Energy-efficient Routing Protocols in Heterogeneous Wireless Sensor Networks

By

Mei Wu

Thesis directed by:

Dr. Martin Collier

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Oct 2012

Faculty of Engineering & Computing

Dublin City University

ABSTRACT

Sensor networks feature low-cost sensor devices with wireless network capability, limited transmit power, resource constraints and limited battery energy. The usage of cheap and tiny wireless sensors will allow very large networks to be deployed at a feasible cost to provide a bridge between information systems and the physical world. Such large-scale deployments will require routing protocols that scale to large network sizes in an energy-efficient way.

This thesis addresses the design of such network routing methods. A classification of existing routing protocols and the key factors in their design (i.e., hardware, topology, applications) provides the motivation for the new three-tier architecture for heterogeneous networks built upon a generic software framework (GSF). A range of new routing algorithms have hence been developed with the design goals of scalability and energy-efficient performance of network protocols. They are respectively TinyReg - a routing algorithm based on regular-graph theory, TSEP - topological stable election protocol, and GAAC - an evolutionary algorithm based on genetic algorithms and ant colony algorithms. The design principle of our routing algorithms is that shortening the distance between the cluster-heads and the sink in the network, will minimise energy consumption in order to extend the network lifetime, will achieve energy efficiency. Their performance has been evaluated by simulation in an extensive range of scenarios, and compared to existing algorithms. It is shown that the newly proposed algorithms allow long-term continuous data collection in large networks, offering greater network longevity than existing solutions. These results confirm the validity of the GSF as an architectural approach to the deployment of large wireless sensor networks.

Acknowledgements

I would like to present my biggest thanks to my supervisor, Martin Collier, for his guidance, patience and expertise.

I am also grateful to a number of people who have contributed, directly or indirectly, to this work: Chanle Wu, Gang Ye from Computing School at Wuhan University; friends and previous colleagues in Dublin City University: Piotr, Szymon, Leonardo, Anna, Huanhuan; my colleagues from Dublin City University and especially from the Switching and Systems Laboratory and Network Processing Group; the China Education Council for funding my Ph.D., and the Dublin City University for sponsoring my tuition.

Finally, I especially thank my dear parents and brother who have always supported and encouraged me to work hard and overcome the stress experienced during my Ph.D. life.

LIST OF PUBLICATIONS

- MEI WU AND MARTIN COLLIER, *Combinatorial cluster-based evolutionary route selection in multi-hop heterogeneous sensor networks*, in preparation, 2012.
- MEI WU AND MARTIN COLLIER, *An evolutionary routing algorithm for maximizing the lifetime of heterogeneous wireless sensor networks*, in preparation, 2012.
- MEI WU AND MARTIN COLLIER, *Extending the Lifetime of Heterogeneous Sensor Networks using a Two-level Topology*, Proceedings of the 11th IEEE International Conference on Computer and Information Technology (CIT), Pafos, Cyprus, Aug, 2011.
- MEI WU AND MARTIN COLLIER, *TSEP: A Localised Algorithm for Extending the Lifetime of Sensor Networks*, Proceedings of the IEEE Wireless Advanced (WiAD), London, UK, June, 2011.
- MEI WU AND MARTIN COLLIER, *A Routing Algorithm for Large-scale Sensing Using Wireless Sensor Networks*, Proceedings of the China-Ireland International Conference on Information and Communications Technologies (CI-ICT), Wuhan, China, Oct, 2010.
- MEI WU, CHANLE WU AND MARTIN COLLIER, *A Three-tier Structured Model of Overlay Networks*, Proceedings of the International Conference on Knowledge and Systems Engineering (KSE), Hanoi, Vietnam, Oct, 2009.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks: a review

Sensor networks can improve our lives in many ways [1]. Wireless sensor networks (WSNs) are applied across a broad range of application domains. A sensor network is an infrastructure that consists of sensing, computing and communication elements. An example of a wireless sensor network is shown in Figure 1.1. The network gives an administrator the ability to instrument, observe and react to events in a specified environment. The main object of wireless sensor networks is to reliably detect and estimate event features from the collective information provided by sensor nodes.

1.2 Motivation

A great deal of research focuses on the homogeneous environment [2]. In homogeneous networks, all node play the same role. I work on heterogeneous wireless sensor networks. As the range of applications for wireless sensor networks grows, and as the available hardware diversifies, the prevalence of heterogeneous

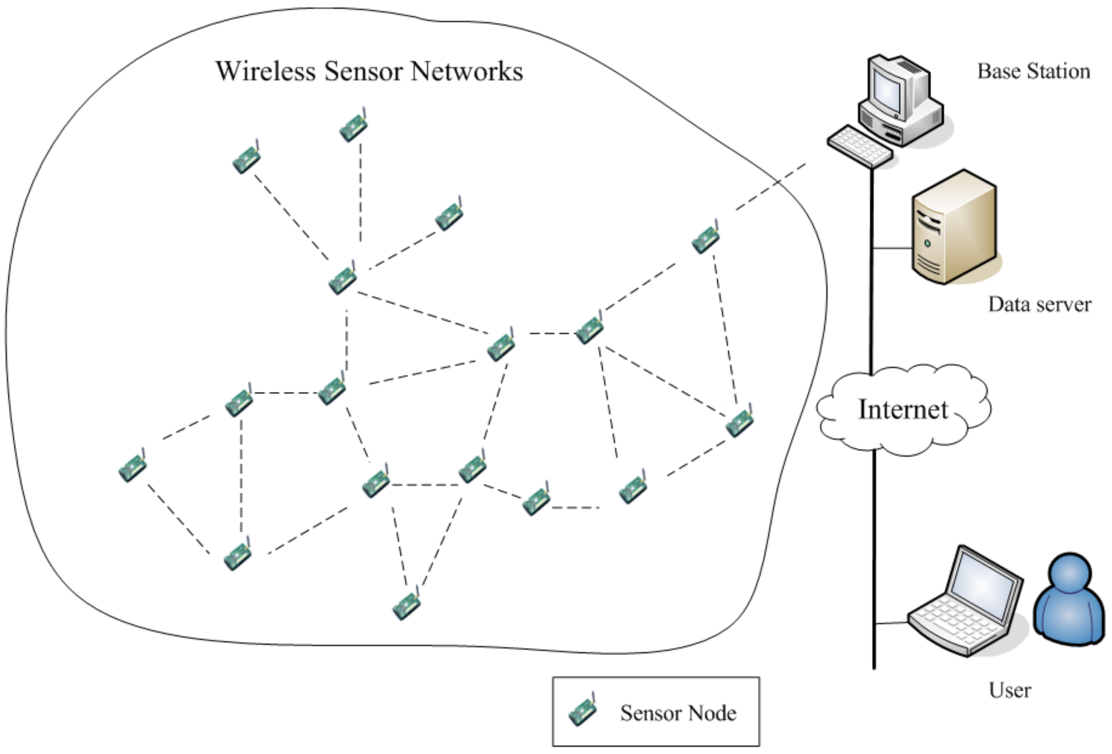


Figure 1.1: *A wireless sensor network scenario*

wireless sensor networks will increase. A heterogeneous wireless sensor network is one where the constituent motes are not all of the same hardware design, and may not execute the same code, or perform the same functions. In particular, various sensors may not have the same maximum battery capacity.

Energy saving always is a key concern in sensor networks. Because some of environments might be very dangerous, for instance, forest and building fire, volcanic mountain and underwater. Nobody would like to recharge or change batteries of sensor nodes in the network. Thus, the most challenging design issue in sensor networks is limited and non-renewable energy provision. It is desirable to develop energy-efficient processing techniques that minimise power requirements across all levels of the protocol stack and minimise the amount of message passing for network control and coordination.

Heterogeneity of the network, the redundancy of the transmitted data, and the large number of nodes may raise many challenges for the design of routing protocols in sensor networks. Meanwhile, owing to dynamic process environments and the inherent limitation of various hardware and software resources, no single topology will always be best for all applications.

Since above differences, many new algorithms have been proposed for the routing problem in wireless sensor networks.

To optimise energy consumption of routing protocols in wireless sensor networks, data aggregation and in-network processing, clustering technology, genetic algorithms and ant colony algorithms are employed in routing techniques proposed in the literature. Such techniques offer various possibilities for routing optimisation but also cause various problems.

1.3 Thesis contribution

This thesis contains a survey of the state of the art in routing protocols for wireless sensor networks. A detailed description is given of how to provide energy-efficient routing solutions for heterogeneous sensor networks. A software framework (GSF) is presented for the development of new routing algorithms. Three novel routing protocols have been developed using this framework.

They are respectively TinyReg, TSEP and GAAC, with the following design principles:

- TinyReg: a routing algorithm based on regular-graph theory;
- TSEP: a topological stable election protocol;
- GAAC: an evolutionary routing algorithm based on genetic algorithms and ant colony algorithms.

The novel aspects of these algorithms are:

- TinyReg and TSEP: my work differs from other studies in that I consider the optimal energy consumption of different types of nodes by using topology control to keep the transmission distance low. Direct transmission to the base station should be avoided.

- TSEP: this is a localised algorithm, but features more robustness than its predecessors for sensor networks of larger scale. I randomly pick nodes as cluster heads and rotate this role to evenly distribute the energy load among the sensors.
- GAAC: I am the first researcher to combine three approaches (namely clustering, genetic algorithms and ant colony algorithms) in a single routing algorithm.

These algorithms are suitable for different scenarios as follows:

- TinyReg is suitable for a sparse network.
- TSEP is scalable, and so is suited for deployment in a network consisting of a large number of nodes.
- GAAC is also scalable. It can produce a group of candidate solutions from a single execution of the algorithm, making it robust.

1.4 Thesis outline

I propose an outline of algorithms (TinyReg, TSEP, GAAC) which are shown in Figure 1.2. The remainder of this thesis is organised as follows:

Chapter 2 The design metrics in wireless sensor networks are presented in this chapter. Different applications and the classification of sensor networks are described.

Chapter 3 This chapter presents an overview on routing protocols in sensor networks. It describes the design challenges for routing protocols in sensor networks. It also describes the important routing protocols in sensor networks. Finally, it introduces the background of the computational intelligence techniques, genetic algorithms and ant colony algorithms.

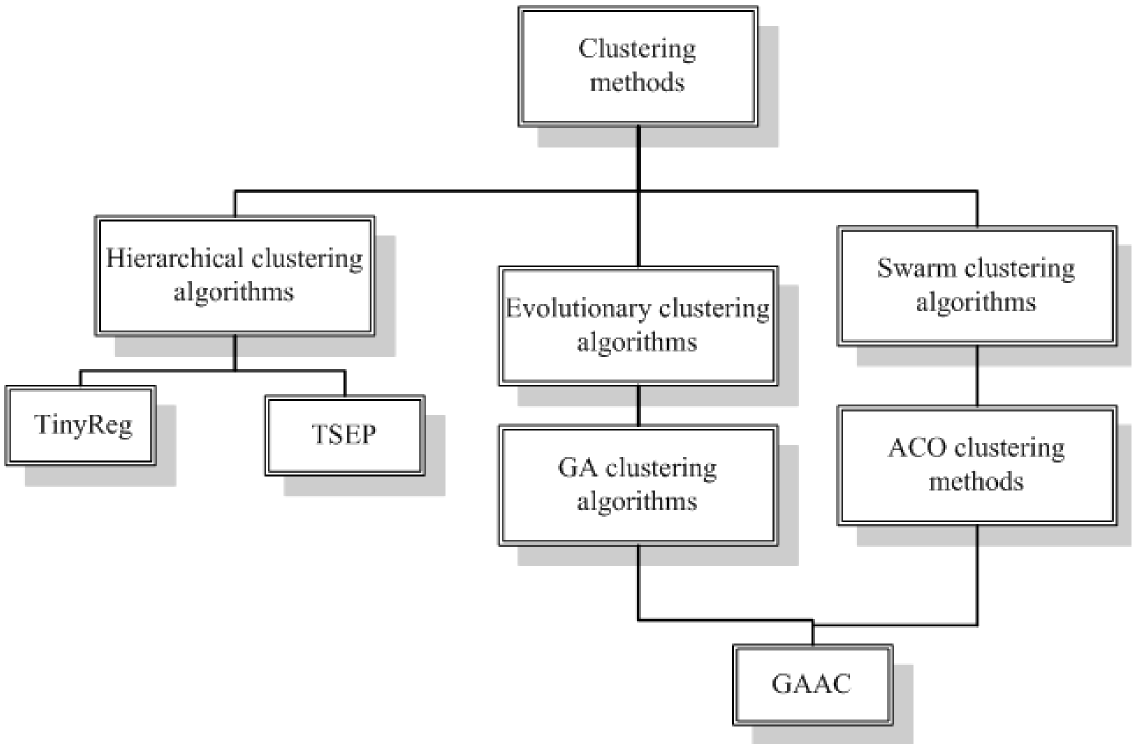

Clustering methods
Hierarchical clustering algorithms
Evolutionary clustering algorithms
Swarm clustering algorithms
TinyReg
TSEP
GA clustering algorithms
ACO clustering methods
GAAC

Figure 1.2: *Taxonomy of my approaches.*

Chapter 4 I propose a generic constrained optimised framework (called generic software framework (GSF)) for cooperating routing protocols in sensor networks. A novel three-tier hybrid model for routing protocols has also been designed in this chapter.

