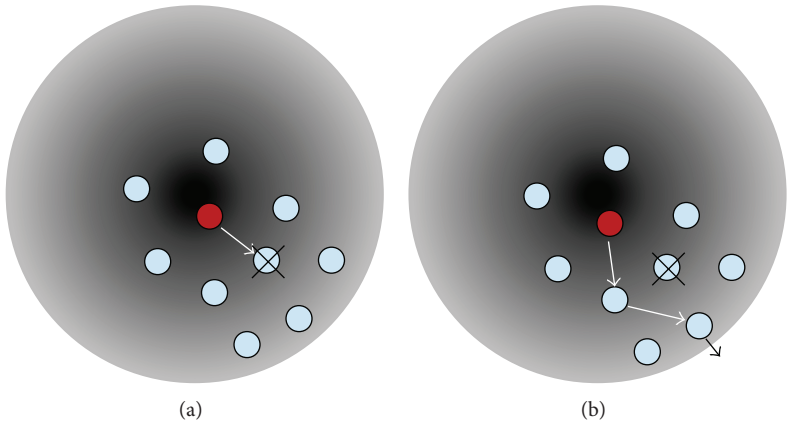


Figure 14: Self-healing mechanism to deal with node failures.

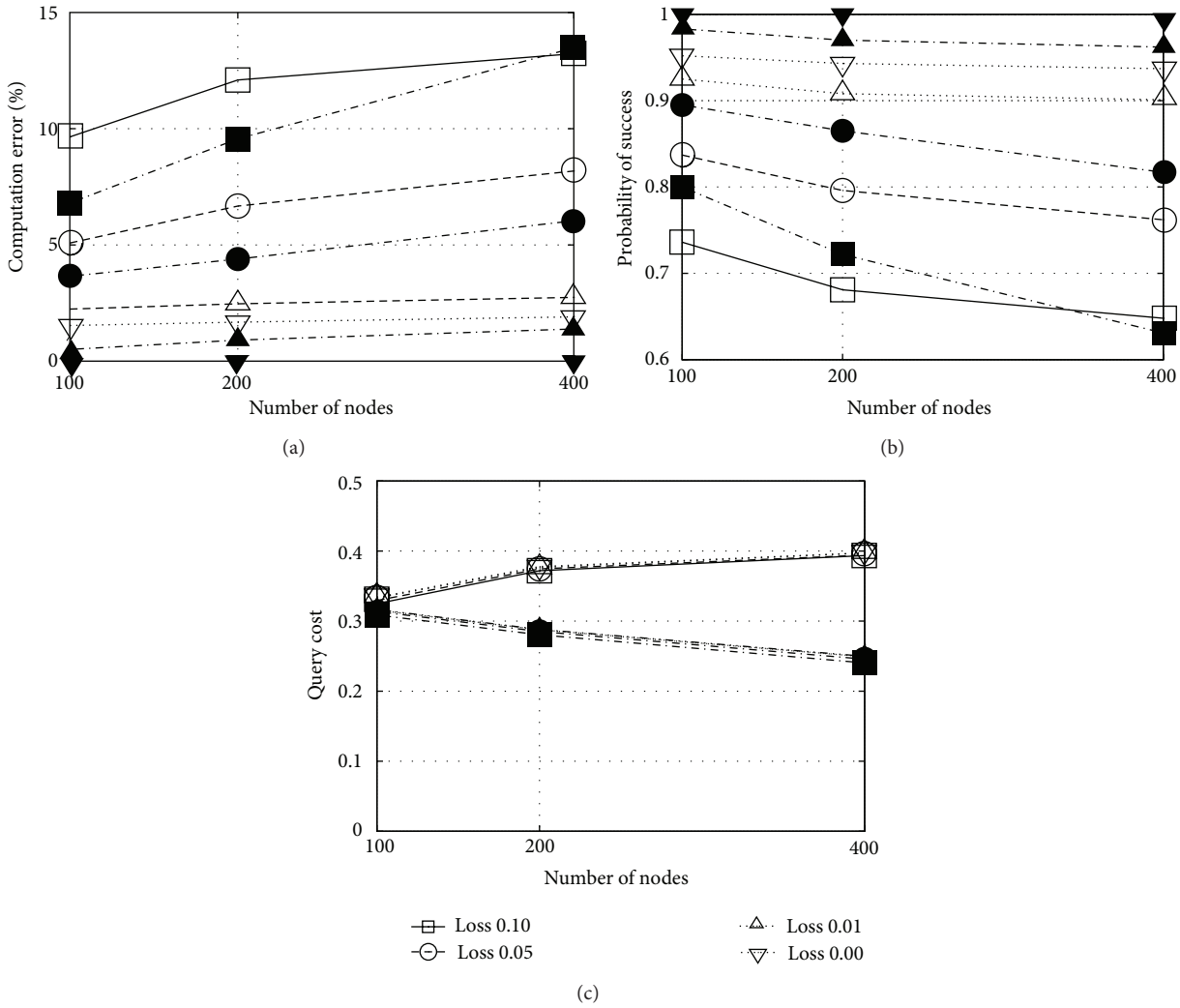


Figure 15: Loss of packets. Stateless (white) and stateful (black).

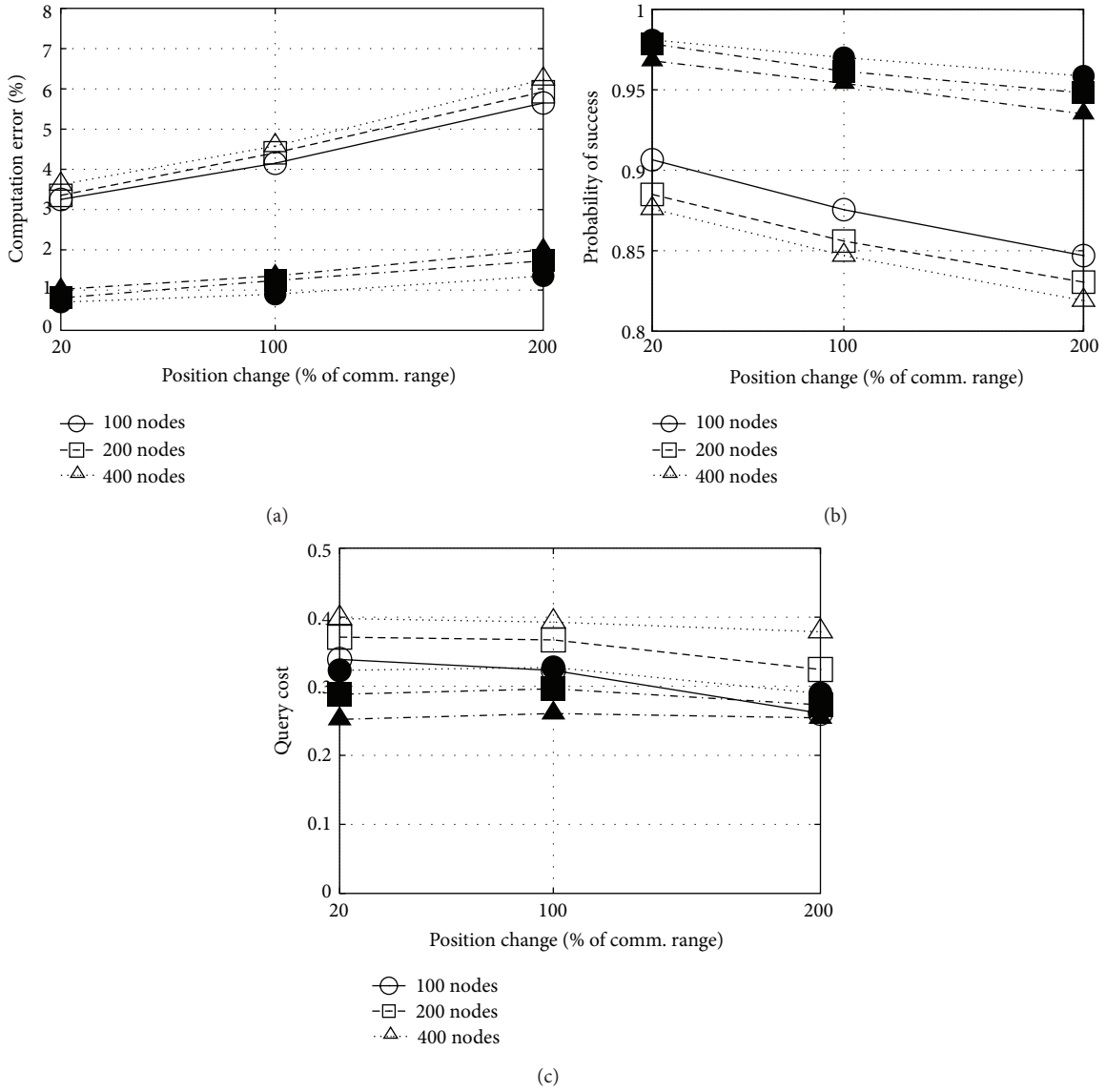


Figure 16: Dynamic events. Stateless (white) and stateful (black).

the network, is the highest that it can have during iteration i. This is a consequence of a centralized control. This feature makes more reliable stateful scheme for large-scale WSNs. A summary of the main features and setting parameters of the proposed schemes is shown in Table 2.

7. Conclusion

In this work, we analyzed the problem of in-network filtering to compute efficiently type-threshold functions in a WSN, minimizing both communication cost and computation error. In this context, in-network filtering schemes can be used to forward only relevant data towards a sink node for processing purposes as an alternative to data aggregation. To this end, two nature-inspired schemes were proposed that can drive this filtering process. The first one considered a stateless filtering scheme where the sink node implemented a learning algorithm to feed a decision rule based on the well-known simulated annealing search process. Instead, the second approach, dubbed stateful scheme, considered a distributed learning scheme inspired by the ant colony

Table 2: Comparison of proposed schemes.

Feature	Scheme	
	Stateless (PASA)	Stateful (PhINP)
Type of control	Centralized	Distributed
Control parameter	T	λ
Parameter location	Inside the query	Local at each node
Flooding behavior	$T \gg$	$\lambda_{\min} = 1$ or $\lambda_{\text{dec}} = 0$
Convergence speed	Logarithmic, linear D	$\lambda_{\text{inc}}, \lambda_{\text{dec}}, \lambda_{\min}$
Readaptation capability	No	Yes
Readaptation speed	—	$\lambda_{\text{inc}}/\lambda_{\text{dec}}$

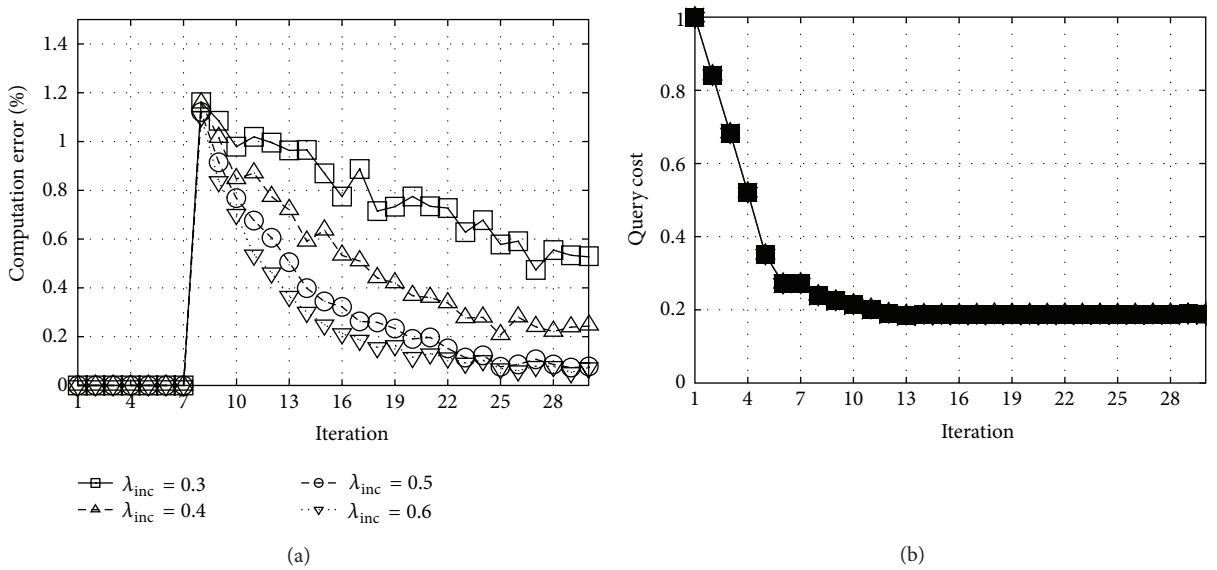


Figure 17: Readaptation capability. Stateful scheme.

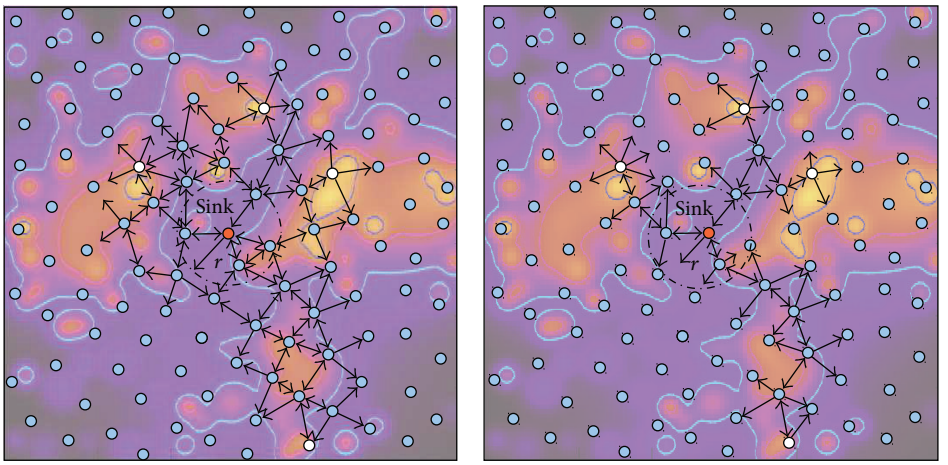


Figure 18: Queried area. Stateful (a) and stateful (b).

behavior where nodes kept a state to reinforce paths to the sink. We show by simulation that communication costs can be significantly reduced with respect to traditional schemes while keeping the computation error bounded.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially funded by Universidad Tecnológica Nacional, FONCyT IP-PRH Postgraduate Grant Program, SECYT-UNC Research Program, and CONICET Postgraduate Grant.

References

[1] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] 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.

[4] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[5] R. Rajagopalan and P. Varshney, "Data-aggregation techniques in sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

[6] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 755–764, 2005.

[7] A. Giridhar and P. R. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 98–107, 2006.

[8] G. Riva and J. Finochietto, "A parallel and adaptative query routing scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 243–247, Ottawa, Canada, June 2012.

[9] G. G. Riva and J. M. Finochietto, "Pheromone-based in-network processing for wireless sensor network monitoring systems," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 6560–6564, Ottawa, Canada, June 2012.

[10] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748–2761, 2009.

[11] M. Franceschelli, A. Giua, and C. Seatzu, "Distributed averaging in sensor networks based on broadcast gossip algorithms," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 808–817, 2011.

[12] A. Meliou, C. Guestrin, and J. M. Hellerstein, "Approximating sensor network queries using in-network summaries," in *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN '09)*, pp. 229–240, San Francisco, Calif, USA, April 2009.

[13] Y. Xu, T. Fu, W. Lee, and J. Winter, "Processing K nearest neighbor queries in location-aware sensor networks," *Signal Processing*, vol. 87, no. 12, pp. 2861–2881, 2007.

[14] Y. Xu, W. Lee, J. Xu, and G. Mitchell, "Processing window queries in wireless sensor networks," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pp. 270–280, Atlanta, Ga, USA, April 2006.

[15] H. Huang, J. Hartman, and T. Hurst, "Efficient and robust query processing for mobile wireless sensor networks," *International Journal of Sensor Networks*, vol. 2, no. 1-2, pp. 99–107, 2006.

[16] J. Ahn, S. Kapadia, S. Pattem, A. Sridharan, M. Zuniga, and J. Jun, "Empirical evaluation of querying mechanisms for unstructured wireless sensor networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, pp. 17–26, 2008.

[17] J. Faruque and A. Helmy, "RUGGED: Routing on fingerprint gradients in sensor networks," in *Proceedings of the IEEE International Conference on Pervasive Services (ICPS '04)*, pp. 179–188, Novi Sad, Servia and Montenegro, July 2004.

[18] J. Zhang, X. Zhu, and H. Peng, "Bi-filtered forwarding: a quasi-optimal routing algorithm for query delivery in wireless sensor networks," *International Journal on Smart Sensing and Intelligent Systems*, vol. 6, no. 3, pp. 993–1011, 2013.

[19] J. Heidemann, F. Silva, and D. Estrin, "Matching data dissemination algorithms to application requirements," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 218–229, November 2003.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] A. Zimmerman and J. Lynch, "A parallel simulated annealing architecture for model updating in wireless sensor networks," *IEEE Sensors Journal*, vol. 9, no. 11, pp. 1503–1510, 2009.

[22] R. V. Kulkarni, A. Förster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 1, pp. 68–96, 2011.

[23] K. A. Yau, P. Komisarczuk, and P. D. Teal, "Reinforcement learning for context awareness and intelligence in wireless networks: review, new features and open issues," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 253–267, 2012.

[24] F. Dressler and O. B. Akan, "A survey on bio-inspired networking," *Computer Networks*, vol. 54, no. 6, pp. 881–900, 2010.

[25] M. Saleem, G. A. Di Caro, and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions," *Information Sciences*, vol. 181, no. 20, pp. 4597–4624, 2011.

[26] A. M. Zungeru, L. Ang, and K. P. Seng, "Classical and swarm intelligence based routing protocols for wireless sensor networks: a survey and comparison," *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1508–1536, 2012.

[27] B. Park, S. Park, E. Lee, S. Noh, and S. Kim, "Large-scale phenomena monitoring scheme in wireless sensor networks," in *Proceeding of the 71st IEEE Vehicular Technology Conference (VTC '10)*, pp. 1–5, Taipei, Taiwan, May 2010.

[28] M. Li and Y. Liu, "Iso-Map: energy-efficient contour mapping in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 699–710, 2010.

[29] J. Sun, "Multi-threshold based data gathering algorithms for wireless sensor networks," *Journal of Networks*, vol. 4, no. 1, pp. 30–41, 2009.

[30] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[31] Z. Cheng and W. B. Heinzelman, "Flooding strategy for target discovery in wireless networks," *Wireless Networks*, vol. 11, no. 5, pp. 607–618, 2005.