Chapter 5 A novel clustering routing algorithm (called TinyReg) is proposed for the heterogeneous environment in this chapter. In TinyReg, cluster-heads are selected using a function $f(E, D)$ (E is the residual energy; D is the distance of cluster heads).

Chapter 6 This chapter proposes a novel topological stable election protocol (TSEP). It utilises local energy balancing to achieve global energy balancing.

Chapter 7 In this chapter, I propose a novel energy-efficient evolutionary clustering routing algorithm (called GAAC) for solving flexible multi-constraint optimal routing problem in sensor networks. GAAC combines genetic algorithms (GA) and ant colony optimisation algorithms (ACO), speeds up the searching speed of solutions.

Chapter 8 This chapter concludes the thesis and explores suggestions for future research.

CHAPTER 2

WIRELESS SENSOR NETWORKS

2.1 WSN vs ad-hoc networks

The wireless sensor network belongs to the class of ad-hoc networks. Ad-hoc networks and sensor networks share some challenges such as energy constraints and routing. An ad-hoc network is a kind of general infrastructure-less, cooperation-based and opportunistic network. The simplest example of an ad-hoc network is a set of computers connected together via cables to form a small network. In ad-hoc networks, routing protocols are expected to implement three main functions: (1) to determine and detect network topology changes; (2) to maintain network connectivity; (3) to calculate and find proper routes.

Sensor networks feature some traffic patterns which are different from ad-hoc networks. The main type of traffic pattern is many-to-one where all nodes report to a single base station in sensor networks. This traffic pattern is less prevalent in ad-hoc networks. On the other hand, sensor networks are less likely to consist of mobile nodes, with different lifetime requirements and energy constraints [3].

The routing protocols for wireless sensor networks address the following requirements:

- A sensor network is application-specific [4]. The requirements of the network change with the target application.
- It is desirable to aggregate similar data from adjacent nodes before forwarding [5].
- Routing to and from specific nodes typically is not required [6].
- Sensor networks are "data centric" and data is typically requested based on certain attributes, e.g., temperature $\geq 30°C$ [7]. In ad-hoc networks, data is requested from a specific node.

2.2 Technical challenges of WSNs

The following key factors are encountered in designing wireless sensor networks:

- **Hardware.** Sensor network hardware platforms include the use of Micro Electro-Mechanical Systems (MEMS) sensor technology, digital circuit design, system integration for low power consumption and a low-power sophisticated radio frequency (RF) front-end and associated control circuitry [8] [9]. The design of a sensor node requires a combination of micro-sensor technology, low power signal processing, lower power computation, and low cost wireless networking capability. Figure 2.1 shows one example of the architecture of a sensor node.

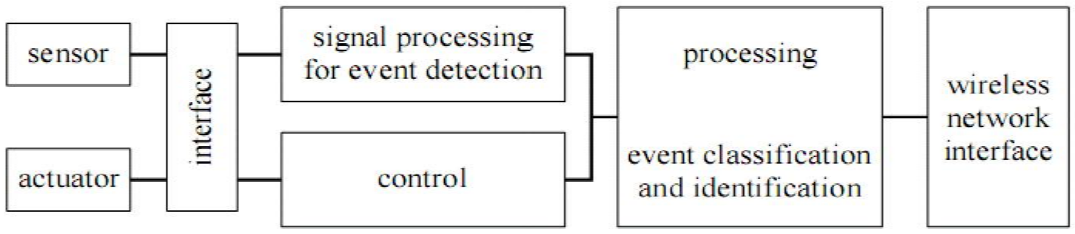

sensor
actuator
interface
signal processing for event detection
control
processing
event classification and identification
wireless network interface

Figure 2.1: *The architecture of a sensor node*

- **Wireless networking.** The challenge of designing routing protocols is to provide a robust and energy-efficient communication mechanism meeting application requirements in a wireless sensor network. Wireless networking includes the design of physical layer methods (e.g., modulation, source and channel coding, channel access methods), routing issues and mobility management issues.

- **Applications.** Wireless sensor network applications require the efficient extraction, manipulation, transport and representation of information. The information is derived from sensor data. Different systems have various functional components (e.g., detection and data collection components, signal processing components, data fusion and notification components).

2.3 Design metrics

There are a number of design metrics that determine the performance of a sensor network. These metrics include energy efficiency, latency, accuracy, fault tolerance and scalability.

- **Energy efficiency.** In many scenarios, nodes have to rely on a limited supply of energy (e.g., batteries). When sensors are battery operated, it is usually not practicable to simultaneously replace or recharge these energy sources in the field. It is wise to manage energy to extend the lifetime of the network [10].

- **Latency.** Many sensor applications (e.g., multimedia networks) require delay-guaranteed service. In these applications, sensed data must be delivered to the user within a certain delay [11].

- **Accuracy.** Any sensor application needs obtain accurate information. The accuracy of information can be improved through joint detection and estimation of multiple sensors [11].

- **Fault tolerance.** When nodes or routing links failures happen, protocols must provide a wide variety of robustness guaranteed network service. The feasible methods are through redundancy and collaborative processing and communication [11][12].

- **Scalability.** Scalable routing algorithms can operate efficiently in a wide range sensor network, which contains thousands or hundreds of thousands of nodes. Network performance must not significantly degrade as the network size or node density increases. How to design such routing protocols is very important to the future of sensor networks [11][12].

As shown above, the design of a sensor network consists of the resolution of numerous trade-offs of between energy consumption, delay and throughput, etc.

2.4 Application fields of WSNs

This section briefly describes some applications for wireless sensor networks. Since the number of fields of application is growing rapidly, it is very difficult to compile an exhaustive list of sensor network applications. The range of applications scales from habitat monitoring and home automation to traffic control and health care, from civil engineering to military war fields. Some emerging application scenarios include home monitoring and water monitoring.

2.4.1 Application examples

- **Home, civil and environment engineering applications.** Home control applications provide flexible management of lighting, heating, and cooling systems from anywhere in the home. Civil applications can enable the extension and upgrading of building infrastructure with minimal effort. Sensor networks can be deployed in remote areas to monitor environmental conditions such as microclimate changes, volcanic and seismic activity. They can also monitor wild animals in their natural habitat [1].

- **Industrial applications.** Specific applications for industrial and commercial spaces include monitoring of warehouses, fleet management, factories, assembly lines, workflow, inventory and materials processing systems (chemical, cooling, gas flow) [13].

- **Medical applications.** One of the most important and rapidly growing application areas is an application in the medical field, e.g., pre-hospital and in-hospital emergency care, disaster response and stroke patient rehabilitation. Patient and doctor tracking systems permit home monitoring for chronic and elderly patients and provide long-term care facilitation and trend analysis [14].

2.4.2 Type of applications

Many of these applications share some basic characteristics. The interaction patterns between sources and sinks shares some typical patterns as follows [15]:

- **Event detection.** A specified event can be detected by sensor nodes, such as forest fires, grass fires, volcanic eruptions, etc. When several different events occur, the events need to be classified.

- **Periodic measurements.** Sensors can be tasked with periodically reporting measured values. These event reports can be triggered by detected events.

- **Function approximation and edge detection.** When a physical value such as temperature changes from one place to another, it can be regarded as a function of location. To approximate this function, the network uses a limited number of samples taken at each sensor node. The edge detection function is to find "edges" or to it only sends a message when the sampling value jumps across the boundary of the space-time region.

- **Tracking.** When the source of an event is mobile, the sensors can report the event source's position to the sink, potentially with estimates about speed and direction. For example, an intruder might be tracked in surveillance scenarios.

Sensor network applications feature a diversity of deployment options, maintenance options and options for energy supply [15]. Applications range from fixed deployment of sensor nodes (for example, in machinery maintenance applications) to random deployment by dropping a large number of nodes from an aircraft over a forest fire. In mobile applications, sensor nodes can move themselves. In some scenarios, there is no need or possibility to maintain sensors as the networks are only deployed in a short-term manner with a maximum mission time, such as in disaster recovery operations. In other cases, applications need to periodically perform maintenance on sensors.

2.5 Classification of WSNs

As sensor networks include a variety of applications, environment and technical requirements may greatly differ in different applications. Therefore, designers should build application-oriented sensor networks. Meanwhile, different sensor networks have some common properties [16][17][18][19].

2.5.1 Single-hop networks vs multi-hop networks

Sensor networks can be classified as single-hop or multi-hop networks depending on the number of hops between sensor nodes and the base station [17]. In a single-hop sensor system, all sensor nodes transmit the data directly to the base station without using intermediate nodes; in a multi-hop sensor network, some nodes deliver their data to the base station via intermediate nodes.

2.5.2 Aggregating networks vs nonaggregable networks

Based on the sensor node density and data dependency, sensor networks can be classified as aggregating or non-aggregable [18]. The aggregating scheme is suitable for large-scale systems that have massive node counts, higher node density and limited capacity. In non-aggregable systems, all data from each node will be sent directly to the destination. As a result, the total traffic load of the system will increase rapidly with the enlargement of the network size.

2.5.3 Deterministic networks vs dynamic networks

According to the spatial distribution of sensor nodes, sensor networks can be classified as deterministic or dynamic [18]. The control of dynamic schemes is more complex and its implementation is scalable and flexible compared to deterministic schemes. When the positions of the sensor node are obtained or preplanned, deterministic schemes can be used. Otherwise, the network needs to use dynamic control algorithms.

2.5.4 Self-configurable vs non-self-configurable networks

Based on the control scheme, sensor networks can be classified as self-configurable or non-self-configurable [18]. The self-configurable scheme fits better in large-scale systems and can perform complicated monitoring tasks and data dissemination. The non-self-configurable scheme relies on a central controller to command the sensor nodes. It may be used in small-scale networks.

2.5.5 Proactive networks vs reactive networks

On the basis of the function mode of the network and the type of target applications, sensor networks are classified as proactive networks or reactive networks [16]. Proactive networks periodically switch on the sensors and transmitters to sense the environment and to transmit the data of interest. They provide a snapshot of the relevant parameters at regular intervals. They are suited to applications that require periodic data monitoring, such as collecting data about temperature change over a particular area. The nodes in reactive networks react immediately to sudden and drastic changes in the value of a sensed attribute. They are well suitable for time-critical applications, e.g., intrusion detection, explosion detection, etc.

2.6 Summary

This section gives a general discussion on wireless sensor networks. Future chapters will focus on this network and propose a framework and an architecture based on the above characteristics of sensor networks.

CHAPTER 3

ROUTING IN WIRELESS SENSOR NETWORKS

In this chapter, I first analyse the primary design goals and challenges for routing protocols in wireless sensor networks (WSNs). Then I discuss various routing strategies and describe some important routing protocols in sensor networks.

3.1 Routing challenges in WSNs

Routing is a key element for sensor networks. In this section, I will describe four main challenges in designing routing protocols for WSN: scalability, heterogeneity, robustness and energy efficiency.

Heterogeneity problem of WSNs is discussed in [20]. Sensor networks can be classified into homogeneous sensor networks and heterogeneous sensor networks [11]. In a homogeneous network, the sensor nodes have identical capacities and functionality with respect to the various aspects of sensing, communication, and resource constraints [11]. A heterogeneous WSN is one where the constituent motes are not all of the same hardware design, and may not execute the same code, or perform the same functions. In particular, various sensors may not have the same maximum battery capacity. Since the dominant process consuming energy in a mote is communication [21], the upper bound on the lifetime of sensor networks is constrained by the communication costs and battery capacity. Therefore, maximising the lifetime of a heterogeneous WSN requires the network routing protocol to take account of the heterogeneity of the motes.

On the other hand, to comply with the self-working paradigm, WSN routing protocols are designed to ensure devices coordinate their actions and to exploit redundancy. Redundancy of data can occur when multiple sensors generate the same data within the vicinity of a phenomenon [12].

3.1.1 Scalability / network scale

A sensor network may have thousands or hundreds of thousands of nodes which spread over a wide geographical area [22], unlike a typical ad-hoc network which has no more than a few hundred nodes. Therefore, routing must be designed to scale to support several thousands of sensor nodes in the sensor field. Hierarchy [23][24][25], clustering [26][27][28] and location-awareness [29] techniques have been proposed to cope with scalability in large sensor networks.

3.1.2 Resources constrained at each sensor node

Sensor nodes are designed with mostly simple devices of low cost according to the economy of scale. In contrast, a typical computer serves a wired internet or an ad-hoc network with unconstrained energy availability, high CPU capacity and several GB of memory capacity. Economic factors dictate that sensor networks must cover a wide spatial area with the lowest cost sensing devices. As a result, sensors tend to have limited battery energy, a relatively low performance CPU and constrained memory size. Therefore, energy consumption is a key concern in WSNs.

3.1.3 Robustness

Robustness of the network resources requires tolerance of sensor nodes that nodes may fail, lose battery power or be temporarily unable to communicate due to environmental factors [30].

3.1.4 Energy efficiency

Energy is a precious resource in wireless sensor networks. When sensors are battery-powered, sensor networks are expected to have lifetimes for several years. It is thus a crucial requirement to conserve and save the battery power in wireless devices. There are two factors which hinder the possibility to recharge the power supplies of WSN nodes.

- **A large number of nodes.** It is difficult to recharge nodes in perhaps thousands of nodes in a sensor network deployment.
- **The complexity of the application environment.** It may be dangerous or time-consuming to replace sensors that have failed due to battery energy depletion.

Therefore, one of our main optimisation goals in the design of routing methods is to maximise energy efficiency.

3.2 Important routing protocols in WSNs

In this section, I discuss some important examples of wireless routing protocols. I summarise and classify the following approaches into four main categories, flat routing protocols, hierarchical routing protocols, location-based routing protocols and QoS-based routing protocols. There are some protocols that fit under more than one category. Table 3.1 shows the classification of the protocols covered in this section.

Table 3.1: *Routing protocols in sensor networks*

Routing protocol	**Flat**	**Hierarchical**	**Location-based**	**QoS-based**
SPIN	Yes			
Directed Diffusion	Yes			
LEACH		Yes		
PEGASIS		Yes		
TEEN, APTEEN		Yes		
Younis et al.		Yes	Yes	
MECN		Yes	Yes	
SPAN		Yes	Yes	
GEAR		Yes	Yes	
GAF		Yes	Yes	
VIBE		Yes	Yes	
SAR	Yes			Yes
SPEED	Yes		Yes	Yes
MERR, AMERR		Yes		Yes

3.2.1 Flat routing protocols

In flat networks such as point-to-point networks, each node plays the same role and sensor nodes collaborate to work. In flat routing protocols, all nodes play the same important role.

I describe below some important approaches of routing histories such as SPIN [31], Directed Diffusion [30] in detail and highlight the key ideas. There are many flat routing protocols based on them.

A. **Sense Protocols for Information via Negotiation** (called SPIN) is proposed by Heinzelman *et al.* ([31]). It is a family of data dissemination protocols and designed for scenarios where each node disseminates its information to every node in the network. It assumed that all nodes in the network are potential base stations. The main idea of SPIN is using meta-data negotiation and resource adaptation to eliminate the transmission of redundant data.

Because the amount of data at each node is large in SPIN, thus, SPIN-1 ([31]) (the first of the SPIN family of protocols) assigns a high-level name to completely describe collected data. The collected data called meta-data, is a kind of small size data relative to the data itself. It works in three-stages (ADV-REQ-DATA) and the process of operation is shown in Figure 3.1. The SPIN protocol starts when a node obtains new data and sends an ADV message to its neighbours indicating that it is willing to disseminate this data. An ADV message contains meta-data. The meta-data's format is application-specific. In SPIN, the interpretation of meta-data depends on application's context. Having received an ADV message, the neighbour checks its data to see whether it has already received or requested the advertised data. If not, it responds by sending an REQ message for the missing data back to the sender. An REQ message tells the receiver that the sender is interested in the data and asks for the DATA. A DATA message contains actual sensor data with a meta-data header. The neighbour sensor node repeats this process with its neighbours. As a result, the entire sensor area will finally receive a copy of the data.

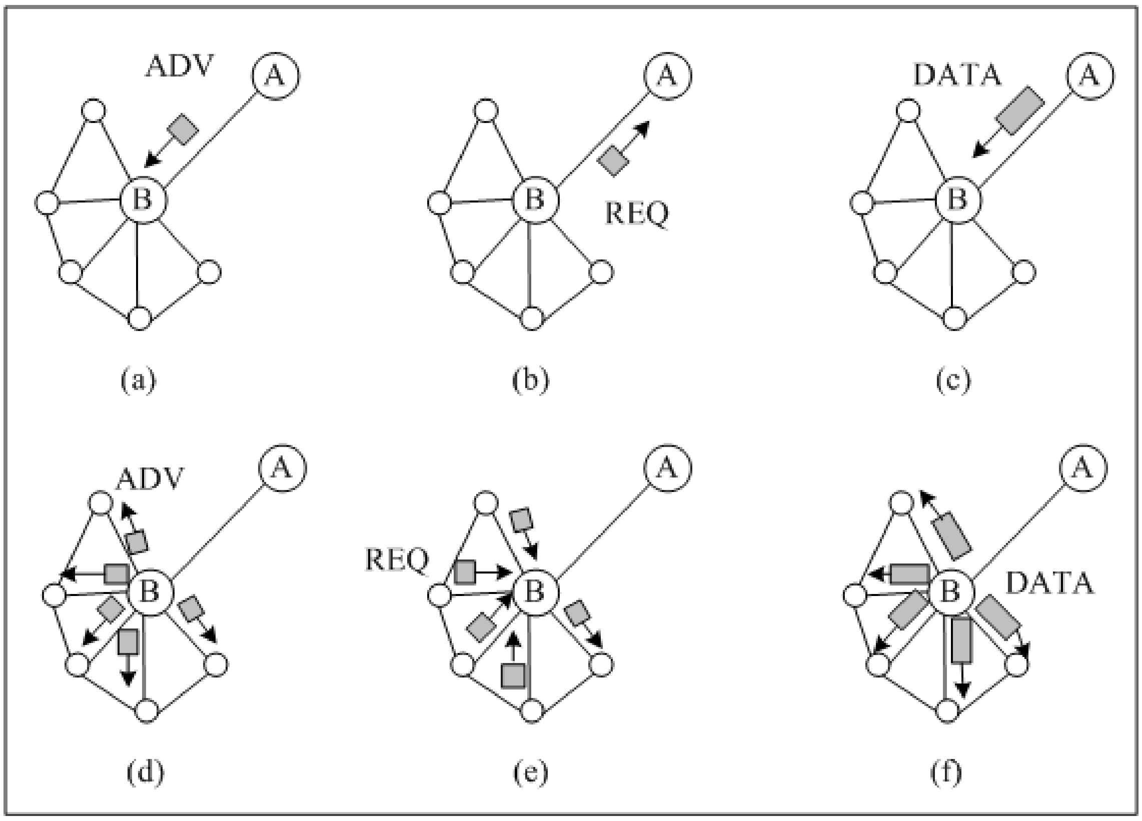


Figure 3.1: *The process of 3-stages operation of SPIN protocol. (a) Node A starts by advertising its data to node B. (b) Node B responds by sending a request to node A. (c) After receiving the requested data, (d) node B sends out advertisements to its neighbours, (e)-(f) who in turn send requests back to B. Source: [31].*

As an extension to SPIN-1, SPIN-2 ([31]) incorporates a threshold-based resource awareness mechanism and negotiation. SPIN-2 checks the current energy level of a node and if energy in the node is abundant, it communicates using the same three-stage protocols as SPIN-1. If the node energy approaches a low-energy threshold, SPIN-2 will reduce its participation in the protocol.

In SPIN protocols, no topological information is required. Each node only need know its single-hop neighbours. If the network topology changes frequently, these changes only travel one hop before the nodes can continue running the algorithm. As a result, SPIN can be run in an un-configured network with a small startup cost to determine nearest neighbours. SPIN protocols claim to be able to transmit 60% more data for a given amount of energy than the conventional flooding method. However, SPIN is not suited to an environment where the sensors require reliable delivery of data packets over regular intervals. For instance, if the nodes that are interested in the data are far away from the source node, and intermediate nodes between source and destination are not interested in that data, such data will not be delivered to the destination at all.

B. **Directed Diffusion** [30] relies on flooding to gather route information and deliver data. Directed Diffusion issues data queries on demand as the base station sends queries to the sensor nodes by flooding some tasks (as defined below) while sensors in SPIN advertise the availability of data. That is the main difference between Directed Diffusion and SPIN.

A task description announces an interest for data matching the attributes. An interest is defined using a list of attribute-value pairs (e.g., type, interval, geographical area, duration, etc). The sink broadcasts the interest through its neighbours. Each node receives the interest and maintains an interest cache with distinct items for later use. This process continues until gradients are set up from the sources back to the base station. Figure 3.2 shows the gradients established in the case where interests are flooding through a sensor field. A gradient is a reply link to a neighbour from which the interest was received. It is characterised by the data rate, duration and expiration time which are derived from the received interest's fields. When interests fit gradients, paths of information flow are formed from multiple paths. Then, the best paths are reinforced by re-sending the original interest through selected path from the sink with a smaller interval, to prevent further flooding.

In the Directed Diffusion protocol, there is no need for maintaining global network topology. Directed Diffusion is based on a query-driven data delivery model. However, it may not work efficiently for on-demand applications such as environmental monitoring where it requires continuous data delivery to the sink.