[32] L. F. M. Vieira, U. Lee, and M. Gerla, "Phero-trail: A bio-inspired location service for mobile underwater sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 4, pp. 553–563, 2010.

[33] G. Riva and J. Finochietto, "Pheromone-based in-network processing for wireless sensor network monitoring systems," *Journal of Network Protocols and Algorithms*, vol. 4, no. 4, pp. 156–173, 2012.

[34] G. G. Riva, J. M. Finochietto, and G. Leguizamon, "Bio-inspired in-network filtering for wireless sensor monitoring systems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 3379–3386, Cancun, Mexico, June 2013.

[35] Omnet ++ simulation library, http://www.omnetpp.org/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 348606, 7 pages
http://dx.doi.org/10.1155/2014/348606

Research Article

Web Spider Defense Technique in Wireless Sensor Networks

Alejandro Canovas,[1] Jaime Lloret,[1] Elsa Macias,[2] and Alvaro Suarez[2]

[1] *Integrated Management Coastal Research Institute, Universidad Politécnica de Valencia, Spain*
[2] *Departamento de Ingeniería Telemática, Universidad de Las Palmas de Gran Canaria, Spain*

Correspondence should be addressed to Alejandro Canovas; alcasol@posgrado.upv.es

Received 22 April 2014; Accepted 1 July 2014; Published 23 July 2014

Academic Editor: S. Khan

Wireless sensor networks (WSNs) are currently widely used in many environments. Some of them gather many critical data, which should be protected from intruders. Generally, when an intruder is detected in the WSN, its connection is immediately stopped. But this way does not let the network administrator gather information about the attacker and/or its purposes. In this paper, we present a bioinspired system that uses the procedure taken by the web spider when it wants to catch its prey. We will explain how all steps performed by the web spider are included in our system and we will detail the algorithm and protocol procedure. A real test bench has been implemented in order to validate our system. It shows the performance for different response times, the CPU and RAM consumption, and the average and maximum values for *ping* and *tracert* time responses using constant delay and exponential jitter.

1. Introduction

A wireless sensor network (WSN) is distributed in nature. It consists of several electronic devices with a memory, a processor, and one or more elements that sense the environment [1, 2]. One of their main issues taken into account in their deployment is their power limitation and their need to save energy [3, 4]. Sensor nodes can communicate among them using a particular or standard communication technology network interface card. The sensed values can be forwarded to a central manager that usually is a computer (or similar device). The computer allocates a manager that is in charge to manage the WSN. The most common strategy to read values of sensed elements consists of interrogating the manager in order to obtain a set of sensed values. In this sense, a WSN is used to collect and monitor the related information about a specific environment. This procedure has relevance in several cases: vigilance, oceanographic values of a strategic installation, police related information, and many more.

Generally, WSNs are used to sense private data. Some of them can also transmit critical data. Thus, it is very important to secure the collection of data and detect and avoid external intrusions. An intruder may be able to access unauthorized data, spread erroneous data and/or malicious code, implement unauthorized changes to data or sensor software, or steal data. Moreover, an intruder could initiate attacks to the network from that sensor node and open new doors to other intruders.

This must be done taking into consideration four requirements: data confidentiality, data authentication, data integrity, and denial-of-service (DoS) attack avoidance [5]. Different surveys on WSN security are presented in [6–13]. In [14, 15] a list of attacks and counterattacks are surveyed.

This work is focused on intrusion attacks in WSNs. A network intrusion detection system (IDS) is an essential element in a computer security strategy [16]. An IDS is a device or a software application that monitors network and system activities for malicious activities or policy violations. The IDS produces reports to a central system that allow humans to intervene or that can be responded by computer systems in an attempt to stop the intrusion. In a WSN, this attack means that an attacker (malicious user) wants to illegally read the data sensed by a set of sensors. We suppose the malicious user can interrogate the sensors in the WSN bypassing the control of the WSN manager/administrator. The difficult task here is to discover when a malicious attack is

happening and which the particular properties of the attacker are. The main idea behind this is that the IDS can learn about the attacker in order to prevent future attacks. Several techniques have been used to design the IDS.

(i) The intrusion detection policy proposed in [17] monitors the communication between neighboring nodes and finds those nodes that are not working normally. Some general rules are defined to detect such nodes, which are called compromised nodes. They simulated transport and routing layer in order to analyze the performance of the proposed policy. They showed that each node should be treated independently in the WSN, and purely centralized detection schemes may fail to identify the network behavior whether it is normal or it is under any attack.

(ii) Due to the huge volume of network traffic, coding the rules becomes difficult and time-consuming. Data mining techniques, used for example, in anomaly based systems [18], can build network intrusion detection models adaptively. They can analyze and predict the behaviors of users in order to know if these behaviors are attacks or a normal behavior.

(iii) The traffic prediction can also be used to model a mechanism against any intrusion detection. In [19] it is shown that, by inspecting received packet features, a sensor can identify an intruder impersonating a legitimate neighbor.

(iv) A honeypot is usually a valuable surveillance tool that provides early warnings to system administrator about the trends of malicious activity in the WSN. A wireless honeypot can be used to gather information about the intruder in the WSN, taking into account several implementation techniques for wireless local area network [20]. In a WSN a fake access point could be implemented by a sensor that responds with fake data to the intruder. A very interesting survey that includes results of honeypot technology applied to WSN can be found in [21]. The information sensitivity, resources, and time are the most important factors in choosing the type of honeypot for any WSN. We differentiate two types of honeypots: (a) low-interaction, which only monitors for anomalies, and (b) high-interaction, where detailed information of the requests is used for predicting future attacks using pattern recognition. A multilevel security defense is presented in [22], which considers a hierarchical WSN. The authors arrange regular sensors, gateways that are in charge to control regular sensors, base stations that control the gateways, and honeypots that collaborate with base stations. In each level a different kind of attack can be controlled.

(v) Artificial intelligent based mechanisms: exploiting knowledge about the nature of biological systems can result in valuable information about the attacker. For example, bioinspired solutions are applied to efficient computing (bioinspired computing), making robots that are inspired by the biological systems (bioinspired robotics), technical developments in engineering (bioinspired systems), and networking (bioinspired networking). So, they can also be applied to the design of an IDS for WSN. Honeypot can be considered an artificial intelligent technique due to the fact that it mimics the biological nature of particular species. Artificial intelligence is becoming an effective method to be applied in security detection systems [23].

This work centers our attention on artificial bioinspired security mechanisms for IDS in WSNs. We have designed an algorithm and a protocol to detect an intrusion attack inspired in the web spider behavior when an attack suffered in its web [24]. We technically implement our algorithm and protocol considering the honeypots technique. In contrast to [17], we are not concerned with routing inside the WSN, but in addition to that work we propose a transport and policy algorithm and protocol. We do not inspect the traffic of the intruder (as [19] did) but we consider it to reduce the rate of attacks it can do. Our objective is to gain time to find out information about the intruder. To do this, we implement a low-interaction sensors honeypot that tries to detect the intruders and then delays the answer to them for earlier learning of their future behavior. In contrast to [22], we do not consider a hierarchical WSN. We consider all the nodes are regular and have the same role in the network (honeypot sensors and real sensors).

The paper is organized as follows. In Section 2, we analyze the works found about bioinspired mechanisms used in security. Section 3 describes our web spider-inspired proposal. The system algorithm and protocol are explained in Section 4. Test bench experiments and results are included in Section 5. Finally, Section 6 draws the conclusion and future work.

2. Related Work

The section shows some works related to bioinspired mechanisms for security in WSN.

A survey on practical applications and open research issues for bioinspired self-organized networking (SON) systems is presented in [25]. The benefits of using these bioinspired techniques against conventional SON solutions include, but are not limited to, lower MAC delays, communications overhead and hardware complexity, higher adaptivity to changes, and resource utilization. Considering the benefits of these techniques, SON systems, such as WSN and wireless ad hoc networks, can exploit the improvements introduced by the bioinspired techniques compared to the isolated conventional SON solutions.

The authors in [26] apply the biological knowledge about the human immune system to propose a new network security mechanism to disable the fraudulent nodes in a WSN. Bioinspired algorithms provide dynamic, adaptive, and real-time methods of intrusion detection. The work included in [27] presents a review on genetic algorithm, artificial immune, and artificial neural network (ANN) based intrusion detection systems (IDS) techniques used in WSN.

Moreover, an algorithm inspired on the human immune system behavior to detect intruders in WSN is presented in [28].

A key component of bioinspired response methods is the use of feedback from the network to better adapt their response to the specific attack [29]. The author developed a method to calculate response times for a WSN that could be used to improve the bioinspired method for selecting the most suitable intrusion response for ad hoc networks.

In [30], a honeypot based framework is proposed that is used to earlier learn future attacks of the intruder and serve as a defensive countermeasure. It is based on the biological behavior of a particular species of ant. The ants store food forming a living repository of food and are often attacked by raiders. They considered a WSN as composed by two types of ants: honey ants and real ants. They strategically distribute the honeypot sensors (honey ants) that will mimic the physical data (real ants). Then, the IDS will induce traffic from alleged intruders to these honeypot sensors. This is done by implementing a swarm intelligence algorithm that takes into account the communication among sensors like the ants do. They route virtual values to confuse the intruder and also to make it believe that it is receiving real values. In this way the intruder could be discovered earlier.

Most bioinspired methods for WSN intrusion attacks are generally applied to a single protocol layer of the OSI stack, for example, (i) genetic algorithm at the physical layer; (ii) antiphase synchronization at the MAC layer, a bioinspired method based on the behavior of Japanese tree frogs; (iii) ant colony optimization at the network layer; (iv) and quantified trust models at the application layer. At present the combination of several bioinspired methods for WSN is applied to improve the system performance [31].

We propose a honeypot implementation for IDS in a WSN, which is bioinspired in the behavior of the web spider. We have only found one paper that uses a web spider-inspired mechanism [32]. It proposes a bioinspired algorithm based on the social behavior of spiders from Congo to detect and eliminate misbehaving sensor nodes in WSN. The biological inspiration comes from the fact that these kinds of spiders form a collaborative group to listen vibrations of victims in the web in order to hunt them. The bioinspired algorithm is distributed among sensor nodes (spiders) and it works as follows: one or more sensor nodes detect an attack from a suspected node (victim); then the sensor node sets a first level of alert and sends this detection to all their neighbors (collaboration); to reduce false alarms in the detection, the algorithm sets that if a second attack from the same suspected node is detected for the same sensor that detected the first attack or for a neighbor sensor, then the suspected node is considered as an intruder node. The paper does not present how and why this node is considered suspicious and how to reduce this intruder.

As far as we know there is not any other work published that uses the web spider behavior for WSN security. Moreover, the work presented in this paper is completely different from [32]. We have used different parts of the web spider behavior than the ones presented in [32].

3. Web Spider Defense Description

This section presents the description of the web spider defense technique and how it is applied to our system.

Spiders are often underestimated as suitable behavioral models. Spiders show surprising cognitive abilities, changing their behavior to suit their situational needs [33]. All spiders are predators. There are many types of spiders and there is a wide variety of methods used by them to capture their prey. Some spiders are hunters that chase and overpower their prey. Other spiders instead weave silk snares, or webs, to capture their prey [34, 35]. Some spiders inject poison into their prey. The poison paralyzes victims making them lose mobility. After paralyzing victims, spiders usually wrap their victims with silk and soften the meat with gastric juice. Finally spider absorbs the result of this mixture. The behavior that we are going to use in our system is the behavior of web spiders that use poison to paralyze their prey once it is trapped in the web. There are several types of web spiders, which can be spiral orb web, tangle web or cobweb, funnel web, tubular web, and sheet web.

When a spider wants to capture a prey, it builds a web and waits till some flies or mosquitoes are trapped in it. When it happens, the spider has a delicacy to attack bigger preys. It has just to wait some time till a new prey sees the fly and/or the mosquito and gets trapped when it tries to catch them. Now the procedure to paralyze this big prey is injecting poison, which slows down the mobility of the prey till it has no mobility.

This procedure is used by our system. We will use one or several fake wireless sensor nodes placed in the WSN, which announce network services and provide false data. These nodes have few or no security. It (or they) will be honeypots for the intruders. The idea of attracting attackers is not really new. It has been used in many other types of networks [36]. As soon as the fake wireless sensor node detects a connection, it will contact the network administrator, which will follow the connection and gather information from the intruder (such as getting the IP address and DNS name). Fake wireless sensor nodes, where security level is very low, will detect intruders by using any of the existing intrusion detection systems [11]. They will send data to sink nodes as regular nodes, but these fake data will be discarded by the sink node. In order to keep the intruder busy, the fake wireless sensor node slows down the replies to the intruder messages, like the poison of the spider when the prey is trapped in the web.

The system uses the connection establishments to keep intruders trapped. Every request is replied before the timeout, but it is delayed in order to let the system administrator gather information about the intruder. The system administrator is a node that is placed in the network, whose purpose is to gather information about a node through its IP address, DNS name, traces, and so forth.

4. System Algorithm and Protocol

This section presents the algorithm designed for our system and the protocol created for the proper operation of our system.

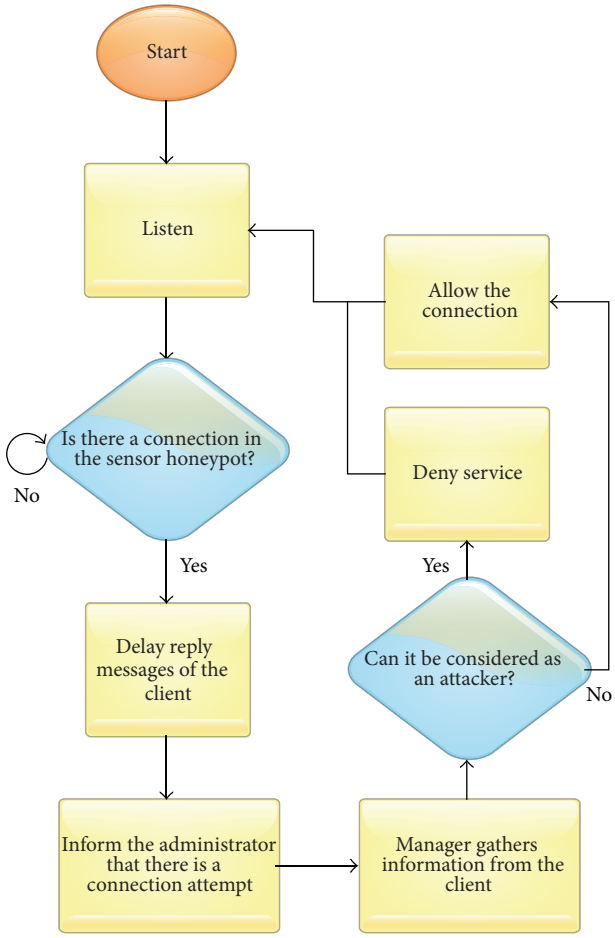


Figure 1: Spider defense algorithm.

Figure 1 shows the algorithm used for the intrusion or attackers detection and the steps followed to slow down their connections. At the beginning, the system listens if the fake wireless sensor node is receiving any connection request. If it receives a request, it slows down the connection and informs the network administrator that it has a possible intruder. This slow process is performed by a "wait procedure," which delays the replies. The delay time is lower than the threshold used by TPC connections for the exceeded time. These delays in the replies allow the network administrator to gather information about the intruder in order to identify it. The network administrator will be able to use any information gathering technique using echo request/reply, who is, and so forth. This information will be used to know if the user establishing the connection is an intruder or an attacker. If system confirms that the user is an intruder or an attacker, it will deny the service. If the user has the rights to perform this task because it belongs to the system, then the connection is established correctly and it goes to the listen state.

The designed protocol is shown in Figure 2. When the fake wireless sensor node receives a connection, it first sends a message to the network administrator in order to ask whether it is a trustable node or an intruder/attacker. Meanwhile it slows down the connection. The network administrator

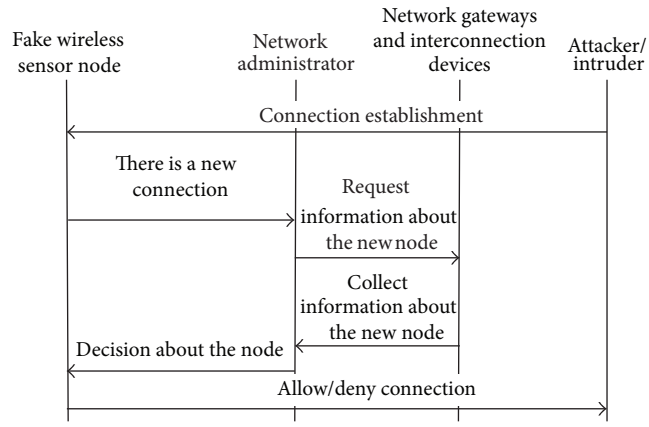


Figure 2: Network protocol.

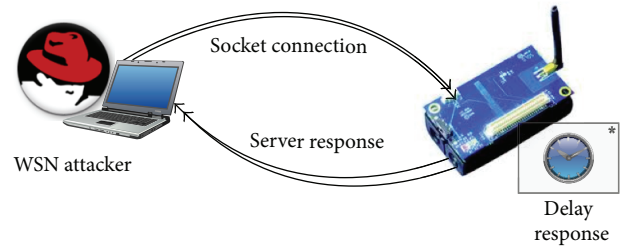


Figure 3: System architecture for the 1st experiment.

requests information about that node (by using its IP address, DNS name, traces, etc.) to the network gateways and interconnection devices. It gathers the information received about the type of node establishing the connection and informs the fake wireless sensor node. Then, it takes the appropriate action by denying or accepting the connection.

5. Test Bench Experiments

In order to carry out the performance study two experiments have been made. In both cases, the WSN attacker acts as the *client* and the wireless sensor node as *server*. Both communicate using TCP sockets and the communication is established following a three-way handshake algorithm. The last answer (segment [FIN, ACK]) is delayed to give time to the network manager to diagnose the connection as a secure or insecure one.

5.1. Experiment 1. Figure 3 shows the system architecture used for the first experiment. The WSN attacker uses a MacBook Pro with the following characteristics: Intel Core 2 Duo 2.4 GHz processor and 2 GB RAM. The sensor node has a 1.6 GHz processor and 1 GB RAM. The communication between the WSN attacker and sensor node is wireless. Wireshark sniffer program running in the WSN attacker node is used to compute the elapsed reply time. Both, client and server programs, have been coded in Java programming language.