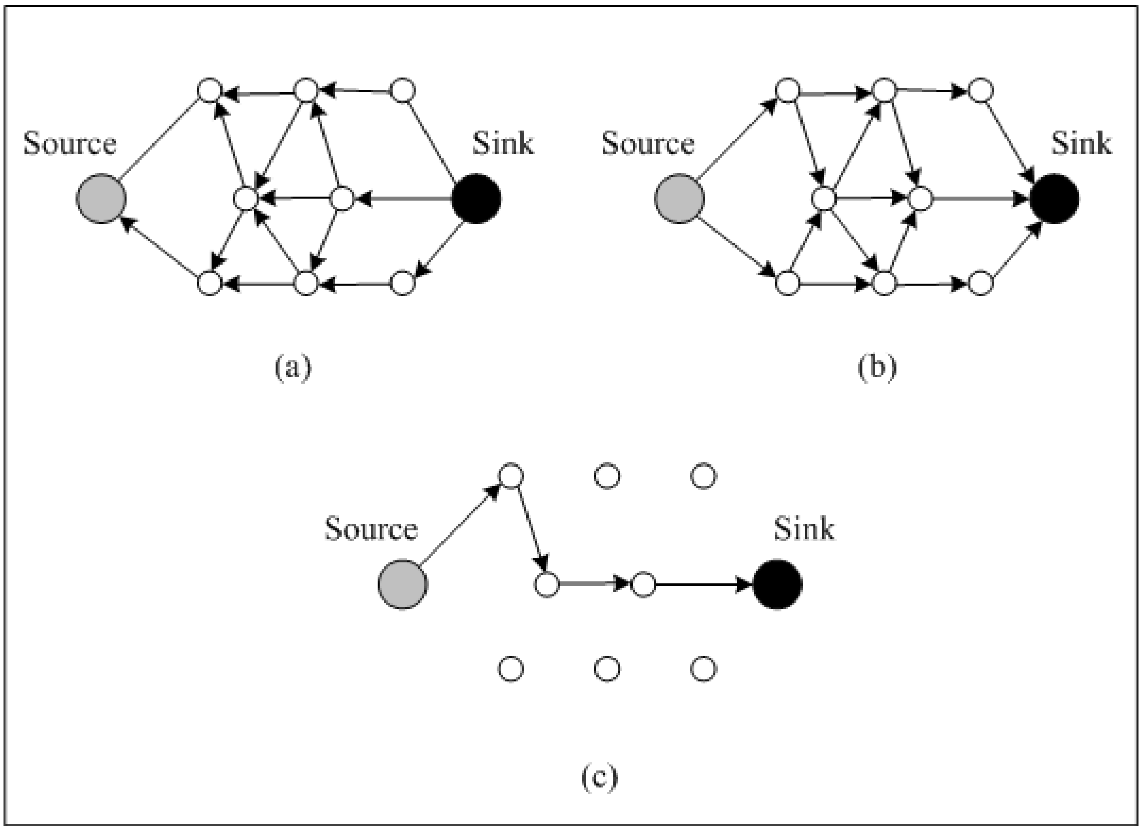


Figure 3.2: *Directed Diffusion protocol phases: (a) Propagate interest, (b) Initial gradients setup, (c) Data delivery along reinforced path. Source:[30]*

3.2.2 Hierarchical routing protocols

A hierarchical architecture typically comprises two layers of routing where one layer is used to select cluster-heads and the other is for routing [12]. Nodes can play different roles in the network; by clustering the nodes, cluster-heads can do some aggregation and reduction of data to save energy. Higher energy nodes process and send the information while lower energy nodes perform the sensing.

Hierarchical routing is an efficient way to lower energy consumption within a cluster. It decreases the number of messages transmitted to the base station. It has special advantages related to scalability and efficient communication. Hierarchical routing protocols include LEACH, PEGASIS, TEEN, APTEEN, MECN, SMECN, SOP, VGA, HPAR, HEED, CPCHSA [12] [22].

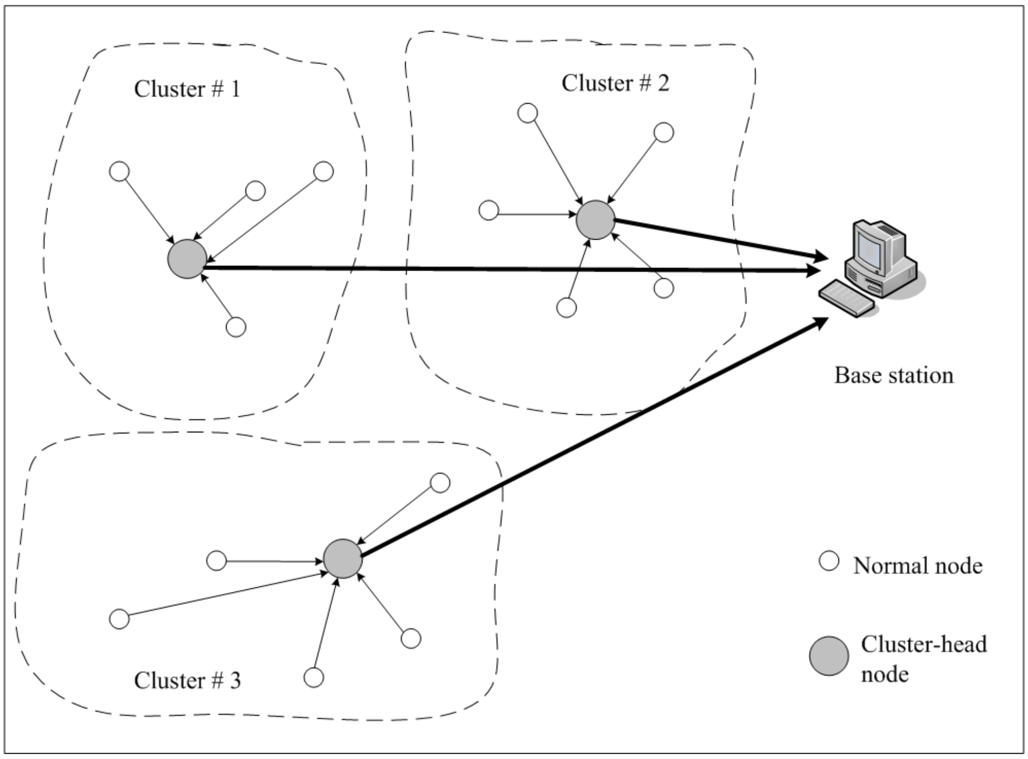


Figure 3.3: *The structure in the LEACH protocol.*

A. **Low-energy adaptive clustering hierarchy** (LEACH) [28] is designed to provide energy-optimal network organisation. In LEACH, it is assumed that all nodes are homogeneous and energy-constrained; in addition, each node is able to adjust its wireless transceiver's power consumption. It adopts a hierarchical approach and divides the network into a set of clusters. Figure 3.3 shows the two-hop clustering structure in the LEACH protocol. It is a clustering hierarchy scheme based on a Voronoi diagram. It is able to collect data from sensors and deliver it directly to the sink. In LEACH, two-hop data dissemination is used as a trade-off between the transmission and receiving energy consumptions. Data messages from each sensor node are first transmitted to a local cluster-head, and then forwarded to the base station.

There are two phases in the forming of the network: the setup phase and the steady-state phase. In the first phase, cluster-heads are selected and clusters are organised. All the data are local to the cluster. LEACH randomly selects a few sensor nodes as cluster-heads (CH) and rotates this role to evenly distribute the energy load among the sensors in the network. The node i becomes a cluster-head for the current round if the random number (chosen by the node and the value is between 0 and 1) is less than the following threshold:

$$T(i) = \begin{cases} \frac{p}{1-p\times(r \mod \frac{1}{p})}, & \text{if } i \in G \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

where p is the desired percentage of cluster-heads (e.g. 0.05), r is the current round, and G is the set of nodes that have not been cluster-heads in the last $1/p$ rounds. These cluster-heads advertise themselves to guide the clustering of the sensor networks. Then a sensor node joins the cluster from which the strongest advertisement signal is received. The non-cluster-head nodes inform the selected cluster-heads that they would like to be included in the cluster. Then the cluster-head creates a TDMA schedule and assigns each node a time slot to transmit. The schedule is broadcast to all the nodes in the cluster.

During the steady state phase, the sensor nodes can begin sensing and transmitting data to cluster-heads. After certain period, determined a priori, the network goes back into the setup phase again and enters another round of selecting new cluster-heads. The duration of the steady state phase is longer than the duration of the setup phase in order to minimise overhead.

LEACH reduces communication energy by as much as 8 times compared with direct transmission and minimum transmission-energy routing [28]. LEACH is able to increase the network lifetime and requires no global knowledge of the network. However, it assumes that all nodes can transmit with enough power to reach the base station if needed and it uses single-hop routing, where each node transmits directly to the cluster-head and the base station. Hence, it is not applicable to networks deployed in large scale sensor networks.

B. **Power-efficient gathering in sensor information systems (PEGASIS) and**

Hierarchical-PEGASIS [32][33] are enhancements of the LEACH protocol. The key idea in PEGASIS is to form a chain among the sensor nodes so that each node will receive from and transmit to a close neighbour in a static and homogenous network. Cluster-heads take turns transmitting to the base station. Thus the average energy spent by each node per round is reduced.

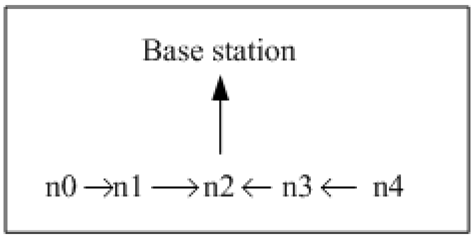


Figure 3.4: *Date gathering in a chain of PEGASIS. Source: [32]*

It assumes that all nodes have global knowledge of the network. It employs a greedy algorithm to construct the chain. The process of construction is shown in Figure 3.4. Node n2 is the leader which passes its token along the chain to node n1 and node n3 respectively. Node n0 passes its data to node n1, n1 aggregates node n0's data with its own, then n1 transmits to the leader n2. Meanwhile, node n4 transmits its data to node n3. Node n3 aggregates node n4's data with its own and then transmits to node n2. Node n2 aggregates its data with its two neighbours' data and transmits one message to the base station. Whenever a node dies, the chain will be reconstructed and the threshold, which is adaptive to the remaining energy levels in nodes, can be changed to determine which nodes can be the leader. Nodes take turns being the leader.

Unlike LEACH, a chain for multi-hop routing is formed. Simulations show that PEGASIS performs better than LEACH by about 100% to 300% when 1%, 20%, 50% and 100% of nodes die for different network sizes and topologies [32]. Although PEGASIS avoids the clustering overhead of the LEACH protocol, it introduces considerable overhead and delay for distant nodes on the chain. It requires dynamic topology adjustment, since a node on the chain needs to know the energy status of its neighbours in order to know where to route its data. In addition, the single leader can become a bottleneck.

Hierarchical-PEGASIS [32] is an extension to PEGASIS with the objective of decreasing the delay in transmitting packets to the base station. To avoid collisions and signal interference among the sensors, Hierarchical-PEGASIS improves simultaneous transmissions of data messages through incorporating signal coding CDMA (code division multiple access) and spatial transmissions.

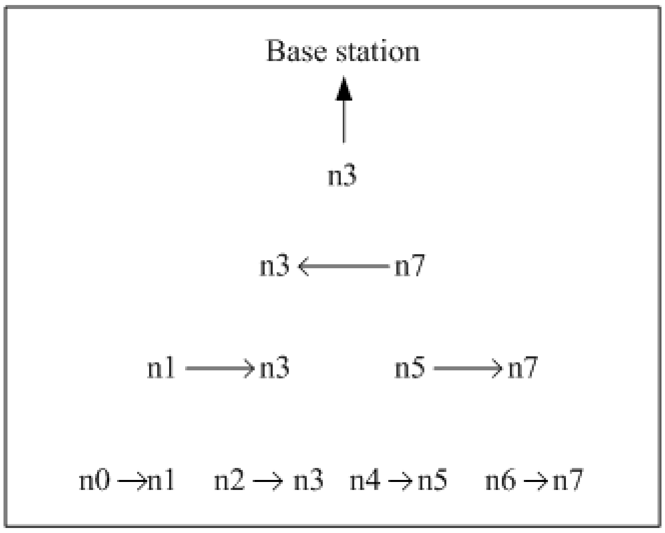


Figure 3.5: *Date gathering in a chain of Hierarchical-PEGASIS. Source: [33].*

The process of data gathering in a chain based binary scheme is shown in

Figure 3.5. The chain of nodes with CDMA capability forms a tree-like hierarchy; each node in a level of the hierarchy transmits data to the node in the upper level. The tree is balanced and the delay would be in $O(log^N)$ where N is the number of nodes [32]. Nodes that are receiving at each level rise to the next level in the hierarchy. Node n3 is the leader in position 3 (counting from 0) on the chain for round 3. Each node in an even position will aggregate its data with its received data and sends to its right neighbour. At the next level, n3 is still in an odd position (1). So node n1 sends its data to n3 and n5 sends its data to n7. At the third level, node n3 is in an even position but n7 will aggregate its data and transmit to n3. Finally, node n3 will combine its current data with the data received and transmit the aggregated data as a message to the base station. The hierarchical-PEGASIS protocol performs better than the PEGASIS by a factor of about 60 in terms of energy-delay product.

C. **Threshold sensitive energy efficient sensor network protocol (TEEN)/ the adaptive threshold sensitive energy efficient sensor network protocol (APTEEN)** [16][34] are two proposals for time-critical applications. In time-critical applications, it is important to respond to sudden changes in the sensed attributes such as temperature. TEEN [16] is a hierarchical approach using a data-centric mechanism. The TEEN architecture is based on a hierarchical grouping where close nodes form clusters, this clustering process is repeated in the second level until the base station is reached.

The hierarchical model is shown in Figure 3.6. When all clusters are formed, the cluster-heads broadcasts two thresholds to the nodes. They are a hard threshold and a soft threshold for sensed attributes. The hard threshold is the minimum possible value of an attribute to trigger a sensor node to switch on its transmitter and transmit to the cluster-head. It allows the node to transmit only when the sensed attribute is in the range of interest. Thus it reduces the number of transmissions significantly. The soft threshold further reduces the number of transmissions that might occur when there is little or no change in the sensed attribute. Thus, the user can control the tradeoff between energy efficiency and data accu-

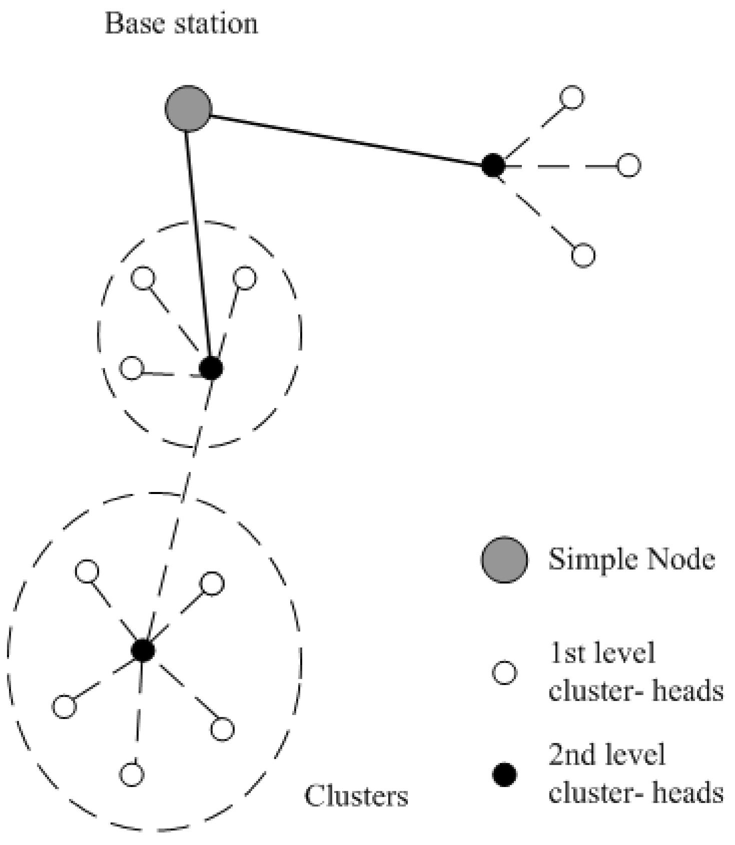


Figure 3.6: *Architecture of TEEN and APTEEN. Source: [16][34].*

racy.

However, TEEN does not support applications where periodic reports are needed. Hence the user may not get any data at all if the thresholds are not reached.

APTEEN [34] is an extension to TEEN. It is a hybrid protocol that changes the periodicity or threshold values used on the TEEN according to user needs and the application type. APTEEN supports three different query types: historical queries (by analysing past data values), one-time queries (by taking a snapshot view of the network) and persistent queries to monitor an event for a period of time. It uses the same architecture as TEEN. The time line is shown in Figure 3.7.

Simulations of TEEN and APTEEN show that they outperform LEACH [16][34]. TEEN gives the best performance among TEEN, APTEEN and LEACH. The main drawbacks of the two approaches are the overhead and complexity associated

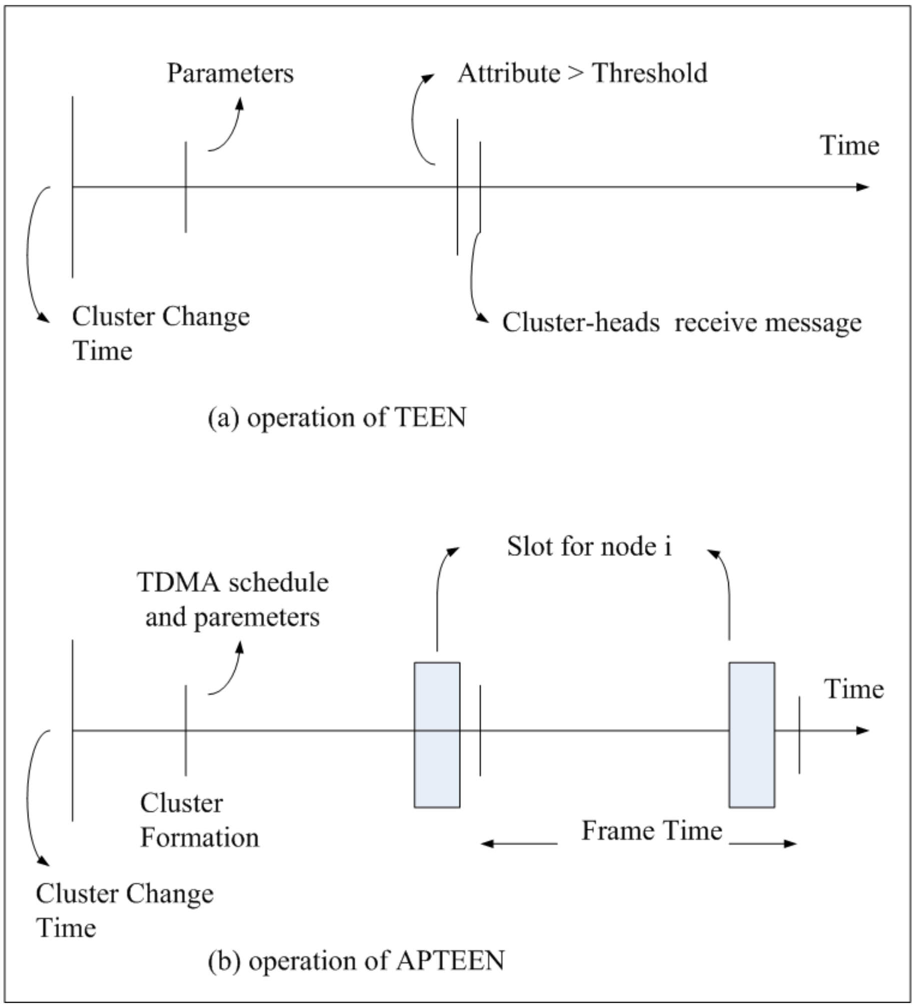


Figure 3.7: *Time line for the operation of TEEN and APTEEN: (a) TEEN – when cluster-heads are to change, new values of the hard threshold and soft threshold are broadcast; (b) APTEEN – the cluster-heads broadcast attributes, thresholds, schedule and count time. Source: [34].*

with forming clusters at multiple levels.

D. **Energy-aware routing in cluster-based sensor networks** is an approach proposed by Younis *et al.* ([35]). The main objective is extending the life of the sensors in a particular cluster for applications of sensor networks in military and disaster management applications. The main idea behind a clustering sensor network is presented in Figure 3.8. Sensors are grouped into clusters in the network. Route setup in a cluster is centralised at the gateway (namely, cluster-heads), where the gateway informs nodes in the clusters newly assigned states and how to route the data. Nodes in a cluster can be in one of four main states: sensing only, relaying only, sensing-relaying and inactive.

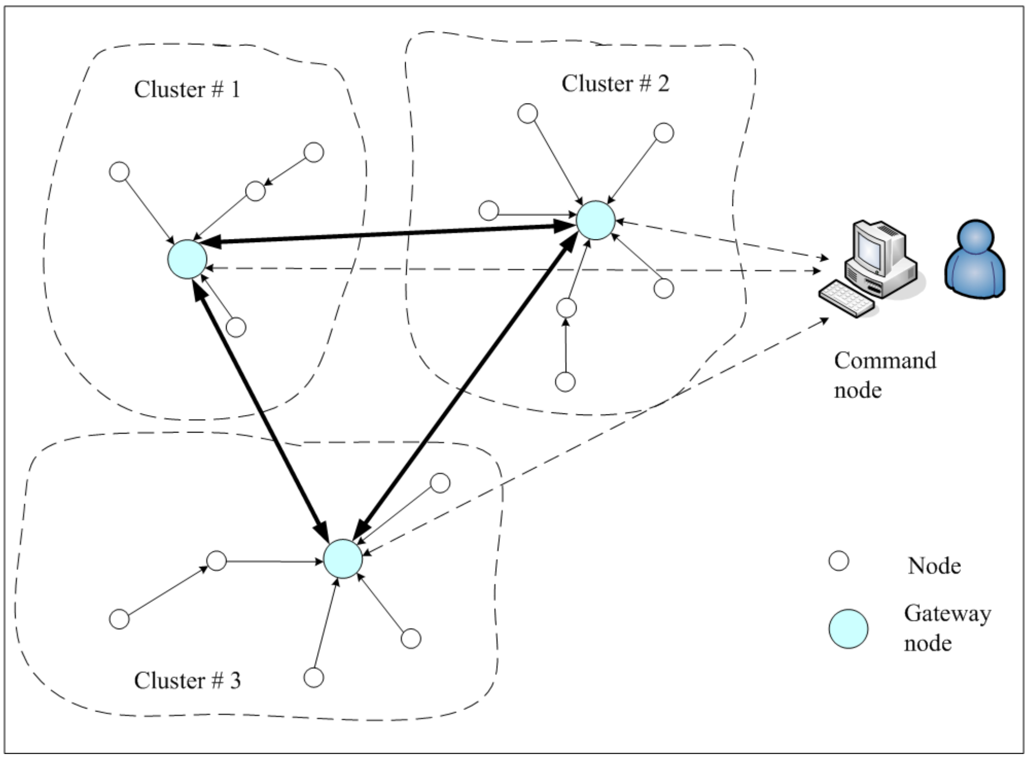


Figure 3.8: *A multi-gateway clustering sensor network. Source: [35].*

To set the routes, the gateway respectively sends sensing nodes the transmission range and relay nodes a forwarding table. The approach uses Dijkstra's algorithm for solving the shortest path (least-cost) between the nodes i and j. The cost function between the nodes i and j considers energy consumption, delay optimisation and other performance metrics. Using this cost function as the link cost, the routing algorithm finds a least-cost path between the gateway and sensor nodes for each sensing-enable node.

The experimental results have demonstrated that energy-aware routing method performs well in terms of both energy-based metrics (such as network lifetime), as well as contemporary metrics (e.g., throughput and end-to-end delay). However, in order to ensure high sensor coverage, it needs to deploy many gateways in large scale sensor networks. For large-scale network application scenarios, such limitation can increase the deployment cost of the network.

3.2.3 Location-based routing protocols

Location information can be utilised to calculate the distance between two special nodes. So sensor routing protocols can estimate the energy consumption of nodes.

In geographic routing or location-based routing, sensor nodes are addressed by means of their locations. This is based on two principal assumptions: first of all, that every node knows its own and its network neighbour's positions; secondly, that the source of message is informed about the position of the destination. There are three ways to estimate the distance between neighbour nodes: based on incoming signal strengths, exchanging relative coordinate information of neighbouring nodes, and satellite communications if nodes are equipped with a small low-power GPS receiver [29].