Figure 4 shows different response times from the wireless sensor node to the WSN attacker according to the artificial

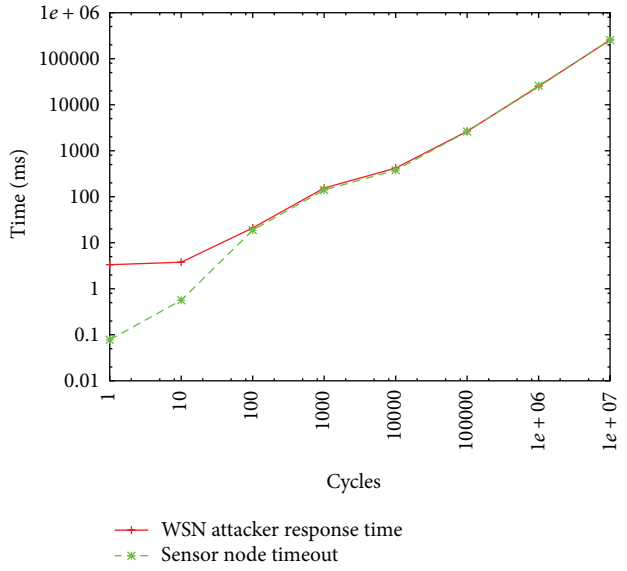


Figure 4: Response time as a function of the number of cycles.

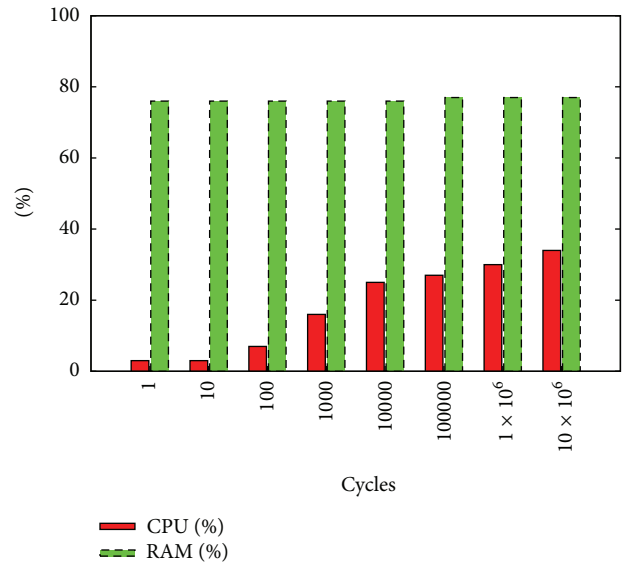


Figure 5: CPU and RAM usage as a function of the number of cycles.

delay introduced by the sensor node (a loop varying the response time generates these different response times). As you can see, if the delay is high, the total amount of time elapsed from the first segment [SEQ] to the reception of the last segment [FIN, ACK] is closer to the artificial delay. This is not true if the artficial delay is low.

Figure 5 shows the CPU and RAM consumption during the loop execution. The measurements were obtained with *top* Linux program. As it can be seen, RAM usage is not high enough to be considerable. On the contrary, the more the delay in the sensor node, the higher the CPU consumption in the sensor node. The delay should be close to the time needed by the network manager to diagnose if the attempt of connection initiated by the WSN attacker is secure or not. Moreover, we have to look for the minimum delay value that will affect the system performance, which is why we performed the second experiment.

5.2. Experiment 2. This second experiment helps us to determine the delay by measuring the reply time of the *tracert* and *ping* to the wireless sensor nodes in a network with different delays.

Figure 6 shows the system architecture used for the second experiment. The WSN attacker and each one of the 12 sensor nodes use the same equipment described for the previous experiment (the attacker uses a MacBook Pro with Intel Core 2 Duo 2.4 GHz processor and 2 GB RAM and the sensor node with 1.6 GHz processor and 1 GB RAM). Again, the communication between the WSN attacker and sensor node is wireless and the Wireshark sniffer program is running in the WSN attacker to compute the elapsed reply time.

Each sensor node is accessible from the WSN attacker via Internet. *NetDisturb* program [37] let us vary several network parameters such as the delay and jitter. Next we present the obtained simulation results.

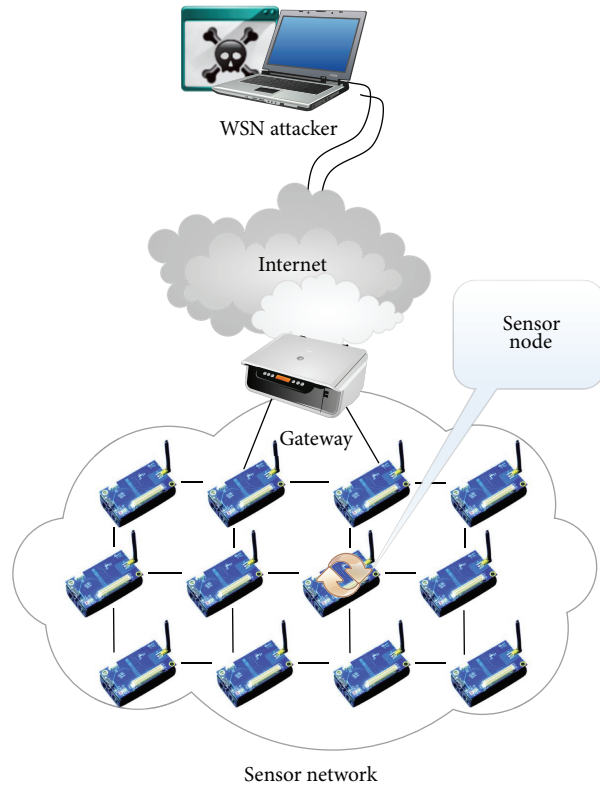


Figure 6: System architecture for the 2nd experiment.

Figure 7 shows both, average and maximum values for *ping* and *tracert* time responses for different constant network delays.

As Figure 7 shows, the more the network delay, the higher the response time for *ping* and *tracert*. An important issue derived from our experimentation is that the probability of

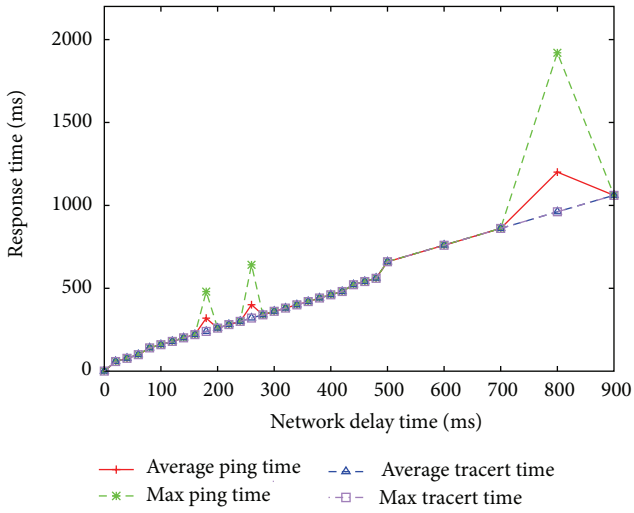


FIGURE 7: Response time for the *pings* and *tracerts* in the WSN with constant delay.

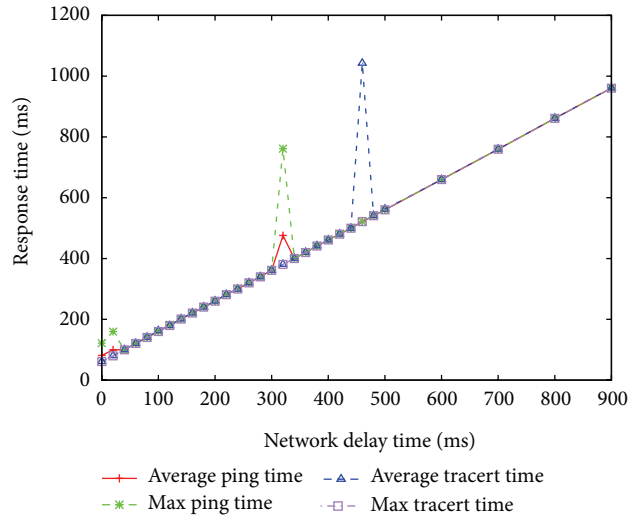


FIGURE 8: Response time for the *pings* and *tracerts* in the WSN with exponential jitter.

having a peak is higher for high network delays. Figure 7 demonstrates that this probability increases from a network delay higher than 200 ms. From this figure, we can make an estimation of the amount of time needed by the network manager to give a diagnosis about the connection between the WSN attacker and wireless sensor node. For example, if the network delay is 100 ms, the network manager takes into account the fact that the response time is 160 ms on average. As a result, the time to answer to the WSN attacker connection request should be greater than 160 ms.

Figure 8 shows results obtained varying exponentially the jitter according to the following equation:

$$f(x) = \lambda e^{-\lambda x} dx \quad \text{if } x \geq 0,$$
$$f(x) = 0 \quad \text{if } x < 0, \tag{1}$$

where $\lambda = 10$ and x is the delay variation.

As Figure 8 shows, there is higher probability to obtain a peak using *tracerts*. Another observation is that *ping* and *tracert* behavior is lineal in this experiment in comparison with Figure 7. The lineal behavior assists the network manager to predict the response time.

6. Conclusion

In this paper, we have presented a bioinspired system that uses the web spider hunting technique. We have explained how all steps performed by the web spider are included in our system. Moreover, we have detailed the system algorithm and the protocol procedure for the proper operation of the system. A real test bench has been implemented in order to validate our system.

In order to carry out our performance study, we have made two experiments. First, we tested performance of the direct communication between the WSN attacker and the wireless sensor node. Then, we performed a second experiment to measure the reply time of the wireless sensor nodes in a network with different delays.

In future works we will make performance experiments using one and several wireless attackers in order to know response times for the *ping* and the *tracert*. Moreover, our system will include other spider behaviors from other types of spiders. Now we are developing the system for a real environment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work has been partially supported by the "Ministerio de Ciencia e Innovación", through the "Plan Nacional de I+D+i 2008–2011" in the "Subprograma de Proyectos de Investigación Fundamental", Project TEC2011-27516.

References

[1] D. Bri, M. Garcia, J. Lloret, and P. Dini, "Real deployments of wireless sensor networks," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (Sensorcomm '09)*, pp. 415–423, Greece, Athens, Ga, USA, June 2009.

[2] M. Garcia, D. Bri, S. Sendra, and J. Lloret, "Practical deployments of wireless sensor networks: a survey," *Journal On Advances in Networks and Services*, vol. 3, no. 1-2, pp. 170–185, 2010.

[3] S. Sendra, J. Lloret, M. García, and J. F. Toledo, "Power saving and energy optimization techniques for wireless sensor networks," *Journal of Communications*, vol. 6, no. 6, pp. 439–459, 2011.

[4] M. Segal, "Improving lifetime of wireless sensor networks," *Network Protocols and Algorithms*, vol. 1, no. 2, pp. 48–60, 2009.

[5] S. Kuncha and P. V. G. D. P. Reddy, "Impact of security attacks on a new security protocol for mobile ad hoc networks," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 122–1403, 2011.

[6] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302–1325, 2011.

[7] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: attack analysis and countermeasures," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 867–880, 2012.

[8] W. T. Zhu, J. Zhou, R. H. Deng, and F. Bao, "Detecting node replication attacks in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 1022–1034, 2012.

[9] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 4, no. 1-2, 2009.

[10] Y. Maleh and A. Ezzati, "A review of security attacks and intrusion detection schemes in wireless sensor network," *International Journal of Wireless & Mobile Networks*, vol. 5, no. 6, pp. 1–12, 2013.

[11] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: a review," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 167575, 7 pages, 2013.

[12] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 56–63, 2007.

[13] H. C. Chaudhari and L. U. Kadam, "Wireless sensor networks: security, attacks and challenges," *International Journal of Networking*, vol. 1, no. 1, pp. 4–16, 2011.

[14] N. Fatema and R. Brad, "Attacks and counterattacks on wireless sensor networks," *International Journal of Ad hoc, Sensor & Ubiquitous Computing*, vol. 4, no. 6, pp. 1–15, 2013.

[15] A. Radhika, D. Kavitha, and D. Haritha, "Mobile agent based routing in MANETs—attacks & defences," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 108–121, 2011.

[16] R. H. Jacobsen, Q. Zhang, and T. S. Toftegaard, "Bioinspired principles for large-scale networked sensor systems: an overview," *Sensors*, vol. 11, no. 4, pp. 4137–4151, 2011.

[17] J. Xu, J. Wang, S. Xie, W. Chen, and J. Kim, "Study on intrusion detection policy for wireless sensor networks," *International Journal of Security and its Applications*, vol. 7, no. 1, pp. 1–6, 2013.

[18] M. S. Sisodia and V. Raghuwanshi, "Anomaly base network intrusion detection by using random decision tree and random projection: a fast network intrusion detection technique," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 93–107, 2011.

[19] H. Zhijie and W. Ruchuang, "Intrusion detection for wireless sensor network based on traffic prediction model," *Physics Procedia*, vol. 25, pp. 2072–2080, 2012.

[20] N. Al-Gharabally, N. El-Sayed, S. Al-Mulla, and I. Ahmad, "Wireless honeypots: survey and assessment," in *Proceedings of the Conference on Information Science, Technology and Applications (ISTA '09)*, pp. 45–52, ACM, March 2009.

[21] V. Gopinath, *Success analysis of deception in wireless sensor networks [M.S. thesis]*, Oklahoma State University, 2010.

[22] S. K. Srivastava, B. K. Mishra, and B. K. Mishra, "Security framework against malicious attacks in wireless sensor network," *International Journal of Advanced Technology & Engineering Research*, vol. 3, no. 5, pp. 7–11, 2013.

[23] N. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Artificial neural network based detection of energy exhaustion attacks in wireless sensor networks capable of energy harvesting," *Journal of Ad Hoc & Sensor Wireless Networks*, vol. 2013, pp. 1–25, 2013.

[24] "Mecanismos de defensa de las arañas," http://www.aracnipedia.com/mecanismos-defensa-aranas/.

[25] Z. Zhang, K. Long, J. Wang, and F. Dressler, "On swarm intelligence inspired self-organized networking: its bionic mechanisms, designing principles and optimization approaches," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 513–537, 2014.

[26] H. Rathore and S. Jha, "Bio-inspired machine learning based wireless sensor network security," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC '13)*, pp. 140–146, Fargo, ND, USA, August 2013.

[27] N. A. Alrajeh and J. Lloret, "Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 351047, 6 pages, 2013.

[28] J. Baburajan, "Intrusion detection in wireless sensor networks using watchdog based clonal selection algorithm," *International Journal of Research in Engineering & Advanced Technology*, vol. 1, no. 1, 2013.

[29] M. K. Amirkolaei, *Enhancing bio-inspired intrusion response in Ad-hoc networks [Ph.D. thesis]*, Edinburgh Napier University, Edinburgh, UK, August 2013, http://researchrepository.napier.ac.uk/6533/.

[30] R. Muraleedharan and L. A. Osadciw, "An intrusion detection framework for sensor networks using Honeypot and Swarm Intelligence," in *Proceedings of the 6th Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '09)*, Toronto, Canada, July 2009.

[31] W. S. Hortos, "Bio-inspired, cross-layer protocol design for intrusion detection and identification in wireless sensor networks," in *Proceeding of the 37th IEEE Conference on Local Computer Networks Workshops, LCN Workshops*, pp. 1030–1037, Clearwater, Fla, USA, October 2012.

[32] K. Benahmed, M. Merabti, and H. Haffaf, "Inspired social spider behavior for secure wireless sensor networks," *International Journal of Mobile Computing and Multimedia Communications*, vol. 4, no. 4, pp. 1–10, 2012.

[33] M. E. Herberstein, *Spider Behaviour: Flexibility and Versatility*, Cambridge University Press, Cambridge, UK, 2011.

[34] Spiderword Website, *Spider Methods of Capturing Prey*, 2014, http://www.spidersworlds.com/spider-methods-of-capturing-prey/.

[35] R. F. Foelix, *Biology of Spiders*, Oxford University Press, 3rd edition, 2010.

[36] M. Ficco, "Achieving security by intrusion-tolerance based on event correlation," *Network Protocols and Algorithms*, vol. 2, no. 3, pp. 70–84, 2010.

[37] NetDisturb website, http://www.zti-communications.com/netdisturb.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 534760, 6 pages
http://dx.doi.org/10.1155/2014/534760

Research Article

A Framework for Obesity Control Using a Wireless Body Sensor Network

Nabil Ali Alrajeh,[1] Jaime Lloret,[2] and Alejandro Canovas[2]

[1] *Biomedical Technology Department, College of Applied Medical Sciences, King Saud University, Riyadh 11633, Saudi Arabia*
[2] *Integrated Management Coastal Research Institute, Universidad Politecnica de Valencia, C/Paranimf No. 1, Grao de Gandia, Gandia, 46730 Valencia, Spain*

Correspondence should be addressed to Nabil Ali Alrajeh; nabil@ksu.edu.sa

Received 23 May 2014; Accepted 1 July 2014; Published 15 July 2014

Academic Editor: S. Khan

Low-cost low-power consumption small wireless sensor devices have empowered the development of wireless body area networks (WBANs). In WBANs many sensors are attached to human body for sensing particular health related information to improve healthcare and quality of life. Obesity is one of the most common problems all over the world, which is amongst main causes of cardiovascular diseases. In this research, we explore hardware and software architecture of WBAN for obesity monitoring. The proposed framework consists of few sensor nodes that monitor body motion, calories calculator, and a personal server running on a personal smart phone or a personal computer. The focus of this research is to make obesity patients easier to get rid of this disease.

1. Introduction

A wireless sensor network (WSN) has many applications in which wireless body area network (WBAN) has gained significant importance. In WBAN [1, 2], small electronic devices are attached with human to monitor specific health related problem such as blood pressure, blood sugar level, and organ movement. The concept of WBAN is presented to facilitate the healthcare issues distantly or to monitor athletes [3]. WBAN consists of small intelligent electronic devices termed as sensors, which are low in power and processing [4]. These small sensors collect heath related data and communicate that data to some medical officers or medical servers so that it can be analyzed and monitor the patient health parameters or to track their fatigue and muscle stress. This kind of mechanism created ease in patient life to enjoy mobility and need not to stay at hospital all the time [5]. In WBAN, three layers play an important role for sensing precise readings of patient health and transmitting accurate information to medical servers, that is, physical layer, MAC layer, and network layer.

At physical layer, WBAN normally faces significant signal loss using narrow band or ultrawide band [6]. Furthermore, less path loss is observed in line of sight (LOS) communication as compared to nonline of sight due to capability of body fluid to absorb waves [7, 8]. Furthermore, mobility can also greatly affect the signal loss; by increasing movement and degree of movement, more signal loss occurs [9–11]. There are many research and design issues in antenna and radios that must be resolved to enable efficient and flawless deployment of WBAN. WBAN is playing role in improving e-healthcare and quality of life.

At MAC layer, it is useful to implement wireless sensor protocols in WBAN [6, 12]. Several protocols have been proposed to guarantee emergency handling in WBAN, such as the one presented in [13].

Due to resource constraints, routing protocols designed for mobile ad hoc networks or wireless sensor networks cannot be used in WBAN. Research community is trying to design specialized routing protocols for WBAN such as TARA [14], ALTR [15]. These protocols are concerned with body heat and finding alternate path. Few LEACH based clustered routing protocols are also presented [16]. Moreover, other alternatives for m-health and low-power wearable sensor networks, such as delay tolerant routing protocols [17] and cooperation mechanisms [18], have been proposed. In

the recent years, there have appeared security proposals that take into account energy harvesting and could be implemented in e-health and m-health systems [19, 20].

Research fields such as mobile sensing [21] and data gathering algorithms for mobile sensors [22] are experiencing huge advances which benefit tremendously m-health and e-health systems.

Overweight or obesity is one of the most common problems all over the world. Most of cardiovascular diseases (CVD) are associated with it, which is the main cause of death in the world. According to the World Health Organization (WHO), worldwide about 17.5 million people die of heart attacks or strokes each year; in 2015, almost 20 million people will die from CVD. These deaths can often be prevented with proper healthcare [23].

In this paper, we present a framework for controlling obesity to facilitate all such patients. This research has two main contributions.

(i) One is designing of a communication system which consists of WBAN, wireless personal area network (WPAN), and wireless metropolitan area network (WMAN). WBAN deals with communication amongst sensor nodes deployed over human body. WPAN enables our proposed system to communicate WBAN with personal computer or smart phone, while WMAN is capable of transferring data from personal server to medical server for medical advice in case of emergency.

(ii) The other is devising personalized body mass index (BMI) and calories calculator. This calculator calculates personalized BMI depending on current weight, height, gender, and so forth. Similarly calories calculator keeps track of daily calories intake and calories burnt. This module also gives intelligent suggestions at the end of the day.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 briefly introduces background of the problem. Section 4 explains proposed architecture. Section 5 discusses results. Finally Section 6 concludes the paper.

2. Related Work

WBAN is currently being used in a variety of healthcare applications. In [24], WBAN architecture is proposed for physiological signal monitoring and health consulting in ubiquitous environment. In this mechanism ZigBee is used to communicate mobile system and physiological devices. Furthermore, this paper proposes few algorithms such as scanning, dynamic discovery, and healthcare profile. Hip rehabilitation system using WBAN is proposed in [25]. In this paper, the author pointed out few challenges such as energy efficiency, reliability, network operation, and low latency. Furthermore, hip rehabilitation system is proposed for such patients who suffer from hip surgery. To measure the force between hip and shoe, capacitive insole sensors are used, while magnetic sensors are used to measure hip and leg position of patients. A complete survey is conducted in [6]. The focus of this survey is to highlight few patient monitoring systems and discussion about different WBAN technologies. Furthermore, the author discusses recent and current research work done in physical, MAC, and network layers of WBAN. Some security issues and mechanisms are also highlighted in this survey paper.

Ambulatory health status monitoring system with the help of software and hardware architecture is presented in [26]. This system monitors body motion and heart activities using multiple sensor nodes and personal server.

In [27], WBAN challenges and opportunities are discussed in detail regarding application areas, communication, storage, energy harvesting, and compatibility issues. Another body area network survey is presented in [28]. In this survey, the focus of discussion is on WBAN intra-BAN and inter-BAN communication modeling, different hardware and devices, physical, MAC and network layer issues, and energy conservation strategies. Furthermore, taxonomy of body sensor projects is highlighted.

In [29], Lopes et al. present SapoFitness, a mobile health application for dietetic monitoring and assessment, which is focused on keeping a daily personal health record of a user for obesity prevention. The application is able to send alerts and messages concerning the user's diet program taking into account his/her physical activity. Its main goal is to help the user to lose weight and have a good and balanced nutritional state.

3. Problem Background

In this section, we will discuss obesity and other related information to the proposed architecture.

3.1. Obesity. Obesity is a medical condition in which excess body fat has accumulated to the extent that it may have an adverse effect on health, leading to reduced life expectancy and increased health problems [30]. Obesity is a state of body which is overweight with high degree of fat. There are a number of risks involved in obesity such as diabetes, heart diseases, and depression. Obesity can be controlled gradually by taking prevention mechanisms such as exercise and dieting.

3.2. Body Mass Index. Body mass index (BMI) is a simple index of weight-to-height which is used to identify overweight in people. The World Health Organization (WHO) definition of overweight and obesity is as follows.

(i) BMI greater than or equal to 25 is overweight.

(ii) BMI greater than or equal to 30 is obesity.

BMI can be calculated using the following expression:

$$\text{BMI} = \frac{\text{weight } (\text{Kg})}{\text{height } (\text{m}^2)}. \quad (1)$$

BMI classification is presented in Table 1.

Table 1: BMI classification.

BMI	Classification
<18.5	Underweight
18.5–24.9	Normal weight
25.0–29.9	Overweight
30.0–34.9	Class I obesity
35.0–39.9	Class II obesity
≥40.0	Class III obesity

3.3. Calories Requirements. A calorie is a unit of energy which is used to power our body. All foods have certain amount of calories. Major sources of calories are carbohydrates, fats, and proteins. Fats have the highest number of calories, where one gram of pure fat contains nine calories, whereas pure protein and pure carbohydrate contain four calories each. Calories play an important role in our diet and understanding calories can help in weight management. One important fact is that calories requirements depend on many factors such as age, weight, height, and gender.

4. Proposed Architecture

In this section, we discuss our proposed architecture for obesity control using WBAN. The proposed architecture consists of software and hardware architecture as shown in Figure 1.

The system procedure follows the algorithm shown in Figure 2. When data are received from the body sensors, the algorithm first checks if values are higher (or lower, depending on the case) than a threshold. Then, the system is able to estimate the BMI, the ideal weight, and the calories, based on the input parameters. Taking into account the values, the system takes information from its database and provides some intelligent suggestions.

In the following subsections we present the hardware and communication model and the software model.

4.1. Hardware and Communication Model. Generally, WBAN devices are equipped with low-power small-size sensors (less than 1 cm^3). The hardware used in the proposed architecture is four small-size sensors which are attached to hands and feet and a computing device which can be smart phone or small personal computer acting as a server. They are connected using a star network topology. We used iMote2 sensors, which have TinyOS and use IEEE 802.15.4 standard for communication with other sensor nodes [31]. Communication distance can be up to 20 meters, with a data rate of 250 kbps. IEEE 801.15.4 standard operates at 2.4 GHz frequency band. The primary objective of these sensor nodes is to monitor the body motion. Body motion readings are transmitted to the server for further processing.

In such cases, where a lot of body sensor nodes are used to cover several parts of the patient body, the end to end delay and congestion would be the main concerns due to the few available nonoverlapping channels. If the medical condition of the patient needs to transmit images or multimedia data to the medical servers, high data rates are needed. Sensor nodes use CSMA/CA at MAC layer. MAC layer performs some vital tasks such as data rate management, load balancing, error control, power control, and collision avoidance. It should be noted that there are many contention free MAC protocols such as TDMA and FDMA which can perform better under centralized mechanism. However, we used CSMA/CA with RTC/CTS mechanism in order to reduce the interference among nodes.

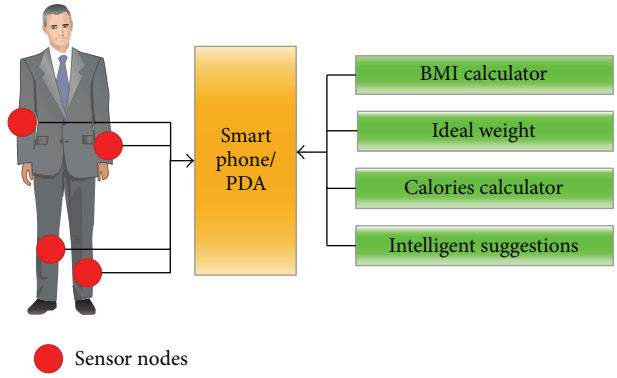


Figure 1: Proposed framework.

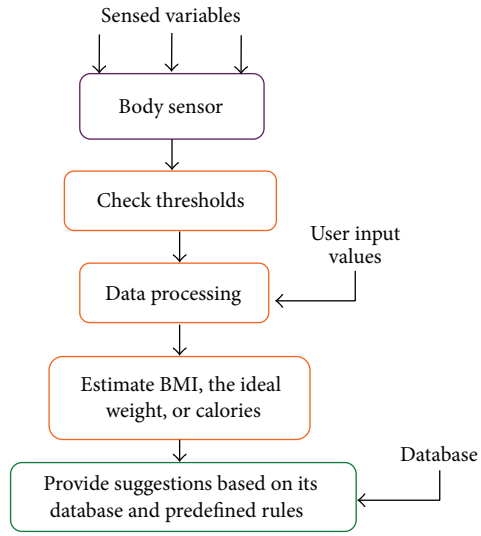


Figure 2: System procedure.

4.2. Software Model. The proposed system is equipped with a software application which includes

(i) BMI calculator,

(ii) personalized calories calculator,

(iii) calories adder module,

(iv) calories consumption module,

(v) suggestion module.

BMI calculator is used to calculate personalized mass index for users as given in Figure 3.

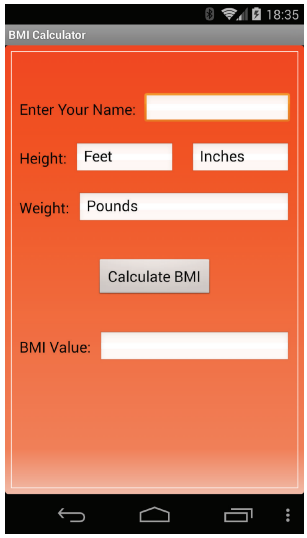


Figure 3: BMI calculator.

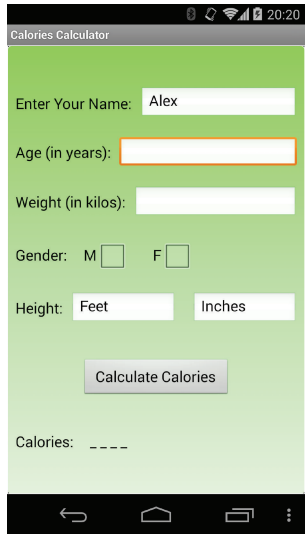


Figure 5: Calories calculator.

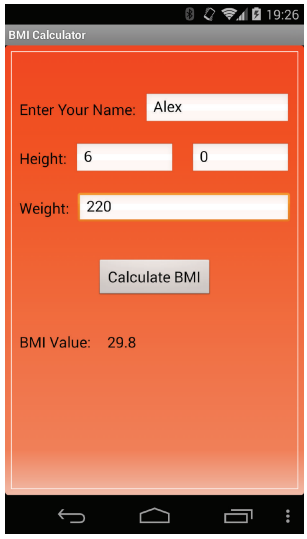


Figure 4: BMI calculation.

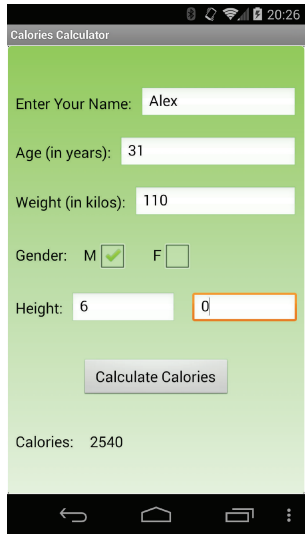


Figure 6: Calories calculation of a user.

The experimental values are presented in Figure 4.

Calories calculator computes daily the calorie requirements of a user based on the age, gender, weight, and height as given in Figure 5.

This calculator estimates the number of calories to maintain current weight. It can estimate the personalized calorie requirements for each user as given in Figure 6.

The number 2540 calories means that the user Alex needs this amount of calories to maintain his/her current weight. If he/she consumes more than this value, the weight will be increased, but if he/she reduces the consumption of calories, the weight can be reduced.

We included a module called calories in the proposed mechanism, in which the user is capable of adding intake calories. The calories adder keeps record of the total calories taken during a day. Calories adder uses a database with the calories of the food. If some food is not mentioned in calories database, it can be entered manually. At the end of the day, the user will be able to know the total amount of calories taken.

Calories consumption module operates in association with the sensors attached to the human body. The values are estimated based on the sensed parameters.

Another important module of proposed system is the suggestion module. This module is activated at the end of the day. It provides useful suggestions after computing daily intake calories and estimating the total consumed calories as given in Figure 7.

5. Results

Efficient data delivery is one of the most desirable aspects of WBAN. Figure 8 presents the data delivery ratio from sensor nodes to a server when the user is in mobility, that is, when the user is running. There is significant difference between

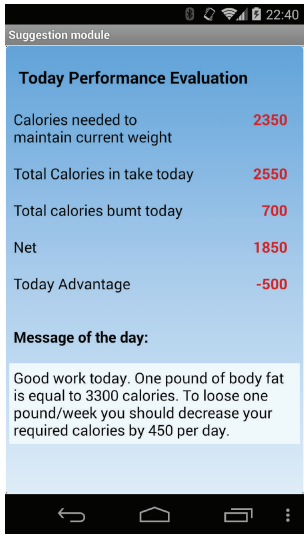


Figure 7: Suggestion module.

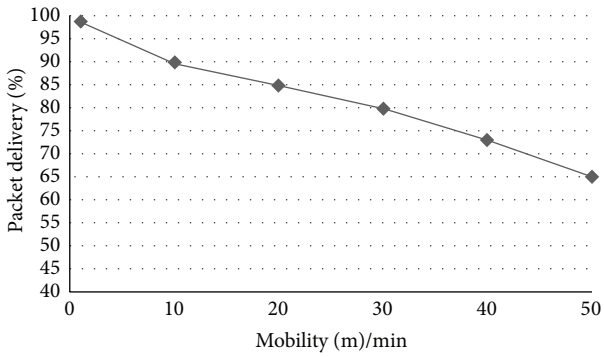


Figure 8: Data delivery rates.

data delivery when the user is in motion than when it is not. The reason is that when human body is in fast motion, there is significant drop in data delivery in ad hoc networks, especially in WBAN.

Figure 9 represents end to end delay when human body is in motion. As soon as the movement increases, end to end delay increases as well.

6. Conclusion and Future Work

One of the major causes of heart attacks is obesity. In this paper, we presented a mechanism of weight management and weight control using WBAN. The proposed mechanism consists of both software architecture and hardware architecture. The hardware architecture deals with sensor nodes and personal server, while software architecture consists of calories calculator, BMI calculator, adder, calories consumption module, and suggestion module. Our results show that when the user is in fast motion, some performance degradation occurs in terms of data delivery and more end to end delay is observed. In future works we are planning to include in the system a device with efficient energy conservation and multiradio multichannel mechanism in order to increase data delivery and reduce end to end delay.

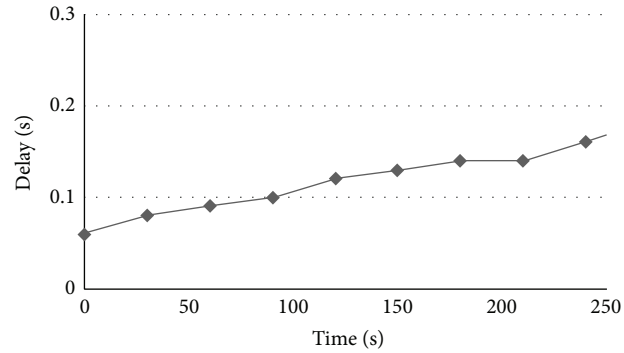


Figure 9: End to end delay in case of body motion.

Conflict of Interests

The authors have no conflict of interests.

Acknowledgment

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for funding this Research Group no. RG-1435-037.

References

[1] K. Van Dam, S. Pitchers, and M. Barnard, "Body area networks: towards a wearable future," in *Proceedings of WWRF Kick Off Meeting*, Munich, Germany, March 2001.

[2] R. Schmidt, T. Norgall, J. Mörsdorf, J. Bernhard, and T. von der Grün, "Body Area Network BAN—a key infrastructure element for patient-centered medical applications," *Biomedizinische Technik*, vol. 47, pp. 365–368, 2002.

[3] M. García, A. Catalá, J. Lloret, and J. J. P. C. Rodrigues, "A wireless sensor network for soccer team monitoring," in *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '11)*, Barcelona, Spain, June 2011.

[4] G. Sun, G. Qiao, and B. Xu, "Link characteristics measuring in 2.4 GHz body area sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 519792, 13 pages, 2012.

[5] J. Tomas, J. Lloret, D. Bri, and S. Sendra, "Sensors and their application for disabled and elderly people," in *Handbook of Research on Personal Autonomy Technologies and Disability Informatics*, pp. 311–330, IGI Global, 2011.

[6] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Journal of Wireless Networks*, vol. 17, no. 1, pp. 1–18, 2011.

[7] T. Zasowski, G. Meyer, F. Althaus, and A. Wittneben, "UWB signal propagation at the human head," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 4, pp. 1836–1844, 2006.

[8] D. Bri, J. Lloret, C. Turro, and M. Garcia, "Measuring specific absorption rate by using standard communications equipment,"

in *Telemedicine and E-Health Services, Policies and Applications: Advancements and Developments*, pp. 81–111, IGI Global, 2012.

[9] A. Fort, C. Desset, J. Ryckaert, P. De Doncker, L. Van Biesen, and P. Wambacq, "Characterization of the ultra wideband body area propagation channel," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICU '05)*, pp. 22–27, Zurich, Switzerland, September 2005.

[10] M. Di Renzo, R. M. Buehrer, and J. Torres, "Pulse shape distortion and ranging accuracy in UWB-based body area networks for full-body motion capture and gait analysis," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 3775–3780, November 2007.

[11] D. Neirynck, *Channel characterisation and physical layer analysis for body and personal area network development [Ph.D. thesis]*, University of Bristol, Bristol, UK, 2006.

[12] S. Sendra, J. Lloret, M. García, and J. F. Toledo, "Power saving and energy optimization techniques for wireless sensor networks," *Journal of Communications*, vol. 6, no. 6, pp. 439–459, 2011.

[13] J. S. Ranjit and S. Shin, "A modified IEEE 802.15.4 superframe structure for guaranteed emergency handling in wireless body area network," *Network Protocol and Algorithms*, vol. 5, no. 2, pp. 1–15, 2013.