To save energy, some location-based schemes demand that some nodes sleep if there is no activity. In a localised manner, the issue of designing sleeping period schedules is addressed in SPAN and GAF [29][36]. Some famous location-based routing protocols include SPAN, GEAR, MFR, GEDIR, GAF. Next, I describe the details of some location-based routing protocols which are MECN, SPAN, GEAR, GAF, VCP and VIBE [37].

In localised routing algorithms [28][38][39], nodes make routing decisions solely on the basis of location of their neighbours and destination. For instance, Heinzelman *et al.* [28] uses data-aggregation and clustering methods to improve the network lifetime performance. Oteafy *et al.* [40] elects dynamically the most reliable next-hop neighbour to relay back to the sink. Bermudez *et al.* [41] presents the use of sleep-based topology control and back-up nodes to reduce the energy consumption of the individual nodes and increase the functional lifetime of the network. However, the energy consumption of cluster-heads for these protocols increases with increasing network scale, since the transmission distance from cluster-heads to the base station is increased as the scale of network increases.

Globalised strategies make routing decisions based on the global state of systems to maximise network lifetime. For these algorithms, it is necessary to explicitly transmit energy information between nodes. A greater number of sensors allows for sensing over larger geographical regions with greater accuracy [12][42]. However, it is impossible to build a global addressing scheme for the deployment of a large number of sensor nodes as the overhead of ID maintenance is high [12].

Unlike the globalised algorithm [12], there is no need to maintain global network topology in Directed Diffusion [43]. Solutions such as [43], which feature direct communication (with the base station), show serious network lifetime performance degradation when the base station is far away from the nodes. In that case, direct communication will require a large amount of transmit power from each node.

A. **Minimum energy communication network (MECN)** is proposed by Rodoplu *et al.* [44]. In MECH, it establishes a theory of minimum energy network for wireless networks by utilising low-power GPS. The main idea of MECN is to find a subnetwork that will have fewer nodes and require less power for transmission between any two particular nodes. That means the global minimum power paths are found without considering all the nodes in the network. It uses a localised search for each node considering its relay region. The protocol has two phases:

Phase 1: by using a localised search, each node picks those few links in its immediate neighbourhood as the only potential candidates. It takes the position of a two-dimensional plane to construct a sparse graph (enclosure graph). The graph consists of all the enclosures of each transmit node in the graph. The enclose graph contains globally optimal links in terms of energy consumption.

Phase 2: finding the optimal links on the enclosure graph. It introduces the distributed Bellman-Ford shortest path algorithm on the enclosure graph and uses power consumption as the cost metric [44]. The cost of node i is defined as the minimum power necessary to establish a path to the master-site for i .

In MECN, a relay region is identified for every node. The relay region consists of nodes in a surrounding area where transmitting through these nodes is more energy-efficient than direct transmission. It illustrates a typical relay region of node pair (i, r) in Figure 3.9, where MECN created the enclosure of a node i by taking the union of all relay regions that node i can reach.

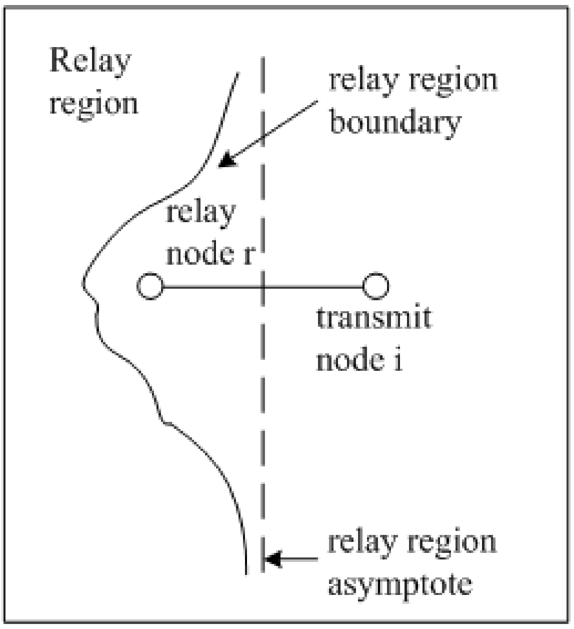


Figure 3.9: *Relay region of transmit-relay node pair (i,r) in MECN. Source: [44].*

SMECN is self-reconfiguring and can dynamically adapt to node failure or the deployment of new sensors [22]. It assumes that every node can transmit to every other node, which is not possible every time. The small MECN (SMECN) [45] is an extension to MECN and considers the obstacles between any two pair of nodes. The sub-network constructed by SMECN for minimum energy relaying is provably smaller (in terms of number of edges) than the one constructed in MECN [45]. Subgraph G′ of graph G, which represents the sensor network, minimises the energy usage satisfying the following conditions:

- The number of edges in G′ is less than in G while containing all nodes in G.
- The energy required for transmitting data from a node to all its neighbours in subgraph G.

Simulations shows that SMECN uses less energy and the maintenance costs of the links are lower than MECN. However, it introduces more overhead in the algorithm to find a sub-network with a smaller number of edges.

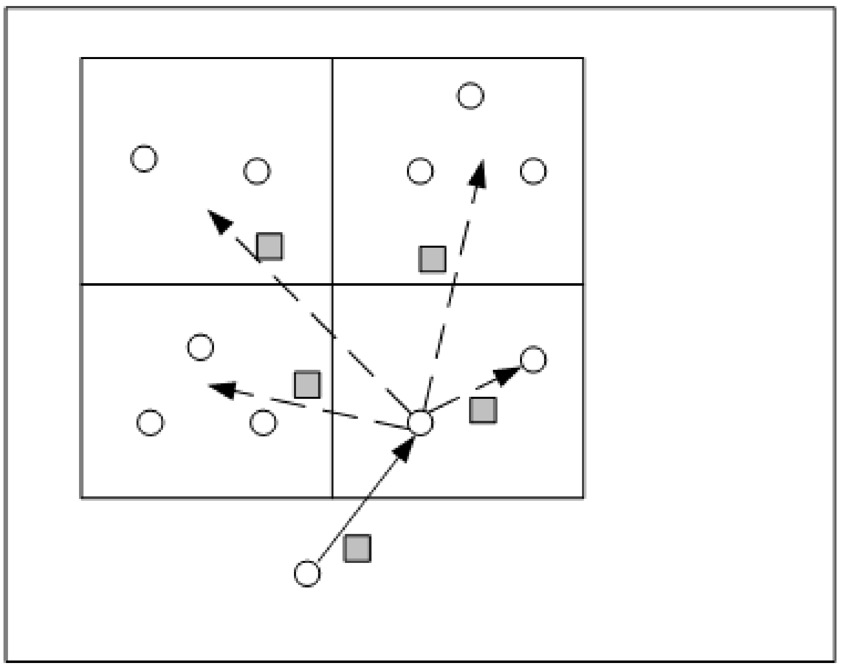


Figure 3.10: *Recursive geographic forwarding in GEAR where rectangles are data, and circles are sensor nodes. Source: [46].*

B. **Geographic and Energy Aware Routing (GEAR)** [46] uses energy-aware and geographically informed neighbour selection heuristics to route a packet toward the destination region. It uses geographic information while disseminating queries to appropriate regions. The main idea is to restrict the number of interests in Directed Diffused by considering a certain region rather than sending the interests to the whole network.

In GEAR, every node keeps an estimated cost and a learning cost of reaching the destination through its neighbours. The estimated cost is a combination of residual energy and distance to destination. The learned cost is a refinement of the estimated cost that accounts for routing around holes in the network. There are two phases in the scheme as follows.

Phase 1: forwarding the interest packets toward the target region. Upon receiving a packet, a node checks its neighbours to see if there is a node that is closer to the target region than itself. If more than one such node exists, then the nearest neighbour to the target region is selected as the next hop.

Phase 2: forwarding the interest packets within the target region. The packets can be transmitted by either recursive geographic forwarding or restricted flooding within the region. When the sensor network is sparse, restricted flooding is better than the former method; when the sensors are densely deployed, recursive geographic forwarding is more energy-efficient than restricted flooding. In Figure 3.10, the region is divided into four subregions and four copies of the packet are created. The splitting and forwarding process is repeated until each region has one and only one node.

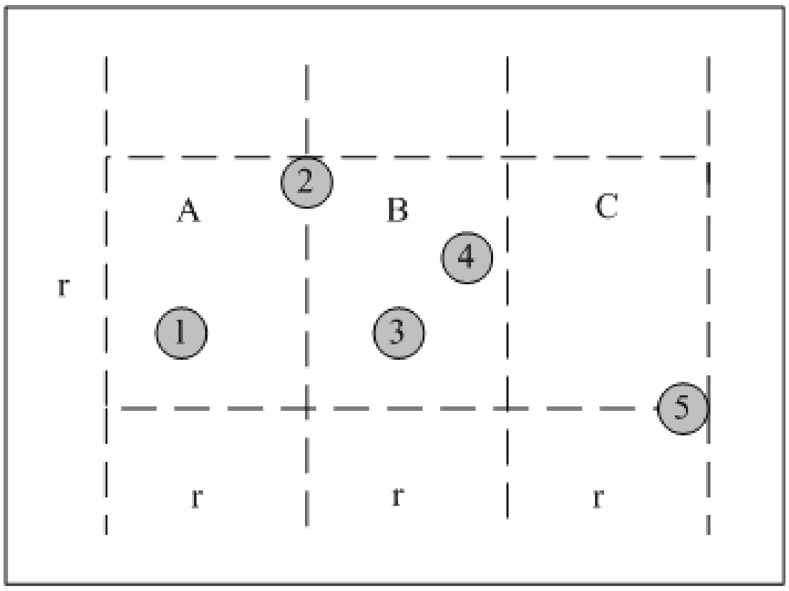


Figure 3.11: *Virtual grids in GAF. Source: [29].*

C. **Geographic Adaptive Fidelity (GAF)**, proposed by Xu *et al.* [29], is an energy-aware location-based routing algorithm for ad-hoc networks, but it could be applicable to sensor networks as well. Its focus is on turning the radio off as much as possible. In GAF, nodes use geographic location information to divide the world into fixed virtual grids. Each node uses GPS-indicated location to associate itself with a point in the grid. Nodes associated with same point on the grid are considered equivalent in terms of the cost of packet routing. For instance, in Figure 3.11, nodes 2, 3 and 4 are equivalent and two of them can sleep. Node 1 can reach any of node 2, node 3 and node 4; nodes 2, 3 and 4 can all reach node 5. Such equivalence is exploited in keeping some nodes located in a particular grid area in sleeping state in order to save energy. Thus, GAF can obviously increases the network lifetime as the number of nodes increases.

There are three states in GAF: *discovery state* (for determining the neighbours in the grid), *active state* (for reflecting participation in routing) and *sleeping state* (when the radio is turned off). Each node in the grid estimates its leaving time of grid and sends this to its neighbours in order to handle mobility. The sleeping neighbours adjust their sleeping time accordingly in order to keep routing fidelity. For each particular grid area, a representative node acts as the leader to transmit data to other nodes. However, it is not very scalable. And GAF strives to keep the network connected by keeping a representative node always in active mode for each region on its virtual grid. While such connectivity is ensured by self-organising the router sensor, MECN maintains an enclosure graph of the network by dynamically changing the transmitting range assignment of the nodes.

D. **SPAN**, which proposed by Chen *et al.* [36], is a distributed proactive algorithm. In SPAN, some nodes are selected as coordinators based on their positions. The coordinators form a network backbone used to forward messages and rotate with time. Unlike GAF, SPAN broadcasts messages to discover and react to changes in the network topology. Further nodes know their geographic positions. In SPAN, new and existing coordinators are not necessarily neighbours. However, it needs to maintain the positions of two or three-hop neighbours in SPAN. As a result, some overhead are introduced.