[14] Q. Tang, N. Tummala, S. K. S. Gupta, and L. Schwiebert, "Communication scheduling to minimize thermal effects of implanted biosensor networks in homogeneous tissue," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 7, pp. 1285–1294, 2005.

[15] A. Bag and M. A. Bassiouni, "Energy efficient thermal aware routing algorithms for embedded biomedical sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems (MASS '06)*, pp. 604–609, Vancouver, Canada, October 2006.

[16] T. Watteyne, S. Augé-Blum, M. Dohler, and D. Barthel, "Anybody: a self-organization protocol for body area networks," in *Proceedings of the 2nd International Conference on Body Area Networks (BodyNets)*, Florence, Italy, June 2007.

[17] M. Quwaider and S. Biswas, "Delay tolerant routing protocol modeling for low power wearable wireless sensor networks," *Network Protocol and Algorithms*, vol. 4, no. 3, pp. 15–34, 2012.

[18] T. M. F. MacHado, I. M. Lopes, B. M. Silva, J. J. P. C. Rodrigues, and J. Lloret, "Performance evaluation of cooperation mechanisms for m-health applications," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 1664–1669, Anaheim, Calif, USA, December 2012.

[19] N. A. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Secure routing protocol using cross-layer design and energy harvesting in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 374796, 11 pages, 2013.

[20] N. A. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Artificial neural network based detection of energy exhaustion attacks in wireless sensor networks capable of energy harvesting," *Ad Hoc & Sensor Wireless Networks*, vol. 22, no. 3-4, 2014.

[21] E. Macias, A. Suarez, and J. Lloret, "Mobile Sensing Systems," *Sensors*, vol. 13, no. 12, pp. 17292–17321, 2013.

[22] N. Meghanathan and P. Mumford, "Centralized and distributed algorithms for stability-based data gathering in mobile sensor networks," *Network Protocol and Algorithms*, vol. 5, no. 4, pp. 84–116, 2013.

[23] World Health Organization Report, "Cardiovascular diseases (CVDs)," http://www.who.int/mediacentre/factsheets/fs317/en/index.html.

[24] J. Jung, K. Ha, and J. Lee, "Wireless body area network in a ubiquitous healthcare system for physiological signal monitoring and health consulting," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 1, no. 1, pp. 47–54, 2008.

[25] M. Soini, J. Nummela, P. Oksa, L. Ukkonen, and L. Sydänheimo, "Wireless body area network for hip rehabilitation system," *Ubiquitous Computing and Communication Journal*, vol. 3, no. 5, pp. 42–48, 2008.

[26] C. Otto, A. Milenkovic, C. Sanders, E. Jovanov, and A. Milenković, "system architecture of a wireless body area sensor network for ubiquitous health monitoring," *Journal of Mobile Multimedia*, vol. 1, no. 4, pp. 307–326, 2006.

[27] M. A. Hanson, H. C. Powell Jr., A. T. Barth et al., "Body area sensor networks: challenges and opportunities," *IEEE Computer Magazine*, vol. 42, no. 1, pp. 58–65, 2009.

[28] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. M. Leung, "Body area networks: a survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, 2011.

[29] I. M. Lopes, B. M. Silva, J. J. P. C. Rodrigues, J. Lloret, and M. L. Proença, "A mobile health monitoring solution for weight control," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP '11)*, Nanjing, China, November 2011.

[30] World Health Organization, *Obesity and Overweight, Fact sheet No. 311*, March 2013, http://www.who.int/mediacentre/factsheets/fs311/en/index.html.

[31] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, "IMOTE2: serious computation at the edge," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '08)*, pp. 1118–1123, Crete, Greece, August 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 768936, 17 pages
http://dx.doi.org/10.1155/2014/768936

Research Article

Energy Efficient Routing in Wireless Sensor Networks Based on Fuzzy Ant Colony Optimization

Ehsan Amiri,[1] Hassan Keshavarz,[2] Mojtaba Alizadeh,[2] Mazdak Zamani,[3] and Touraj Khodadadi[2]

[1] *Department of Computer Engineering, Nourabad Mamasani Branch, Islamic Azad University, Nourabad, Mamasani, Iran*
[2] *Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia*
[3] *Advanced Informatics School, Universiti Teknologi Malaysia, Federal Territory, 54100 Kuala Lumpur, Malaysia*

Correspondence should be addressed to Mojtaba Alizadeh; amojtaba2@live.utm.my

Received 4 March 2014; Accepted 15 May 2014; Published 3 July 2014

Academic Editor: S. Khan

A wireless sensor network (WSN) is a collection of sensor nodes that dynamically self-organize themselves into a wireless network without the utilization of any preexisting infrastructure. One of the major problems in WSNs is the energy consumption, whereby the network lifetime is dependent on this factor. In this paper, we propose an optimal routing protocol for WSN inspired by the foraging behavior of ants. The ants try to find existing paths between the source and base station. Furthermore, we have combined this behavior of ants with fuzzy logic in order for the ants to make the best decision. In other words, the fuzzy logic is applied to make the use of these paths optimal. Our algorithm uses the principles of the fuzzy ant colony optimization routing (FACOR) to develop a suitable problem solution. The performance of our routing algorithm is evaluated by Network Simulator 2 (NS2). The simulation results show that our algorithm optimizes the energy consumption amount, decreases the number of routing request packets, and increases the network lifetime in comparison with the original AODV.

1. Introduction

In WSNs all sensor nodes try to collect data and send sensing data to the base station (BS). The wireless communication uses a fixed node in the center of environment that is used to gather data from other sensor nodes called BS. Sometimes, source and destination are not in the neighborhood and they have multihop distance with each other where this distance leads to more energy consumption. The node energy consumption is the amount of energy each node consumes for sending/receiving data to/from other nodes. The summation of energy consumption by network nodes shows the network energy consumption. Since these energy sources are irreplaceable and have a straight effect towards the network lifetime, new protocols establishment or improvement in current protocols seems necessary for better energy saving in the network.

Routing is one of the most important protocols that consume energy in the network. Routing has been defined as a dynamic optimization task aiming at providing paths that are optimal in terms of some criteria such as minimum distance, maximum bandwidth, and shortest delay and satisfying some constraints such as limited power of nodes and limited capacity of wireless links. Routing is one of the major problems in the WSN and is one of the most interesting research fields in the communication network. The routing of the information is thus done locally and hop-by-hop. Each node independent of other decisions in other nodes and its past decision makes a new routing decision [1]. Routing usually directs based on the routing records that exist in each node routing table (RT) to various destinations in the network. The RT is a data table that is stored in node memory lists routs to special nodes. Nowadays, algorithms act differently to find the route between sender and destination; more algorithms only find a route and some routing algorithms find a multipath. The multiroute algorithms waste more network resources due to more message transmission as opposed to the single-path algorithms [2, 3].

Proactive, reactive, and hybrid [2, 4] are three main categories of routing algorithms. Proactive or, alternatively, the table-driven algorithms always show a fresh list of routes. Proactive algorithms have a slow reaction in the network with more failure and restructuring [2]. Examples of proactive algorithms are FSR [5] and DSDV [6]. The reactive or on-demand routing algorithm uses the flooding technique for finding only one path between the sender and destination. If the route exists before, the new route request will not accept them. High latency time in route finding is one of these algorithms' disadvantage and another is imposed overhead on the network due to the flooding technique; nevertheless it has less overhead compared to that for the proactive group. The most famous routing protocols of this group are DFS [7] and AODV [2, 4]. The third group is known as the hybrids. They are a combination of the reactive routing protocols and proactive routing protocols [4].

Ant colony [8, 9] is a standard solution for finding optimal path (from source to destination) that is proposed by Dorigo et al. in 1992 for solving optimization problems such as travelling salesman problem (TSP) with multiagents. By paying attention to inherent properties of routing, it will be suitable to be solved by the ant colony algorithm. Intelligent agent by the sensors will be able to sense events of the surrounding environment. ACO is based on the intellectual foundation that can easily be described in one sentence: ants select the best path among the existing barriers and constraints in nature to achieve food. This selection of a path is considered optimal among the different paths. Zadeh [10, 11] originally proposes the fuzzy set theory and it has been developed for expanded linguistic values. The linguistic values are terms, which are used instead of the numbers and fuzzy set theory [12]. The use of the fuzzy logic to optimize the metric used in routing approaches for WSNs is a promising technique since it allows us to combine and evaluate diverse parameters in an efficient manner. Moreover, several proposals have shown that the use of fuzzy logic in this kind of networks is a good choice due to the execution of the requirements that can be easily supported by sensor nodes, while it is able to improve the overall network performance [13]. Fuzzy logic consists of a decision system approach which works similarly to human control logic. It provides a simple method to reach a conclusion from imprecise, vague, or ambiguous input information [14].

In this paper, we aim to discover the optimal route in WSNs from the sources node to the BS. We are using intelligent ants that have some knowledge in the fuzzy logic and we call this FACOR algorithm. They will find the optimal path among all paths that may exist from the source to BS. In the first step, the source node sends ants to all neighbors; then each ant tries to find a route to BS. The ants will calculate the fuzzy amounts for their neighbors and the next hop based on these fuzzy amounts will be selected. This step will continue until the ants are able to find a route to BS. If an ant could not find this route in determined time, it kills itself. After that, the BS makes final decisions and determines the winner ant. The winner ant returns to its route and updates the routing table and some more information for all nodes on its route. We use the RREQ and RREP messages for modeling our ants. The simulation results show that this proposal reduces the number of packets needed to find the routes. By this manner we reduce the network bandwidth usage and decrease the amount of energy consumption because each node needs energy to send the packets. Also, our algorithm will be able to find optimal route and this route saves the network energy due to the shortest path selection. Furthermore, our algorithm also decreases the end-to-end delay time for sending and receiving packets. Giving attention to the above context, the network lifetime will increase.

Recently, many researches have been attempting to address this problem. Being attentive to the structural constraints that have WSN such as sensor limitation energy, developing efficient algorithms for routing seems necessary [4]. Sahani and Kumar in [15] proposed a novel routing approach using an ant colony optimization algorithm which uses artificial ants. Each ant chooses the next hop; moreover the pheromone concentration amount attends to the node's remaining energy. By this method, the ant will select a node with longer lifetime.

In [16] Suhonen et al. proposed an energy-efficient multihop routing protocol referred to as TUTWSNR (Tampere University of Technology WSN Routing) for wireless sensor networks. They use cost metrics to create gradients from the source to the destination node. The cost metrics consist of energy, node load, delay, and link reliability information that provide a trade-off between performance and energy usage. A node can query routes from its neighbors, which allows efficient recovery from route losses.

Okdem and Karaboga in [17] presented a novel routing approach using an ant colony optimization algorithm for wireless sensor networks consisting of stable nodes. The main goal of their study was to maintain network lifetime at its maximum. A multipath data transfer is also accomplished to provide reliable network operations, while considering the energy levels of the nodes. They also implement their approach on a hardware component to allow designers to easily handle routing operations in WSNs.

Kadri et al. in [18] are going to adapt the conventional ant routing algorithm for WSNs, by taking into consideration their traffic pattern and devices' constraints. The proposed protocol affects the task of route discovery to the BS which periodically launches forward ants over the network to discover routes and inform sensors about its location instead of letting each sensor do this task individually which consumes sensors' resources and decreases the network lifetime due to the broadcasting nature of the forward ants. They have also proposed execution of a handshake during the route discovery in order to secure links between each sensor and the BS, as the use of the underlying routing requests for the handshake has considerably saved the sensor's battery power with a good threshold of security [19, 20].

In [21] the authors designed an adaptive virtual area partition clustering routing protocol using ant colony optimization (AVAPCR-ACO), which used a virtual area partition scheme to cluster the network, and took advantage of the ant colony optimization to build a routing path among the cluster heads. In [22] the authors presented a mechanism for the wireless sensor network routing which can be more effective

regarding the criteria of route length, end-to-end delay, and network node energy for the quality of mechanism service. The proposed method uses the ant colony-based routing algorithm and the local enquiry to find optimal routes. Also, a fuzzy inference system was used to determine the route quality which showed better performance compared with the equation of route quality. Chakraborty et al. [23] presented a novel trust-based congestion aware routing algorithm for WSNs in which the optimum route for data packet transfer is dynamically selected by the TC-ACO algorithm, on the basis of the trust, congestion level, and internodal distance of the sensor nodes.

The rest of the paper is organized as follows. In the next section, we describe the system model and problem specification. In Section 3, we illustrate our proposed algorithm and complete the representation of our solution with samples. Section 4 presents the simulation and results of simulation by different charts. In the final section, which is Section 5, we offer the conclusion and some recommendations for future works.

2. System Model and Problem Specification

In this section, we aim to describe our system model that has been used in this research and problems that we are going to address.

2.1. Network Model and Assumptions. In this research, we consider a sensor network consisting of N sensor nodes which are randomly deployed over an environment. Sensor nodes collect the sensing data from the surrounding environment and send these collected data to BS. We suppose our network model has been restricted by some assumptions.

Assumption 1. All nodes are homogeneous and have unique identification and they are also initially deployed.

Assumption 2. All nodes have the same initial energy.

Assumption 3. In the initialization status, nodes do not have the information about each other such as location.

Assumption 4. Each node acts as a router and also it is able to sense the surrounding environment.

Assumption 5. Each node is able to communicate with other nodes in its transmission range. The maximum distance between two nodes is covered by the transmission range. The transmission range is determined by the signal strength. Thus, higher transmission powers will increase the number of nodes shared with the medium (connectivity degree) [24], and consequently, the probability of finding the destination among the node neighbors will be heightened. Of course, higher transmission range will have more disadvantages like collision problems. When two or more nodes want to send a packet at the same time over the same transmission medium or channel, collision will occur.

Assumption 6. All nodes have a two-ray-amp antenna model and will be able to communicate with other nodes in its transmission range.

Assumption 7. We used two different message delivery semantics: broadcasting and unicasting. In the broadcasting scheme, the sender sends the packets in its transmission range and all nodes in this transmission range receive these packets and send the packets to one special node.

2.2. Sensor Nodes and Energy Consumption Model. Sensors are usually wireless electronic elements and functions for sensing interactions. Sensors are located in the environment for data-gathering purposes (such as temperature, sound, vibration, pressure, motion, or pollutants) from the monitored area [25]. In recent years, with progress in technology, the wireless devices have been smaller, cheaper, and more powerful. Wireless sensor networks (WSNs) consist of sensor nodes that are designed with special purposes and applications (scientific, monitoring, or military purposes) [25].

In WSN, nodes are usually homogenous and consist of some units such as battery, sensors, transceiver, processor, and memory [26, 27]. The nodes send the collected data to the base station (BS). The BS is responsible for collecting data sent by the other nodes usually located in the center of the environment and sometimes has different units in comparison with the other nodes. Due to its interaction with other nodes and in some cases local data processing, the BS has power source with greater energy, larger memory, or maybe stronger processor. In the real system, all units of each node cooperate in doing delegated tasks that has effect on node energy consumption and in general on the network energy [26]. It should be noted that each node has a limited and irreparable energy source and if a node is turned off due to the completion of power source, it will reduce the connectivity degree and in some cases fragmentation in the network.

2.2.1. Sensor Nodes Behavior. In general, each node is placed in one of the two mechanisms based on the current state: active or sleep. In the active mechanism, a node uses efficient protocol on the network energy instead of turning off the transceiver for saving energy, while a node in the sleep mechanism has no interaction with other network nodes due to the fact that its transceiver is turned off and that the node energy consumption is lower [28].

In the active mechanism, each node will be in one of the three operational modes: transmit, receive, or idle. In the first mode (transmit), more node energies are consumed for turning on the transceiver and packet's transmit. In the second mode (receive), the node with its transceiver turned on receives a packet, demodulation, and decoding [29] where these operations (packet processing and turned-on transceiver) cause energy consumption in the node. After the packet's receiving or sending operations, a node is placed in the idle mode. In the idle mode, each node listens to the communication channel without any sending or receiving in an active manner. In this mode, some functions

Table 1: Some of battery types and some technical specifications.

Battery type	Average voltage during discharge	Milli-amp hours (mAh)	Watt-hours (Wh)	Joules (J)	Weight
Alkaline long life	1.225	1150	1.41	5071	12
	1.225	2122	2.60	9360	24
Nickel cadmium	1.2	300	0.36	1296	11
	1.2	1000	1.20	4320	11
Carbon zinc	1.1	320	0.35	1268	9.7
	1.1	591	0.65	2340	19

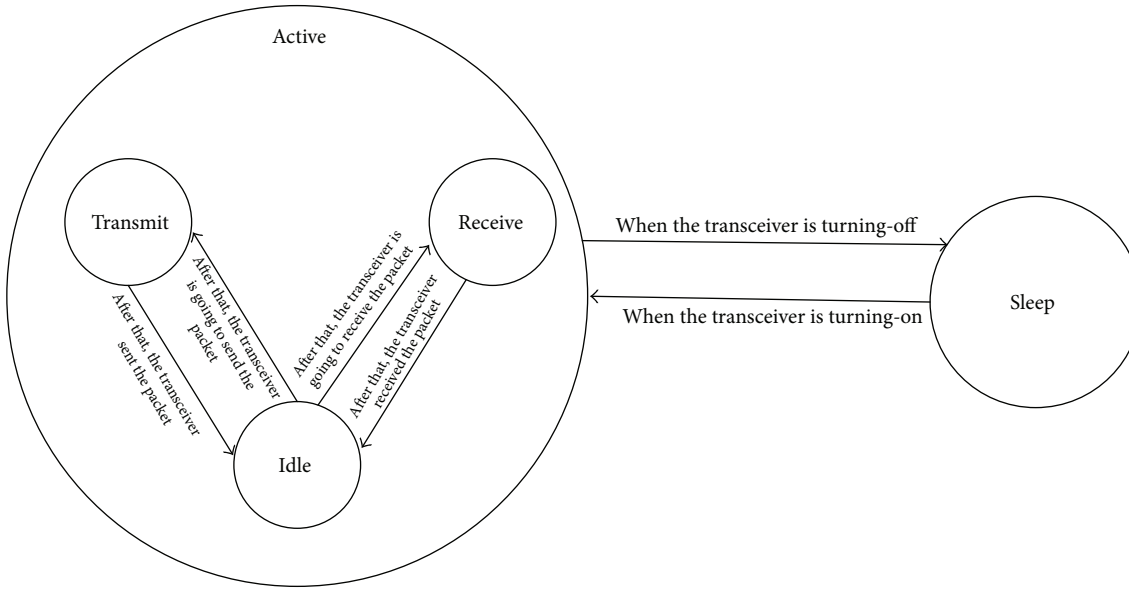


Figure 1: Node states.

in the hardware can be switched off, but all circuits are maintained to be ready to operate. Figure 1 illustrates these states and communications.