E. **Virtual infrastructure-based energy-efficient (VIBE)** [37] routing is a protocol which creates a virtual infrastructure to perform unicasting at the top level and support data aggregation. The key design principle of the VIBE protocol is to reduce the number of needed hops between source and destination, eliminating the transmissions usually needed by other protocols to choose the next hop node or just to communicate the positions of the nodes.

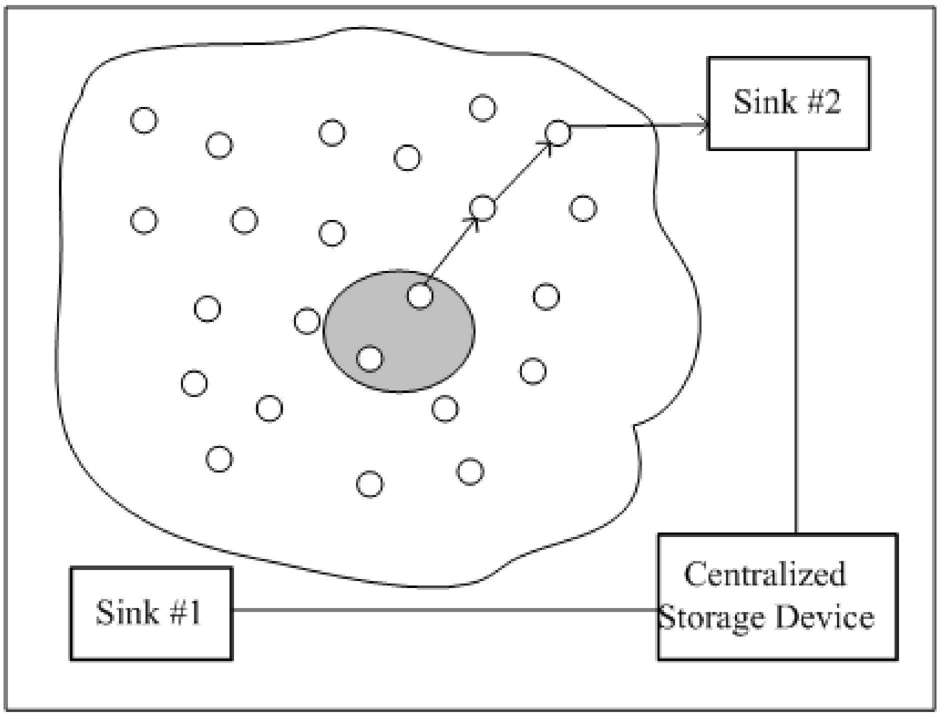


Figure 3.12: *Multihop routing from the area of interest (shade area) to the sink in the sensor field using the VIBE. Source: [37].*

Figure 3.12 shows the routing process from the interest area to the sink using virtual grid node. A communication message has to be transmitted to a ″centralised storage device″ in order to be processed with all the other information from other sensors. Such a device is part of the outside fixed infrastructure and then each sensor knows its location. After a sensor has evaluated its position, each node decides itself if it is a cluster-head or not. Each node computes its distance from the virtual infrastructure defined by the grid and makes this decision.

Then messages are forwarded one by one until the sink is reached. In VIBE, it is sufficient to compute the remaining distance to reach the sink and to compare it with the other possibilities that a message has each time in order to be forwarded.

3.2.4 QoS-based routing protocols

QoS-aware schemes consider end-to-end delay requirements while setting up the paths in the sensor network. In QoS-based routing protocols (SAR, SPEED, MERR, AMERR etc), the network has to balance between energy consumption and data quality, and to satisfy certain QoS metrics (such as energy, bandwidth and delay) when delivering data to the base station [12] .

A. **Sequential assignment routing (SAR)** algorithm [9] is a protocol in the sensor network with a scalably large number of static nodes. It is a table-driven multi-path routing protocol. It aims at energy efficiency and fault tolerance. It introduces the notion of QoS into routing decisions. The objective of the SAR algorithm is to minimise the average weighted QoS metric.

To avoid single route failure, SAR uses a multi-path approach and localised path restoration schemes. To create multiple paths from a source node, SAR builds a tree rooted at one-hop neighbours of the sink. By using created trees, multiple paths from sink to sensors are formed to avoid nodes with low energy or QoS guarantees. One of these paths is selected according to the energy resources and QoS on the path.

A routing decision in SAR is dependent on *three factors*: QoS metric, energy resource on each path, and priority level of each packet. For each packet routed through the network, a weighted QoS metric is computed as the product of the additive QoS metric. A weight coefficient associated the priority level of that packet is computed to evaluate the performance. By measuring the QoS provided to each packet relative to the priority level of the packet, the weighted QoS metric is achieved. To maintain the same weighted QoS metric, higher QoS (lower QoS metric) will be used for higher priority (higher weight coefficient) packets.

If the topology changes due to node failures, a path recalculation is needed. When the topology changes, as a preventive measure, a periodic path calculation is triggered by the sink. At the end of this process, each sensor node will be part of a multi-path protocol tree. Failure recovery is done by enforcing routing table consistency between upstream and downstream nodes on each path.

Simulation results showed that SAR offers less power consumption than the minimum energy metric algorithm, which focus only on the energy consumption of each packet without considering its priority. SAR maintains multiple paths from nodes to the sink to achieve fault tolerance. However, the SAR protocol suffers from the overhead of maintaining state information at each sensor node, especially when the number of nodes is huge.

B. **SPEED** [47]. Its goal is to provide a soft real-time communication service with a desired delivery speed across a sensor network. SPEED uses geographic forwarding to find the paths. It requires each node to maintain information about its neighbours. SPEED hopes to provide congestion avoidance when the network is congested. It ensures a certain speed for each packet in the network so that each application can estimate the end-to-end delay of the packets to find the paths.

The five modules in SPEED are shown in Figure 3.13: SNGF module, neighbourhood feedback loop module, backpressure rerouting module, delay estimation mechanism module, and beacon exchange module. Stateless Geographic Nondeterministic Forwarding (SNGF) is the routing module to choose the next hop candidate that satisfies the desired delivery speed. When a node fails to find a next hop node, the neighbourhood feedback loop module and back-pressure rerouting module are used to prevent voids, and to eliminate congestion by sending messages back to the source nodes so that they will pursue new routes. A node uses delay estimation mechanism to determine whether congestion occur. The beacon exchange module provides the geographic location of the neighbours so that SNGF can do geographic based routing. SPEED uses two on-demand beacons (delay estimation beacon and back-pressure beacon) to identify the traffic changes inside the network. Each node keeps a neighbour table to store informa-

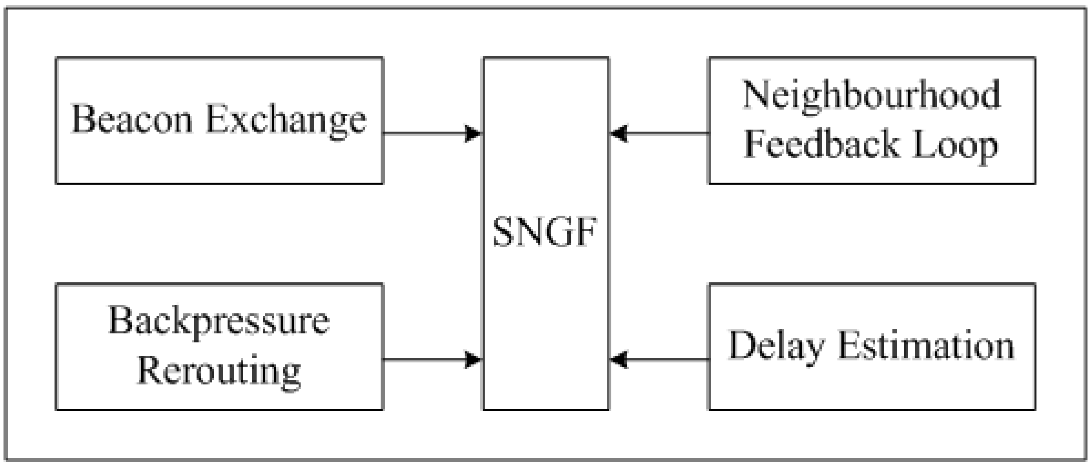


Figure 3.13: *The module of SPEED. Source: [47].*

tion received .

Delay estimation at each node is made by calculating the round trip single hop delay for this packet before an ACK is received from a neighbour as a response to a transmitted data packet. By looking at the delay values, SNGF selects the node which meets the speed requirement. If no such node is found, a relay ratio is calculated based on the Neighbourhood Feedback Loop (NFL) by looking at the miss ratios of the neighbours of a node (the nodes which could not provide the desired speed) and is fed to the SNGF module. If the ratio is less than the random number between 0 and 1 generated by the node, the packet is dropped.

In comparison to AODV and DSR, SPEED has the shortest average end-to-end delay and miss ratio due to the simplicity of the routing algorithm and the even traffic distribution. The SPEED protocol [47], such load balancing is achieved through the SNGF mechanism of dispersing packets into a large relay area. However, SPEED neither considers any further energy metric in its routing protocol nor do the authors compare it to any energy-aware routing protocol.

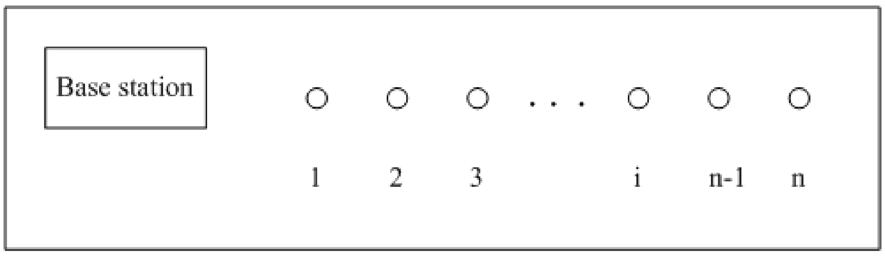


Figure 3.14: *A linear network of n sensors and a base station in MERR. Source: [48].*

C. **Minimum energy relay routing (MERR) [48] and Adaptive MERR (AMERR) [48]** are two routing protocols that utilise energy, and feature low complexity and good scalability. They focus on optimal power consumption in a linear sensor network (see Figure 3.14), where a Poisson model is assumed for the distribution of nodes along a straight line path. It assumes that a sensor node is aware of the distances to its downstream neighbours. The algorithm is localised in the sense that each node decides on the next hop based on the position of itself, of its neighbours, and possibly of the destination node.

Adaptive MERR (AMERR) achieves optimal performance for practical deployment settings, while MERR rapidly approaches optimal performance as sensors are more densely deployed. In AMERR, it makes better routing decisions than MERR at the cost of a lower degree of locality.

Both MERR and AMERR take into account the channel characteristics, the radio component, and the distribution of the sensor nodes along a linear path modelled by a one-dimensional homogeneous Poisson process. For a Poisson rate of 0.01 or greater, MERR rapidly approaches optimal performance as the Poisson rate increase while AMERR achieves the optimal performance.

However, MERR and AMERR do not consider the fairness metric. Some nodes which located on a single optimal path will exhaust their energy quickly. This will lead to a sudden partition of the network.

3.3 Clustering method

There are three approaches to routing in wireless sensor networks: multihop, direct and a hybrid of both [49].

Supposed E is the sample set and C is a nonempty subset of E. Clustering is the collection of classes that satisfy three conditions,

$$C \subset E, C \neq \varnothing \tag{3.2}$$

$$C_1 \cup C_2 \cup \ldots \cup C_k = E \tag{3.3}$$

$$C_i \bigcap C_j = \varnothing, i \neq j \tag{3.4}$$

Clustering divides a data set into clusters of similar objects according a certain pattern. It plays an important role in various fields, such as engineering, mobil robots, economics, medical and life science, geography, etc [50]. The main advantages of clustering are stability and robustness. The clustering problem is considered to be NP-hard [51], and evolutionary algorithms have been applied successfully to a variety of NP-hard problems.