2.2.2. Energy Consumption Model. The energy is one of the most important aspects in the WSNs; when a node loses its energy by any reason, in the worst condition, a big part of network may be misused due to fragmentation. The battery source provides all the energy that a node needs for doing its tasks. Currently, many different types of battery exist for the sensor nodes [30] that they come in various sizes, weights, amounts of energy saved, construction technology, and so forth. Table 1 shows some different types of batteries and technical specifications among them. Due to these limitations in the node battery source, generating an energy model for preventing more energy consumption in WSNs is necessary. The energy model only maintains the total energy and does not maintain the radio states [31]. We used the energy consumption model that has been shown in [32]. The energy spent for the transmission of a l-bit packet over distance d is calculated by the following equation:

$$\begin{aligned} E_{\mathrm{Tx}}(l,d) &= E_{\mathrm{Tx\text{-}elec}}(l) + E_{\mathrm{Tx\text{-}amp}}(l,d) \\ &= l \times E_{\mathrm{elec}} + \left(l \times \varepsilon_{\mathrm{two\text{-}ray\text{-}amp}} \times d^4\right). \end{aligned} \tag{1}$$

The energy consumed by the transceiver for receiving a packet is calculated by the following equation:

$$E_{\mathrm{Rx}}(l) + E_{\mathrm{Rx\text{-}elec}}(l) = l \times E_{\mathrm{elec}}. \tag{2}$$

The item E_{elec} represents the energy consumption of transceiver dissipation and the $l\varepsilon_{\mathrm{two\text{-}ray\text{-}amp}}$ represents the energy consumption for the amplifying transceiver, where $\varepsilon_{\mathrm{two\text{-}ray\text{-}amp}}$ is a constant value and d^4 is a power loss for the channel model, depending on the distance between the transmitter and receiver. The transmit power (E_{Tx}) is the power consumed by the transceiver to transmit a data packet. The size of the data packet determines the transmit power. The bigger the packet size, the more the power consumed. Receiving power (E_{Rx}) is the energy used for receiving packet by the node. The energy cost $E(\rho)$ for a general route ρ is computed by the following equation:

$$E(\rho) = \sum_{i \in \rho} \left(E^i_{\mathrm{Tx}} + E^i_{\mathrm{Rx}}\right). \tag{3}$$

In this equation the final energy consumption amount is equal to the summation of E^i_{Tx} and E^i_{Rx} for each node with the identification i on route ρ. The route includes the sender and routers' node (intermediate nodes until the BS) without the BS.

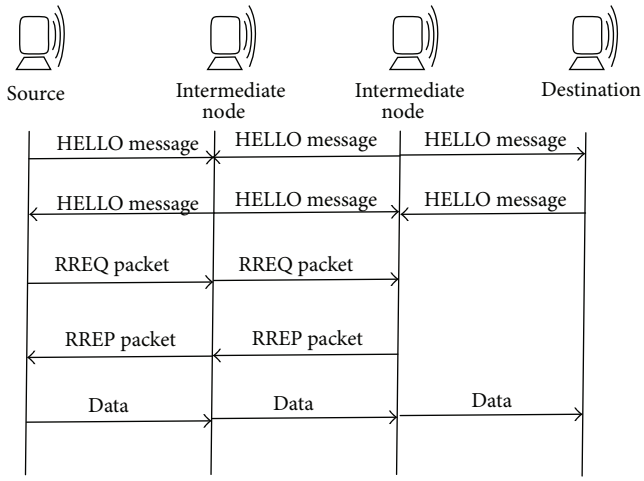


Figure 2: AODV protocol messages.

2.3. WSN Protocols and Improvements. In this section, some protocols that are used in WSNs will be explained like AODV for routing, IEEE 802.11 for MAC layer, and CBR for traffic control. In the last subsection, we discuss all improvements where we are going to reduce the energy consumption amount by applying them in WSNs.

2.3.1. AODV Routing Overview. Ad hoc on-demand distance vector (AODV) is one of the most used reactive routing protocols for WSN. In the reactive routing, paths are determined only when needed [33]. The primary objectives of this protocol are (a) to broadcast discovery packets when needed, (b) to distinguish between local connectivity management (neighborhood detection) and general topology maintenance [34], and (c) to propagate information on node connectivity degree for its neighbors or other interest nodes. Figure 2 depicts the messages that are used in the AODV protocol. A node which is aware of its surrounding environment (e.g., neighbor nodes) locally broadcasts a HELLO message; also the route request (RREQ) packets are sent if a sender is finding a route to BS. In this case, the path is made by route reply (RREP) packet unicasting to sender.

The AODV uses the route discovery mechanism with broadcasting instead of the source routing. Each node has a local routing table (RT) for quick response time to requests and establishment. Each row of RT shows the next hop from this node to the destination. The route discovery process is implemented when a node needs to communicate with other nodes and route information does not exist in its RT. The protocol uses the sequence number for more maintenance of the routing information among nodes. This sequence number will cause the efficient use of network bandwidth by minimizing the network load for the control and data traffic.

When a node wishes to send data to the BS, the source node creates a RREQ packet. This packet contains the source node's IP address, source node's current sequence number, the destination IP address, and destination sequence number that are broadcast in the source transmission range. Broadcasting is done via flooding. Finally, this packet will receive a node

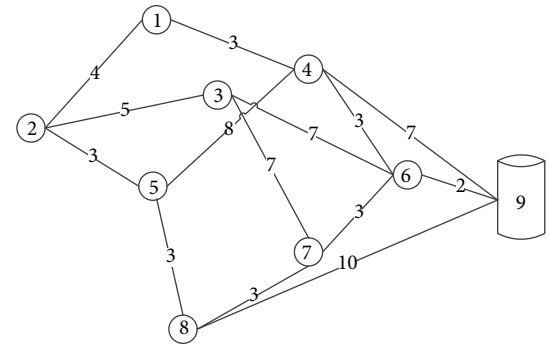


Figure 3: A sample of WNS.

that possesses a current route to the destination. Firstly, the receiver checks the RREQ packet. If the intermediate node has an entry in RT for the desired destination, the intermediate node will make a decision by comparing its entry sequence number with RREQ packet sequence number. If the intermediate node has not received this RREQ before, meaning that it is not the destination and does not have a current route to the destination and RREQ sequence number is bigger than saved sequence number, it rebroadcasts the RREQ [33, 34]. When the intermediate node is capable of replying, it means that it has a sequence number greater than or equal to that contained in the RREQ. If a node receives a packet with the same broadcast ID and source address, it drops this RREQ packet (sometimes maybe a node receives multiple copies of the same RREQ packet) [34].

Once an intermediate node receives a RREQ and its destination, the node sets up a reverse route entry for the source node in its RT. Reverse route entry consists of source IP address, source sequence number, number of hops to source node, and IP address of node from which RREQ is received. Using the reverse route, a node can send a RREP packet to the source. The RREP is unicast in a hop-by-hop model to the source [33]. While the PREP backward to the source, each intermediate node creates a route to the destination and sets up a forward pointer to the node from which the RREP comes.

Node 2, as Figure 3 shows, decided to send the packets to node 9. In the first step after node 2 checked its RT and it does not find any path to BS, it makes RREQ packets and broadcasts these packets. Nodes 1, 3, and 5 receive this packet and since these are not destinations, they will broadcast the RREQ packet for their neighbors. This event will be repeated for nodes 4, 6, 7, and 8. Finally, when node 9 receives this packet for the first time, it makes RREP and unicasts RREP hop-by-hop to node 2 and the intermediate nodes update its RT and create a route to the destination. After the routing discovery is finished, node 2 sends its data to node 9 on this discovered path.

2.3.2. MAC Layer. To avoid collision, the medium access control (MAC) protocols have developed and control how each node could access the channel. IEEE 802.11 is a distributed MAC scheme that works based on the carrier sense

multiple accesses with collision avoidance (CSMA/CA). In this scheme, each node accesses the medium on a contention basis. Before a data transmission begins, the sender and receiver must have a RTS-CTS signaling handshake to reserve the channel [35]. A sender for sending a packet, firstly, with its transceiver senses the channel and looks up its network allocation vector (NAV). Therefore if the channel is free, it sends a RTS (request-to-send) to the destination node, and the destination with a CTS (clear-to-send) approves RTS receiving; otherwise it has to wait until the channel is busy. As the CTS is received by receiver, it can start the data transmission and destination, confirming that data are successfully received via acknowledgment (ACK) [35].

2.3.3. Traffic Protocol. The constant bit rate (CBR) service category is used to transport traffic connections at a constant bit rate, where there is an inherent reliance on the time synchronization between the traffic source and destination. The consistent availability of a fixed quantity of bandwidth is considered appropriate for the CBR service. Cells which are delayed beyond the value specified by the cell transfer delay (CTD) are assumed to be significantly of less value to the application. In the CBR, bandwidth guarantees the peak cell rate of the application.

2.3.4. Improvement in the Original AODV Routing. As it has been mentioned, the AODV is one more used network routing protocol. The AODV protocol uses the message broadcasting delivery with the flooding technique. Finding the methods for reducing energy consumption will cause an increase of the network lifetime and it will be an improvement for WSNs. The following objectives are set in this research.

Improvement 1. In the original AODV, each sender broadcasts RREQ packets. This is costly for the network and it causes more energy consumption and, furthermore, busy network bandwidth. Reducing the number of RREQ packets will be an improvement, which decreases the energy consumption amount, increases network lifetime, and controls the overhead in the network.

Improvement 2. By focusing on the broadcasting of the RREQ packet by each node, for all its neighbors in the original AODV, several routes may exist between the sender and BS. Due to this reason, several RREP packets tend to return to the sender. We try to reduce the number of RREP packets and it proves to be another improvement.

Improvement 3. Another improvement is selecting the best node among the neighbor senders. If an algorithm involves some extra parameters, in addition to the neighborhood like remaining energy, connectivity degree, distance, etc., for selecting next hop, these extra parameters, due to the increased packet delivery speed, will increase energy saving in the network. Also, this technique avoids the route discovery repetition.

3. Proposed Solution

In this section, our proposed algorithm will be explained. In Section 3.1, we describe utilization techniques (fuzzy logic and ant colony) in this research. Section 3.2 clears the proposed algorithm by the pseudocode and the last section illustrates a sample of our algorithm.

3.1. Ant Model and Fuzzy Logic Model. When a node wishes to send collected data to BS as has been stated in Section 2.3.1, it makes a RREQ packet and broadcasts it to all neighbors. We used two kinds of ants: forward and backward ants that have been presented by FAnt and BAnt. RREQ packet is used for modeling the FAnt and BAnt. In this research, we have several FAnts (depending on the source neighbor's number) and a BAntthat will be discussed in the next sections. FAnt tries to find a route from the sender to the BS (a FAnt is launched from the source and intermediate nodes until the BS). While FAnt moves forward, it saves the list of nodes that has been visited in its memory and tries to avoid traveling the same node and BAnt fixes this route. In our algorithm, we restricted and targeted the number of RREQ packets (FAnts) sent by each node in compared with the original AODV. The winner FAnt makes a BAnt with the same parameters and BAnt returns to the source and updates RT and pheromone concentration amount for each node in its path. We used the pheromone rules the same way as in [36].

BAnt when traversing from a node will increase the pheromone as shown in (4), where α is the variable parameter, hm represents the maximum hops between sources to destination, hc is the remaining hops to destination, and En represents node remaining energy. We added the node remaining energy to this equation based on the reason that the ants select the node with greater power. Consider

$$\Delta\tau = \alpha \times (\mathrm{hm} - \mathrm{hc}) \times \mathrm{En}. \tag{4}$$

When the node with identifier n transmits the packet, it updates the pheromone concentration amount with the following equation:

$$\tau_n = (1 - \rho) \times \tau_n + \Delta\tau, \tag{5}$$

where ρ is the pheromone evaporation (pheromone amounts evaporate very soon when no ants traverse from this path) coefficient, the range of ρ is $[0, 1]$, and $1 - \rho$ is the pheromone residue factor. We added a structure to FAnt and BAnt to save the traveling path in addition to the original fields which used AODV for RREQ and RREP packets.

Fuzzy system consists of three parts [37]: fuzzification, inference engine, and defuzzification. Figure 4 shows the fuzzy system factors used in this paper. In a fuzzy logic-based system, calculations are performed by an inference engine. In order to select the inference engine, we have studied two widespread approaches presented in the literature: Mamdani [38] and TSK [39]. Here we use the Mamdani inference engine. The input of a Mamdani fuzzy logic system is usually a crisp value.

(i) Fuzzification. Input variables by membership functions should be converted to linguistic values to determine

Table 2: Some of uses rules.

Rules database					
Rule number	ReE	Dis	ConD	PheA	Out
1	High	Near	High	High	Very good
2	High	Moderate	Medium	High	Good
3	Low	Far	Low	Low	Very bad
4	High	Far	High	Low	Bad

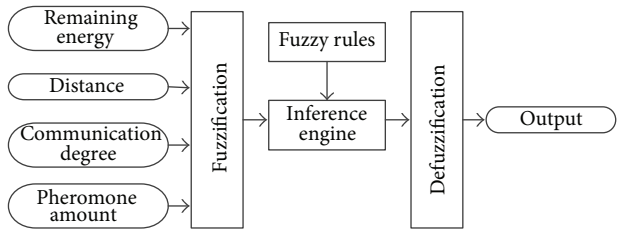


Figure 4: Fuzzy logic system model.

the membership degree. The outputs of this stage are fuzzy values that can be processed by the inference engine.

(ii) Inference Engine. At this stage, the input fuzzy values from the previous stage apply to all fuzzy rules that are kept in the fuzzy database by the inference engine. All rules that are adaptable to input parameters will be activated and then the output of all activated rules will be aggregated. All rules are in the IF-THEN form and here we have 64 different rules, where Table 2 shows some samples of the rules used in this research.

(iii) Defuzzification. The inputs of this phase are those linguistic values and the outputs are nonfuzzy values. Mamdani uses the centroid technique which tries to determine the point where a vertical line divides the combined set into two equal parts. We use the defuzzification formula as follows:

$$Y = \sum_{i=0}^{63} \left((5 * \mathrm{Ph}_i) + (1.5 * \mathrm{Re}_i) + (1.5 * \mathrm{CD}_i) + (1 * \mathrm{Dis}_i)\right) * O_i, \tag{6}$$

$$w = \frac{Y}{\sum_{j=0}^{63} O_j}. \tag{7}$$

In (6) Ph_i is the pheromone concentration amount, Re_i is the remaining energy, CD_i is the connectivity degree, Dis_i is the destination between 2 nodes, and O_i is the output value for rule number i (we will have a full discussion about these parameters in the section below). Finally, a fuzzy logic value is obtained for all 64 rules by (7) (Y is divided by the whole position output of all rules).

3.1.1. Energy Optimized Parameters. If FAnts act intelligently in choosing the next node, we will be nearer to our aim. We aim to assist FAnts in choosing the next node with fuzzy logic. By this method, the FAnts, in addition to the pheromone concentration amount for choosing next node, will use the extra parameters, where these parameters contribute to the more intelligent choices. In this proposal, we focus our attention on factors such as remaining energy of each node as well as the connectivity degree of the same node (the number of neighbors), the distance of node to its neighbors, and then its pheromone concentration amount. These factors are briefly discussed as follows.

Table 3: Linguistic values for input parameters.

Input parameter	Linguistic value
ReE (remaining energy)	Low, medium, and high
Dis (distance)	Near, moderate, and far
ConD (connectivity degree)	Low, medium, and high
PheA (pheromone concentration amount)	Low, medium, and high

(i) Remaining Energy. We know that each node has a battery with limited power, and energy saving for batteries is essential for node survival and network maintenance. On the other hand, as we will select the node with the highest remaining energy, the network lifetime will increase. We assume impact factor 1.5 for the remaining energy.

(ii) Connectivity Degree. The numbers of node neighbors which are in the transmission range of the node are defined as the connectivity degree of the node. This is important because if a node with a higher degree is elected, firstly, with higher probability we can find the destination with this node and, secondly, perhaps the destination is one of its neighbors that can decrease the length of the path as well as reducing the energy consumption of the whole network. Also, we use impact factor 1.5 for connectivity degree.

(iii) Distance of Node. As the distance between two nodes within the transmission range grows, they need more energy to send/receive data, but with high probability the destination is nearest to this node. The distance between the current node to this neighbor calculates with (8) that in this formula (x, y) shows the location of current node and (x_n, y_n) is the location of the destination (here one of current node neighbor). This list has a row for each current node neighbor. We use the impact factor 1 for it. Consider

$$N - \mathrm{distance} = \sqrt{(x_n - x)^2 + (y_n - y)^2}. \tag{8}$$

(iv) Pheromone Concentration Amount. It is the most important factor for selecting the next neighbor node in the network environment. Surely, the ants will select the neighbor node with the higher pheromone concentration amount because more biotypes use this path for arriving to their destinations in the past. The impact factor for it is equal to 5.

3.1.2. Linguistic Values and Membership Functions. As Table 3 shows, each factor is being presented by three linguistic values. Linguistic values, namely, *low, medium,* and *high,*

Table 4: Linguistic values for output parameter.

Output parameter	Linguistic value
Out (output)	Very bad, bad, good, and very good

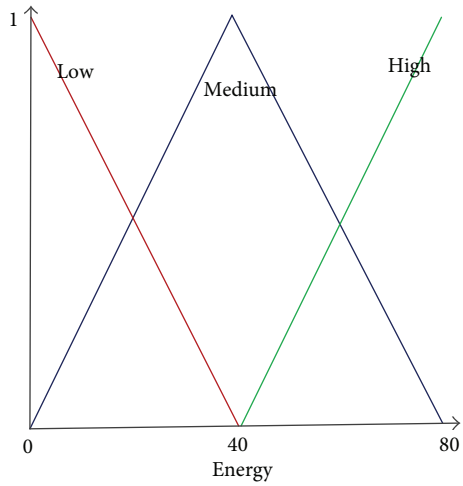


Figure 5: Battery level fuzzy sets.

are used for input parameters of the remaining energy, connectivity degree, and pheromone concentration amount. The use of *near, moderate,* and *far* for distance; also the values of *very bad, bad, good,* and *very good* are used for output parameters that are indicated by Table 4.

Figures 5, 6, 7, and 8, respectively, show the membership functions of the remaining energy, connectivity degree, distance, and pheromone concentration amount. Figure 9 depicts the membership function of the output unit before the defuzzification of results. The input parameters are taken by a membership function with degree one and it becomes a fuzzy value.

3.2. Proposed Algorithm. In our algorithm, each FAnt tries to find the optimal path from the source to destination. In the WSN destination, there is always a BS. The ant for delivering a packet has to move to some nodes that are in the source neighborhood. Each node keeps different structures such as routing table (RT) and neighbors' list (NL). RT (see Table 5) contains special information about node neighbors. The size of RT is different for each node, but for all nodes the fields are similar. In this structure, the first field indicates the destination node identifier (Dis-ID), the second indicates the next hop identifier, and the last, the hop count, shows the traveled hop from the source to this node. NL is a structure that keeps the node neighbors.

Due to the static structure of the WSN, the identified path by ants is always an optimal path. The optimal path is one of the existing paths between the source and destination with the fastest speed, the lowest cost (such as node energy consumption), and the most stable packet delivery. If we presume that nodes have had a little movement, the selected path needs to be reselected in the next communication. We presume in this paper that all nodes are fixed. When

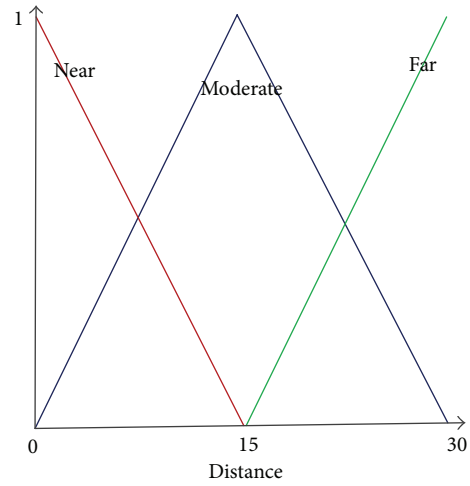


Figure 6: Distance fuzzy set.

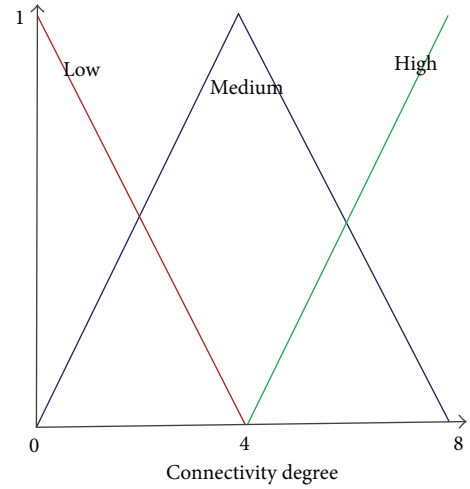


Figure 7: Connectivity degree fuzzy set.

the path is discovered, the data are transmitted on this path between the source and destination. It is clear that the data transmission on a fixed path will cause more energy consumption for routers and sometimes may cause some routers to lose all the power. Our algorithm will repeat the path discovery operation in the determined time to prevent this. Below, we describe our algorithm in two phases and three steps.

Phase 1 (identification).

Step 1 (neighbor's identification). After that, a node which decides to send a packet to the determined receiver or BS triggers this step. The sender broadcasts a HELLO message in the transmission, rings the identifying neighbors, and creates or updates its NL. By this message the sender sends its identifier (S-Id) and sets M-Type (see Table 6). We used the M-Type field to distinguish between messages broadcasted by sender to neighbors and reply message sent by the receiver to sender. Each receiver which receives this message changes

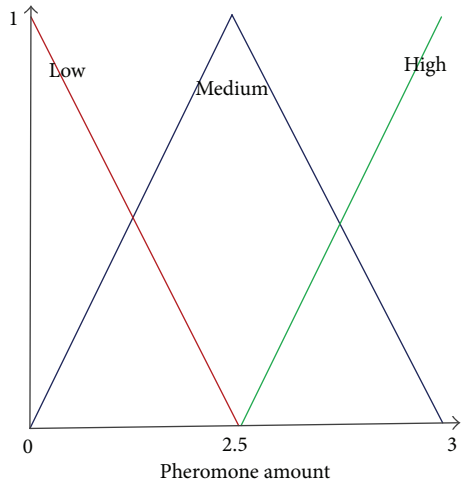


Figure 8: Pheromone fuzzy set.

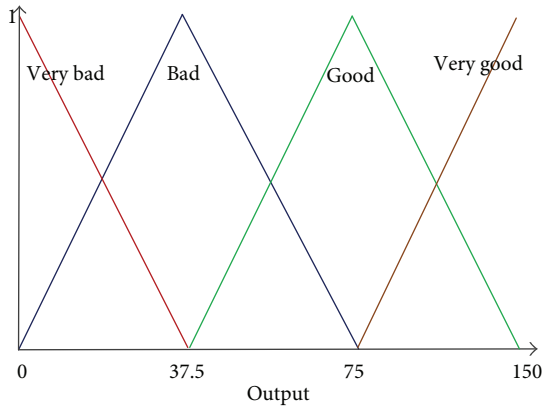


Figure 9: Output fuzzy set.

the M-Type field values by one and unicasts its identifier (N-Id) for the sender. When the primary sender receives the HELLO message with M-Type value equal to one, firstly, it searches in its NL and if this node is not identified before, this node is added in NL. Pseudocode 1 depicts the pseudocode that we used.

Step 2 (route discovery). In this step, as Pseudocode 2 illustrates, our algorithm executes based on the fact that the node is the source or intermediate node. If this node is the source, it sends the FAnts to all neighbors that exist in its NL by unicasting. If this node is an intermediate node, FAnt is unicast for the next hop that has been recognized by the fuzzy logic system (firstly, we went to Step 1 to identify its neighbors and update its NL and then for all neighbors in its NL we calculated the fuzzy amount and selected the node with the biggest fuzzy amount as the next hop). Each FAnt aggregates this new fuzzy amount that is obtained for the next hop with the sum of other fuzzy amounts obtained from other intermediate nodes, adds the current node identifier to the list

Table 5: Routing table structure.

Dis-ID	Next hop	Hop count
	∗Next	

Table 6: Routing table structure.

S-Id/N-Id	M-Type

of traveled nodes, and also has in its memory a field for saving the fuzzy amount.

When the FAnt has reached its destination, if the current node identifier is equal to the BS identifier, we start a timer and wait until other FAnts arrive at the BS. After this timer is expired, the BS selects the best FAnt based on these fuzzy amounts that each ant has its memory, kills other FAnts, creates a BAnt, copies the FAnt path (path-list) in the BAnt path (Dis-list), and goes to the next step. The BS for selecting the winner FAnt, firstly, reads the hop count values from each FAnts memory and takes it in a list for each FAnt independently. Then, with this tip in mind, the FAnt with a shorter hop count is better than FAnts with the largest hop count and it deletes the FAnts with a larger hop count value. Later, the FAnts with equal hop count will remain (it is possible that a FAnt will be reminded that it has the shortest hop count); the final work for selecting the winner FAnt selects the FAnt with the largest fuzzy amount. The theory behind this selection is that this path is the optimal path with the shortest hops, fastest packet delivery, and lowest energy consumption. Also, in this step, if each FAnt finds a loop or cannot find a path to the destination, the FAnt will kill itself.

Step 3 (rollback). As Pseudocode 3 shows, BAnt returns to the source hop-by-hop and updates RT and pheromone concentration amount of all nodes in its path. The traveling path has been saved in the ant memory (Dis-list). Other ants which prevent higher network energy consumption will be killed by the BS in the next step. The latest identifier in this list is the identifier of the first node that the BAnt should visit. The BAnt sets its destination, removes this identifier, increases the pheromone amount (*Ph-Amount* field value) with (4), updates the RT entry for this route, and moves to its destination. This work will continue until this list becomes empty, and when this list becomes empty the BAnt will surely arrive at the source node.

In this step, we used the acknowledgment system to guarantee message delivery. Each node that currently has BAnt saves a copy of it in its memory, then sends it, and waits until it receives an acknowledge packet from the receiver node in determined time; if the sender receives this acknowledge packet it will remove the BAnt's copy; otherwise it will resend this packet.

Phase 2 (relaxing). Finally, nodes do not need to do something new to send the data packets to the destination (Pseudocode 4). They are looking at their RT and select the next hop. Notably, when the packet arrives at a node, it increases its pheromone amount (*Ph-Amount* field value) by

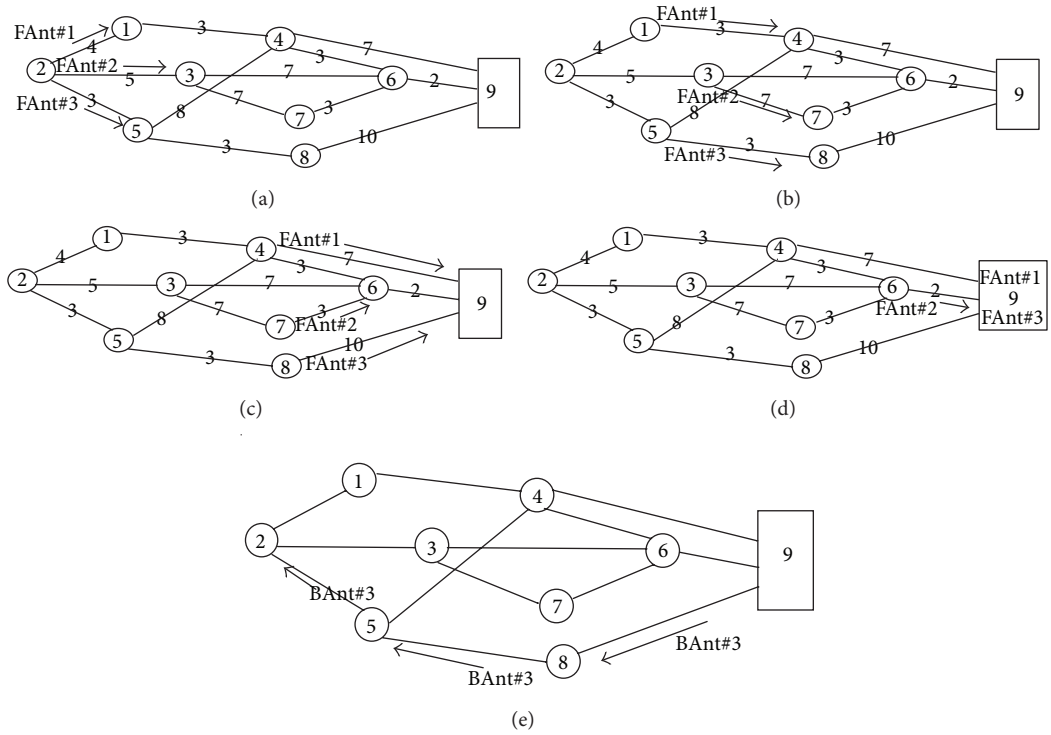


Figure 10: Sample of our algorithm mechanism.

```
SendHELLO (node i, packet P) {
    P.id = i;
    P.M-Type = 0;
    Broadcast p;
}
ReceivedHELLO (packet P) {//node j received packet P from node i
        If (P.M-Type == 0) {
            P.id = current-id;
            P.M-Type = 1;
            Unicast P; // packet P send for sender node;
        }
 Else If (P.M-Type == 1) {
    Rez = Search P.id in current node neighbors-list (NL)
    If (Rez == False) {// Node don't exist in current node Neighbor-list
          Create new neighbor row;
    NL.N-ID = P.id;
    }
 }
}
```

Pseudocode 1: Pseudocode for Step 1.

(5). In this phase, if, by any reason, such as the node dies or due to the node movement, the path will be broken or if the repetition timer is expired, the source will repeat this algorithm and find another optimal path between the source and destination (BS). This operation increases the network lifetime as illustrated in Pseudocode 3.

3.3. FACOR Sample. Figure 10 illustrates the example of the WSN. As this Figure shows, the distance between two neighbors is written on their edge. For example, node number 2 will send a packet to node number 9 (BS). Table 7 illustrates a snapshot of the environment when node 2 decides to send a packet to 9. As this table shows, each node (informing all

```
SendFAnt (Forward-Ant-Packet FAnt, Node Next-Hop, Node Current-Node-ID) {
        If (current-Id == source) {
          NLPointer = First;
          While (NLPointer != NULL) {
                  Create Forward-Ant-Packet FAP;
                  FAnt.id = NLPointer.N-ID;
                  FAnt.dis = Destination;
                  Unicast FAP;
                  NLPointer = NLPointer.next;
           }
        }
 Else {
        FAnt.dis = Next-hop;
        Unicast FAnt;
 }
}
ReceivedFAnt (Forward-Ant-Packet FAnt) {
  If (FAnt.dis != current-Id) {
      Create HELLOMESSAGE packet P;
      SendHELLO (current-Id, P);
      NLPointer = First;
      J = 0;
      FuzzyArray [Neighbors-Count];
      FuzzyID [Neighbors-Count];
       While (NLPointer != NULL) {
          FuzzyArray [j] = Get-Fuzzy-Amount (NLPointer.N-ID);
          FuzzyID [j] = NLPointer.N-ID;
          NLPointer.N-ID = NLPointer.Next;
       }
  Max-amount = FuzzyArray [0];
  Max-id = FuzzyID [0];
  find = 0;
  For (int k = 1; k < l; K++)
      If ((Max-amount < FuzzyArray [k]) And (FuzzyID [k] does not exist in FAnt.Path-list)) {
        Max-amount = FuzzyArray [k];
        Max-id = FuzzyID [k];
        find = 1;
      }
 If ((find == 1) Or (FuzzyID [k] does not exist in FAnt.Path-list)){
    FAnt.Nexthopee = maxid;
    Add current-Id to FAnt.Path-list;
    FAnt.fuzzy-amount += Max-amount;
    SendFAnt (FAnt, Max-id);
 }
 Else if ((find == 0) Or (FuzzyID [0] does not exist in FAnt.Path-list)){
      FAnt kill itself;
    }
      }
 Else if (FAnt.dis == current-Id) {
    Wait (R-Timer); // wait till all ant arrive
    Select winner ant;
    Delete other FAnts;
      Create Backward-Ant Packet BAnt;
      BAnt.Id = FAnt.Id;
      Next-hop = FAnt.Path-list [0];
      Delete FAnt.Path-list [0];
      BAnt.Dis-list = FAnt.Path-list;
      BAnt.Nexthop = Next-hop;
    SendBAnt (BAnt, Next-hop);
 }
```

Pseudocode 2: Continued.

```
}
Get-Fuzzy-Amount (Node id) {
    En = get-node-energy (id);
    Dist = get-distance (Current-Node, id) by (8);
    Pher = get-Pheromone (id);
    Conn = get-connectivity (id);
    Calculate fuzzy amount based on (7);
    Return fuzzy amount;
}
```

Pseudocode 2: Pseudocode for Step 2.

```
SendBAnt (Backward-Ant Packet BAnt, Node Next) {
  Unicast (BAnt, Next);
}
ReceiveBAnt (Backward-Ant Packet BAnt) {
  If (BAnt.Nexthop == Source-Node-ID) {
      Update source node RT;
      Increase Ph-Amount for source by (4)
  Delete BAnt;
      Start Repeat-Timer
  }
  Else {
      Update current node RT entry;
      Increase Ph-Amount for current node by (4)
      Next-hop = BAnt.Dis-list [0];
      Delete BAnt.Dis-list [0];
      BAnt.Nexthop = Next-hop;
      SendBAnt (BAnt, Next-hop);
  }
}
```

Pseudocode 3: Pseudocode for Step 3.

```
ReceiveData (Data-packet dp) {
 Increase pheromone amount by (5)
 find next hop on it RT
 check next-hop status
  if (next-hop be alive)
      Unicast (dp, next-hop);
  else if (next-hop not be alive or Repeat-Timer is expired)
      repeat route discovery algorithm
}
```

Pseudocode 4: Pseudocode for Phase 2.

four basic factors in Section 3.1.1) will decide for the next hop. The last column in this table shows the calculated fuzzy logic values for all nodes at a time of simulation. We should pay attention to this point that fuzzy logic values will be calculated independently for each ant when the ant wants to select the next hop in all simulation times.

Tables 8, 9, and 10 determine the final parameters that each FAnt has when it reaches the receiver (node 9) at the end of Phase 2. If node 2 decides to send a packet, it must follow our steps. Based on these tables in the receiver, FAnt 3 wins this competition and returns to sender for updating RTs. As we see in these tables, the fuzzy amount for FAnt 2 is bigger than the other two, but due to less hops count, FAnt 3 wins.

Figure 10 shows the result of our algorithm on a sample network. Figures 10(a) to 10(d) illustrates the paths that each ant takes for reaching the receiver. Each FAnt has its path and saves this path. The receiver sets a timer when it receives the first FAnt and waits for all FAnts. When the timer expires, the receiver selects the best FAnt, kills others, and makes BAnt (Figure 10(e)). The BAnt returns to the source and updates RT for each node in its path. Probably some FAnts are not able to find a path to the receiver (because the resulting graph of the network connectivity is not always a connected graph). In this situation the FAnt will kill itself when finding deadlock and then the receiver will not wait for many times because of its timer.

4. Simulation and Result

The performance of algorithm is evaluated by network simulator 2 (NS2) [40] and is implemented in two scenarios. NS2 is one of the most famous and most widely used network simulators. In this simulation, we consider a network with 10, 20, 30, 40, 50, 60, 70, and 80 nodes that are randomly placed in a dimension of 1500 M $*$ 1500 M area. Each simulation is running for 180 seconds of the simulation time. As aforementioned, we implemented two scenarios based on the node transmission range (100 M transmission range for scenario 1 and 300 M transmission range for scenario 2) and repeated the simulation for different number of nodes in each scenario. Table 11 lists the simulation parameters. All nodes were taken randomly in the environment and we only controlled the hop count in the simulation time.

Focusing on the different node states, we assume different parameters for each state as shown by Table 11. We have two different values for field, first $1.42681e-12$ when the transmission range is 100 M and then $8.91754e-10$ for 200 M transmission range.

4.1. Energy Consumption. The amount of energy consumption in the network depends on the amount of energy required to transmit a message from a sender to a receiver.

Table 7: A snapshot of our environment when node 2 decided to send a packet to node 9.

Node ID	Neighbor ID	Pheromone amount	Remaining energy (Jul)	Neighborscount	Distance (meter)	Fuzzy logic value
2	1	1.5	1058.32	2	4	1601.98
	3	2.0	1079.96	3	5	1638.44
	5	2.3	1046.43	3	3	1589.65
1	4	0	1022.68	3	3	1542.52
	2	0	1058.32	3	4	1595.98
3	6	2.8	1046.44	4	7	1593.66
	7	2.4	1046.43	3	7	1590.15
	2	0	1058.32	3	5	1595.98
5	4	0	1022.68	3	8	1542.52
	8	2.6	1046.41	3	3	1591.12
	2	0	1058.32	3	3	1595.98
4	6	2.8	1046.44	4	3	1593.66
	9	2.8	1053.42	3	7	1602.63
	1	1.5	1058.32	2	3	1601.98
7	6	2.8	1046.44	4	3	1593.66
	8	2.6	1046.41	3	3	1591.12
	3	2.0	1079.96	3	7	1638.44
8	7	2.4	1046.43	3	3	1590.15
	9	2.8	1053.42	3	10	1602.63
	5	2.3	1046.43	3	3	1589.65
6	3	2.0	1079.96	3	7	1638.44
	4	0	1022.68	3	3	1542.52
	7	2.4	1046.43	3	3	1590.15
	9	2.8	1053.42	3	2	1602.63
9	4	0	1022.68	3	7	1542.52
	6	2.8	1046.44	4	2	1593.66
	8	2.6	1046.41	3	10	1591.12

Table 8: Final parameters for FAnt#1.

N-Ids	Fuzzy amount	Hop count
1, 4, 7	4747.13	3

Table 9: Final parameters for FAnt#2.

N-Ids	Fuzzy amount	Hop count
3, 7, 6, 9	6424.88	4

Table 10: Final parameters for FAnt#3.

N-Ids	Fuzzy amount	Hop count
5, 8, 9	4783.39	3

Table 11: Simulation parameters.

Simulation environment	
Area	1500 m^2
Simulation time	180 seconds
Nodes	10, 20, 30, 40, 50, 60, 70, and 80
Nodes placement	Random
Mobility model	Fix
Traffic	CBR
MAC layer	IEEE 802.11
Transmission range	100 m and 200 m
Maximum battery power (initial energy)	1150 J
E_{Rx}	1.42681e – 12, 8.91754e– 10 wat
E_{Tx}	0.281838 watt

Surely, if a proposed algorithm is able to reduce energy consumption in the network, the network lifetime will increase. The numerical values that we used for both scenarios are illustrated in Table 11. The results of energy consumption for both scenarios are shown in Figure 11. There are many effective parameters on energy consumption in the network such as network setup time, routing setup time, CPU processing, transmitting packets, and receiving packets, and in this proposal we measure the final energy consumption for the network at the end of simulation time.

As this figure illustrates, when we have a larger transmission range, due to this larger transmission range, connectivity degree increases, energy consumption is lower, and the network lifetime will increase. We measured the algorithm

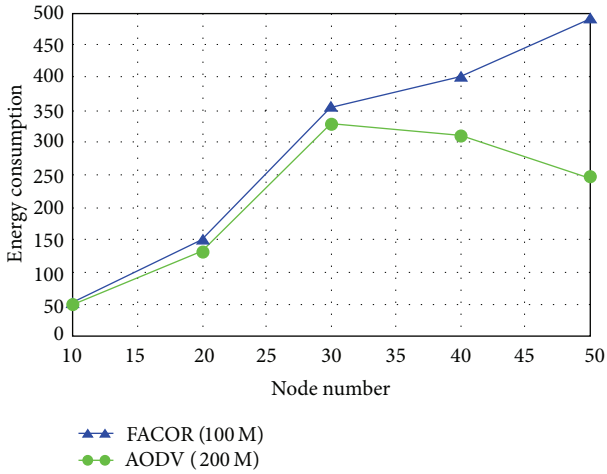


Figure 11: Results of our algorithm in two different scenarios.

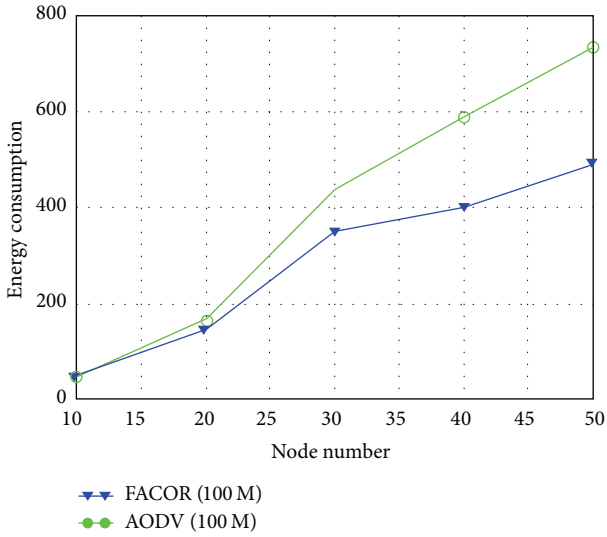


Figure 12: The energy consumption of route discovery.

of the energy consumption after 10 times for each group in different scenarios; also we assumed that the receiver is two hops away from the sender. Another result shows that, by increasing the node number for bigger transmission range, the energy saving will be better in the network (due to increased connectivity degree).

Here, our algorithm has been compared with the original AODV. As Figure 12 demonstrates, our algorithm has optimal performance compared to the original AODV and, approximately, they operate the same way when the number of nodes is lower than 20. This figure shows the AODV with message broadcasting, but our algorithm also shows a better performance in comparison with the AODV without the message broadcasting. As this figure shows, our algorithm for 50 nodes consumed about 500 J energy, but AODV in the same condition consumed about 700 J energy. In this comparison, also we assumed 100 M transmission range and two hops distance between the sender and receiver.

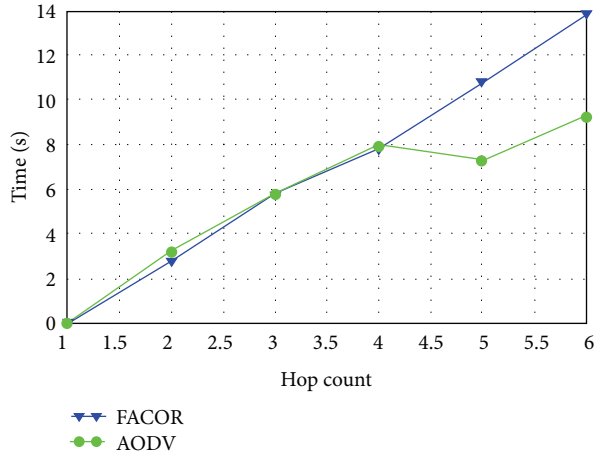


Figure 13: Routing setup time.

4.2. Routing Setup Time. This represents the time spent by a protocol to discover routes from sender to receiver. It is the time since the first discovery packet is sent till the sender has a route to the receiver. By this field, we will measure the time from the moment that the discovery phase starts until the moment that the winner ant updates all nodes in its path to sender.

Figure 13 illustrates the routing setup time for FACOR and original AODV in the same scenario with the transmission range of 100 M for 10 nodes. In this figure, we are going to compare the routing setup time for two algorithms in different hop counts between the sender and receiver. As this figure shows, our proposed method is suitable for bigger hop count, to find the optimal path between sender and receiver; because of waiting time for arriving ants will take bigger time for routing. Of course, this parameter depends on the sender neighbor's number because we broadcast ants for all sender neighbors and whenever the number of ants is less, the routing setup time will be faster. In our simulation we did not control the sender neighbors.

4.3. Average End-to-End Delay. This is the average time taken by a data packet to travel from the source to the destination. It will be a different value with the routing setup time, because the routing path has been determined before and, by this field, we will only measure the time from the moment that a packet has been sent by sender until the moment that packet will be received by the receiver. The end-to-end delay is compared in Figure 14. This Figure depicts our algorithm, although it has a bigger routing setup time for finding optimal path, and when the path was determined, the optimal path of the end-to-end delay or the packet delivery time was better than the original AODV. Paying heed to this point, we can solve the increased routing setup time issue, because the energy will consume more when sending and receiving packets. As Figure 14 shows, the end-to-end delay time is approximately equal when the hop count is below 2. In this figure also, we assumed 10 nodes in a 100 M transmission range.

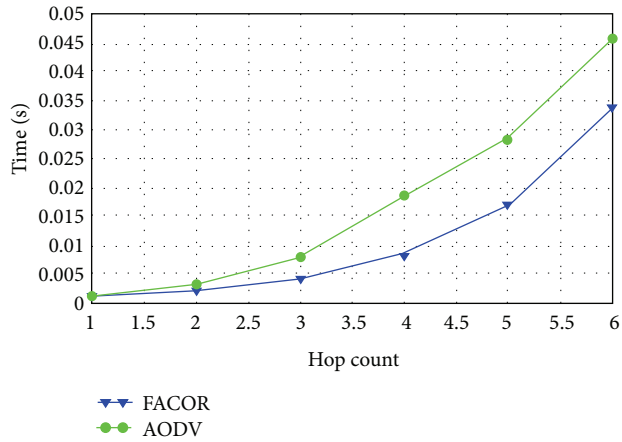


Figure 14: End-To-end delay.

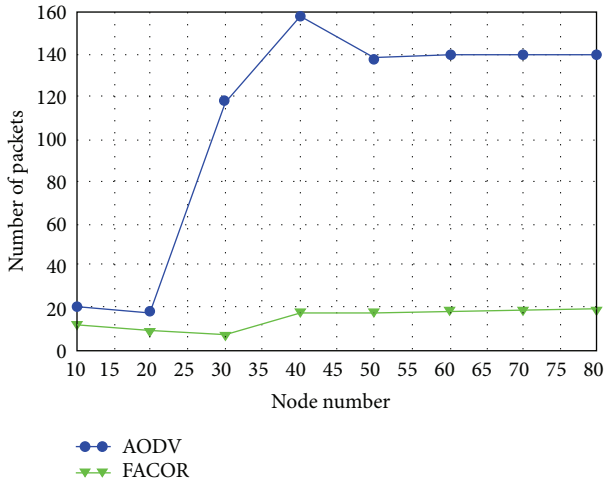


Figure 15: The number of packets.

4.4. Number of Packets. This field represents the amount of packets sent during the identification phase (RREQ packets). A low number of sent packets indicate that a protocol will be more efficient in terms of the energy spent during the route discovery. It is noteworthy that, due to this topic, ants in the first step (just for sender), the number of packets is extremely dependent on the sender neighbor's number. Also, this controls the amount of overhead on the network.

As we see in Figure 15, the number of packets for routing discovery in our proposal is the least. If the number of packets is the least, the amount of energy consumption will be lower. It is the number of packets since the discovery phase is started until ants find a route to the destination. Although the energy consumption amount for sending and receiving packet depends on the packet size and distance between the sender and receiver, our algorithm consumes the average of about 0.281 J for sending and receiving a packet. Also in Figure 15, we passed up the packets sent and received by broadcasting in our algorithm and original AODV.

Finally, some anomalies in the figures are due to the fact that the nodes are randomly placed in the environment. Furthermore, when we have a transmission range equal to 100 M, each node in any distances in this transmission range is a neighbor and maybe sometimes one neighbor has 2 M distance and another has 99 M distance, but both of them are a neighbor for a given node.

5. Conclusion and Future Work

Due to some limitations in sensor nodes such as limited power battery source and some commonly used operations like routing, designing of optimal and efficient algorithm appears to be necessary. In this paper, we propose a novel optimal routing algorithm for WSN with fuzzy logic ant optimization colony routing called FACOR. Our proposed algorithm based on ant's intelligent optimization select the short path between the sender and receiver. Simulation results show the correct operation of the protocol and its suitability to be used in a wide range of applications and scenarios. The performance of this proposal has been compared with the original AODV. In this work, we calculated the routing setup time, end-to-end delay for packet delivery, and the number of packets sent in the routing discovery phase and energy consumption. Our algorithm increases the network lifetime by reducing the nodes energy consumption and the number of packets. Also FACOR has proven its efficiency by achieving better average results than the other proposals. Finding a solution for routing problems (e.g., nodes failure) and the addition of new nodes in the network is an open area for future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank Professor Tien Van Do for the valuable and compassionate comments and guidance on doing this work.

References

[1] P. M. Fernandez, *Circuit switching in the Internet [Ph.D. thesis]*, Stanford University, 2003.

[2] H. Frey, S. Rührup, and I. Stojmenović, "Routing in wireless sensor networks," in *Guide to Wireless Sensor Networks*, pp. 81–111, Springer, Berlin, Germany, 2009.

[3] E. Amiri, H. Keshavarz, H. Heidari, E. Mohamadi, and H. Moradzadeh, "Intrusion detection systems in MANET: a review," in *Proceedings of the International Conference on Innovation, Management and Technology Research*, pp. 1–6, Malacca, Malaysia, 2013.

[4] H. Keshavarz, R. M. Noor, and E. Mostajeran, "Using routing table flag to improve performance of AODV routing protocol for VANETs environment," in *Proceedings of the 9th International Conference on Computing and Information Technology (IC2IT '13)*, pp. 73–82, May 2013.

[5] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1369–1379, 1999.

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[7] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, pp. 153–181, Springer, New York, NY, USA, 1996.

[8] M. Dorigo, *Optimization, learning and natural algorithms [Ph.D. thesis]*, Politecnico di Milano, Milan, Italy, 1992.

[9] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the 1st European Conference on Artificial Life*, pp. 134–142, 1991.

[10] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[11] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[12] M. Ghazanfari and G. M. Rezaei, *An Introduction to Fuzzy Sets Theory*, Iran University of Science and Technology Press, Tehran, Iran, 2006.

[13] A. M. Ortiz and T. Olivares, "Fuzzy logic applied to decision making in wireless sensor networks," in *Fuzzy Logic—Emerging Technologies and Applications*, pp. 221–240.

[14] A. M. Ortiz, F. Royo, T. Olivares, J. C. Castillo, L. Orozco-Barbosa, and P. J. Marron, "Fuzzy-logic based routing for dense wireless sensor networks," *Telecommunication Systems*, vol. 52, no. 4, pp. 2687–2697, 2013.

[15] S. K. Sahani and K. Kumar, "Multi-routing in wireless sensor networks using an ant colony optimization (ACO)," *International Journal of Computer Networking,Wireless and Mobile Communications*, vol. 3, no. 3, pp. 87–98, 2013.

[16] J. Suhonen, M. Kuorilehto, M. Hännikäinen, and T. D. Hämäläinen, "Cost-aware dynamic routing protocol for wireless sensor networks—design and prototype experiments," in *Proceedings of the IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '06)*, pp. 1–5, Helsinki, Finland, September 2006.

[17] S. Okdem and D. Karaboga, "Routing in wireless sensor networks using an ant colony optimization (ACO) router chip," *Sensors*, vol. 9, no. 2, pp. 909–921, 2009.

[18] B. Kadri, M. Feham, and A. Mhammed, "Efficient and secured ant routing algorithm for wireless sensor networks," *International Journal of Network Security*, vol. 16, no. 2, pp. 149–156, 2014.

[19] E. Ghazizadeh, M. Zamani, J.-L. A. Manan, and M. Alizadeh, "Trusted computing strengthens cloud authentication," *The Scientific World Journal*, vol. 2014, Article ID 260187, 17 pages, 2014.

[20] M. Alizadeh, W. H. Hassan, M. Zamani, S. Karamizadeh, and E. Ghazizadeh, "Implementation and evaluation of lightweight encryption algorithms suitable for RFID," *Journal of Next Generation Information Technology*, vol. 4, no. 1, pp. 65–77, 2013.

[21] D. Ma, J. Ma, L. Cheng, and P. Xu, "An adaptive virtual area partition clustering routing protocol using ant colony optimization for wireless sensor networks," in *Advances in Wireless Sensor Networks*, pp. 23–30, Springer, Berlin, Germany, 2014.

[22] M. Jafari and H. Khotanlou, "A routing algorithm based an ant colony, local search and Fuzzy inference to improve energy consumption in wireless sensor networks," *International Journal of Electrical and Computer Engineering*, vol. 3, pp. 640–650, 2013.

[23] A. Chakraborty, S. Ganguly, M. K. Naskar, and A. Karmakar, "A trust based congestion aware hybrid ant colonyoptimization algorithm for energy efficient routing in wireless sensor networks (TC-ACO)," http://arxiv.org/abs/1312.4077.

[24] L. H. A. Correia, D. F. Macedo, A. L. dos Santos, A. L. Loureiro, and J. M. S. Nogueira, "Transmission power control techniques in ad hoc networks," in *Guide to Wireless Sensor Networks*, pp. 469–489, Springer, Berlin, Germany, 2009.

[25] E. Amiri, H. Keshavarz, A. S. Fahleyani, H. Moradzadeh, and S. Komaki, "New algorithm for leader election in distributed WSN with software agents," in *Proceedings of the IEEE International Conference on Space Science and Communication (IconSpace '13)*, pp. 290–295, Melaka, Malaysian, July 2013.

[26] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[27] S. Abolfazli, Z. Sanaei, M. Alizadeh, A. Gani, and F. Xia, "An experimental analysis on cloud-based mobile augmentation in mobile cloud computing," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 146–154, 2014.

[28] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*, John Wiley & Sons, New York, NY, USA, 2009.

[29] C. Chiasserini and M. Garetto, "An analytical model for wireless sensor networks with sleeping nodes," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1706–1718, 2006.

[30] H. Li, C. Yi, and Y. Li, "Battery-friendly packet transmission algorithms for wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3548–3557, 2013.

[31] A. Kailas, D. Brunelli, and M. A. Weitnauer, "Comparison of energy update models for wireless sensor nodes with supercapacitors," in *Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems (ENSSys '13)*, article 2, 2013.

[32] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[33] I. D. Chakeres and E. M. Belding-Royer, "AODV routing protocol implementation design," in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops*, pp. 698–703, March 2004.

[34] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, New Orleans, Louisiana, February 1999.

[35] X.-H. Lin, Y.-K. Kwok, and H. Wang, "Energy-efficient resource management techniques in wireless sensor networks," in *Guide to Wireless Sensor Networks*, S. C. Misra, I. Woungang, and S. Misra, Eds., pp. 439–468, Springer, London, UK, 2009.

[36] D. Cheng, Y. Xun, T. Zhou, and W. Li, "An energy aware ant colony algorithm for the routing of wireless sensor networks," in *Intelligent Computing and Information Science*, Communications in Computer and Information Science, pp. 395–401, Springer, Berlin, Germany, 2011.

[37] E. Amiri, A. Harounabadi, and S. Mirabedini, "Nodes clustering using Fuzzy logic to optimize energy consumption in Mobile

Ad hoc Networks (MANET)," *Management Science Letters*, vol. 2, no. 8, pp. 3031–3040, 2012.

[38] J.-S. R. Jang and C.-T. S. Sun, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, New York, NY, USA, 1996.

[39] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[40] K. Fall and K. Varadhan, *The Ns Manual: The VINT Project*, University of California, Berkeley, Berkeley, Calif, USA, 2001.