3.4 Genetic algorithms (GA)

A genetic algorithm is a search heuristic that mimics the process of natural evolution first proposed by Charles Darwin in 1859. In the 1960's, Holland conducted pioneering works in this field [52][53]. Genetic algorithms differ from traditional search technologies. Genetic algorithms are proposed to automatically acquire and accumulate the necessary knowledge about the search space and to control self-adaptively the entire search process through random optimisation techniques.

The genetic algorithm is useful to find global or local optima in a large search space; it is particularly useful to avoid combinatorial explosion in applications involving design and optimisation. It has found many successful applications in such areas, as solving combinatorial optimisation problems and nonlinear problems, with complicated constraints or non-differentiable objective functions [54].

Algorithm 1 : the pseudo code of a genetic algorithm

```
1: Objective function $f(x)$, $x = (x_1, ...x_n)^T$
2: Encode the solution into chromosomes (binary strings)
3: Define fitness $F$ (eg, $F \propto f(x)$ for maximisation)
4: Generate the initial population
5: Initial probabilities of crossover($p_c$) and mutation ($p_m$)
6:     while ($t$ <Max number of generations)
7:         Generation new solution by crossover and mutation
8:             if $p_c$ >rand, crossover; end if
9:             if $p_m$ >rand, mutate; end if
10:        Accept the new solutions if their fitness increase
11:        Select the current best for new generation
12:    end while
13: Decode the results and visualisation
```

The genetic algorithm approach generates aggregation trees which span all the sensor nodes. Genetic operators include crossover, mutation and inversion. Genetic operators are performed on candidate solutions. According to their corresponding fitness values, good solutions are retained and unqualified ones are screened out. In each iteration cycle, genetic operators are performed to produce a population of new candidate solutions within pre-defined operation times. The above process does not stop until either the algorithm converges upon a particular solution or the allocated execution time is exhausted. The pseudo code of a genetic algorithm [55] is shown in Algorithm 1.

Crossover is an operation of segments exchange based on two solutions [54]. Given the following two parent solutions with their crossover points represented by "/":

S_1= 0101010/$\underline{11100}$

S_2= 1010011/$\underline{01010}$

the children solutions produced by a crossover operator are as follows:

S'_1= 0101010/$\underline{01010}$

S'_2= 1010011/$\underline{11100}$

The **inversion** operator involves reversing the order data in a solution segment [54]. Given one parent solution with the inversion segment enclosed by a pair of "/":

S_3= 010/$\underline{11101}$/1100

an inversion operator produces the following child:

S'_3= 010/$\underline{10111}$/1100

The **mutation** operator involves choosing one or more loci (loci is the specific location of a gene.) randomly in the individual string and changing their values (i.e. 0-1 reverses) with the current mutation probability. Given one parent solution as follows:

S_4= 01$\underline{\mathit{0}}$101111100

a mutation operator is performed on the gene locations denoted in italic type and produces the following child:

S'_4= 01$\underline{\mathit{1}}$101111100

Genetic algorithms can work effectively in finding the best and optimal solution of NP problems. The knapsack problem is a classical example of problems with a genetic algorithm implementation, as presented in Algorithm 2 ([56]).

3.4.1 Open issues in GA research

There are still some unsolved issues in genetic algorithms:

1. A variety of objection function fitness value setting methods is suggested in the literatures [57][58][59] and there is no unified formulation of the fitness function;

Algorithm 2 : genetic algorithms for the knapsack problem in C language

```
main ( )
{
    int gen_no;
    initialise( );
    generate(oldpop);
      for (gen_no=0; gen_no<maxgen; gen_no++)
      {
          evaluate(oldpop);
          newpop=select(oldpop);
          crossover(oldpop);
          mutation(newpop);
          inversion(newpop);
          oldpop=newpop;
      }
}
```

2. Premature convergence can occur in genetic algorithms. Their convergence speed is faster than ant colony algorithms but they might converge to a local optimum rather than the global one [60];

3. They do not make full use of the previous path feedback information in the convergence process [59].

In order to improve the computing performance of genetic algorithms, many researchers propose improvements in the decoding setting, the choice of the initial population, the choice of the fitness function, the choice of genetic operators, the choice of control parameters (p_c, p_m), redefining the algorithm structures [51][55] [61][62].

3.4.2 Factors influencing the performance of GA

The population's convergence rate is affected by various parameters (such as the choice of the fitness function, the choices of the crossover rate and the mutation rate [61]).

- **Design of the crossover operator**

A good design of crossover operator may ensure that the desirable gene segments of old individuals are inherited by the new individuals of the children generations; meanwhile, a low-efficient crossover operator may cause the search process to be trapped in a status or to be prone to ceasing.

- **The value of the probability of operations** (p_c, p_m)

The value of probability of crossover (p_c) will affect the searching results. If p_c is too small, the crossover occurs sparsely, i.e., which is not an efficient evolution [55]. Therefore, to avoid this situation, p_c is usually very great and normally in the range of $0.7 \sim 1.0$. If the mutation probability (p_m) is too high, the solutions could still oscillate about the optimal solution rather than converging upon it [55].

- **Choice of the fitness function**

One important issue is the formulation of an appropriate fitness function. A fitness function may determine the selection criterion in a particular problem [55]. There are many ways of defining a fitness function because there are a variety of special problems. Appropriate fitness functions will ensure that those solutions with higher fitness be selected efficiently; the poor form of fitness function may result in incorrect or meaningless solutions.

Simulation results in [63] show that the genetic algorithm gives better lifetime than the single-best-tree algorithm; for the small network size, the genetic algorithm gives the same network lifetime as the cluster-based maximum lifetime data aggregation algorithm does [57][64].

3.4.3 New approaches to genetic algorithms in WSNs

In the objective function, the amount of the total network energy consumption can be minimised based on the network *distance* and the *number of clusters*. Although the following approaches improve the performance of a genetic algorithm solution, there are a large number of redundant iterations once a genetic algorithm solution approaches the operation. These redundant iterations will result in low efficiency.

Jin *et al.* [65] maximised the fitness value to determine a good solution and try to find appropriate cluster-heads to minimise the total network distance. The fitness value depends on the number of cluster-heads (N_{CH}) and the communication distance (D).

$$Fitness = w \times (TD - D) + (1 - w) \times (TN - N_{CH}) \tag{3.5}$$

where

1. w ($0 \leq w \leq 1$) is a pre-defined application-dependent weight,
2. TD is the sum of the distances of all nodes to the sink,
3. D is the sum of the distances from regular nodes (i.e., non-cluster-head nodes) to cluster-heads plus the sum of the distances from all cluster-heads to the sink,
4. TN is the number of nodes,
5. N_{CH} is the number of cluster-head nodes.

The shorter is the value of D or the lower is that of N_{CH}, the higher the fitness value of an individual will be. The binary gene value can be "1", indicating that the corresponding node is a cluster-head node; otherwise, it is a regular node.

Mudundi *et al.* [66] proposed a fitness function which consists of three parameters: the number of cluster-head nodes (N_{CH}), the network Euclidean distance (ND) between all the nodes in each cluster and their cluster-heads, and the weight (w) for adjusting both ND and N_{CH} when computing a fitness value for each chromosome.

$$Fitness = w \times N_{CH} + (1 - w) \times ND \tag{3.6}$$

Each chromosome is represented as a fixed length list equal to the size of the network. The gene value could be "-1" (indicating that the corresponding node is dead), "0" (referring to a regular node) or a positive number (indicating the number of cluster-heads).

Martin and Hussian [67] extended the fitness function to the estimated transfer energy (E), the direct distance between all nodes to the base station (D) and the number of transmissions (T).

$$Fitness = \sum_i \alpha(\omega_i, f_i), \forall f_i \epsilon (C, D, E, SD, T) \tag{3.7}$$

where

1. ω_i is the preassigned arbitrary weight for adjusting the fitness parameters,
2. C is the sum of the distances from sensor nodes to cluster-heads and the distance from cluster-heads to the sink,
3. D means the direct distance to the sink and it is the sum of the distances from nodes to the sink,
4. E represents the energy consumed to transfer the aggregated message from the clusters to the sink,
5. SD is the standard deviation in cluster distances,
6. T is the number of transmissions assigned by the base station.

The harmony search algorithm ([62]) has been developed for improving the longevity of the network and reducing the energy consumption in the clustered routing. The algorithm defines the fitness of solution as follows:

$$Fitness = w \times f_1 + (1 - w) \times f_2 \tag{3.8}$$

$$f_1 = max\{\sum_{\forall i \epsilon C_j} d(node_i, CH_j)/||C_j||\} \tag{3.9}$$

$$f_2 = \sum_{i=1}^{TN} E(node_i) / \sum_{j=1}^{k} E(CH_j) \tag{3.10}$$

where

1. f_1 is the maximum of the ratio of Euclidean distance of nodes $(\sum_{\forall i \epsilon C_k} d(node_i, CH_j))$ to the number of nodes of cluster C_j;

2. $\sum_{\forall i \epsilon C_k} d(node_i, CH_j)$ is the sum of the distance between the nodes and their cluster-heads, and $d(node_i, CH_j)$ is the distance between $node_i$ and all cluster-heads CH_j, $j \in [1, k]$;

3. $||C_j||$ is the number of nodes of cluster C_j;

4. TN is the number of nodes;

5. k is the number of cluster-head nodes;

6. f_2 is the ratio of the energy of all the alive nodes in the network with the current energy of cluster-heads in the current round.

Enan *et al.* [51] tried to minimise the fitness function, which is defined as the total dissipated energy in the network. The fitness function consists of the sum of the energy dissipated in transmitting from the non-cluster-heads to their cluster-heads and the energy spent by cluster-heads to the sink.

$$Fitness = (\sum_{i=1}^{k} \sum_{s \epsilon C_i} E_{TX_{s,CH_i}} + E_{ER} + E_{DA}) + \sum_{i=1}^{k} E_{TX_{CH_i,BS}} \tag{3.11}$$

where

1. k is the number of cluster-heads,

2. C_i is the number of a non-cluster-head node associated with the ith cluster-head node, s belongs to the set of C_i.

3. $E_{TX_{i,j}}$ is the energy dissipated for transmitting data from node i to node j,

4. E_{TX} is the energy dissipated during the process of transmitting information,

5. E_{ER} is the energy dissipated during the process of receiving information,

6. E_{DA} is the energy dissipated in aggregating data on cluster-heads.

The E_{TX}, E_{ER} and E_{DA} parameters used match those in the SEP [39] energy model.

3.5 Ant colony optimisation algorithms (ACOs)

Swarm intelligence implements the artificial intelligence by modelling the group behaviour of biological species, such as colonies of ants, groups of fishes, bees and flocks of birds. The ant colony optimisation algorithm is one class of swarm intelligence algorithms (SI). It was introduced in [68] as a heuristic method for solving combinatorial optimisation problems. It is inspired by the behaviour of ant colonies finding the shortest path between their nest and a food source. The principle of ACO algorithms is to find the optimal solution through the exchange of information between individuals and mutual cooperation [69]. As one of swarm intelligence, the main advantages of ant colony optimisation algorithms are as follows:

- Individuals work in a distributed way. There is no centralised control in the system. Therefore, the ultimate solution of the whole system may be unaffected if one or some individuals are paralysed [70].
- Individuals work in an independent and cooperative way [70].
- It provides a more extensible way to build the system, since the individuals communicate with each other in an indirect way and which will not bring too much communication costs when the group scale increases [70].
- It provides a convenient way of obtaining realisations of the system because the function is simple and the execution time of each individual is short [70].

In the natural world, foraging ants explore surroundings of their nests in a random manner. As shown in Figure 3.15, it is the simplest double bridge problem with two branches where the right-side route is shorter than the left-side route. At the initial stage at the nest, the ants have an equal probability of choosing each route randomly. The first batch of ants are fed workers and they depart from the same nest (gray node). After some iterations, almost all the ants will move along the shorter route. It is straightforward to extend the algorithm for two routes to support multiple routes at a node. The ant colony algorithm features as several steps as shown as follows:

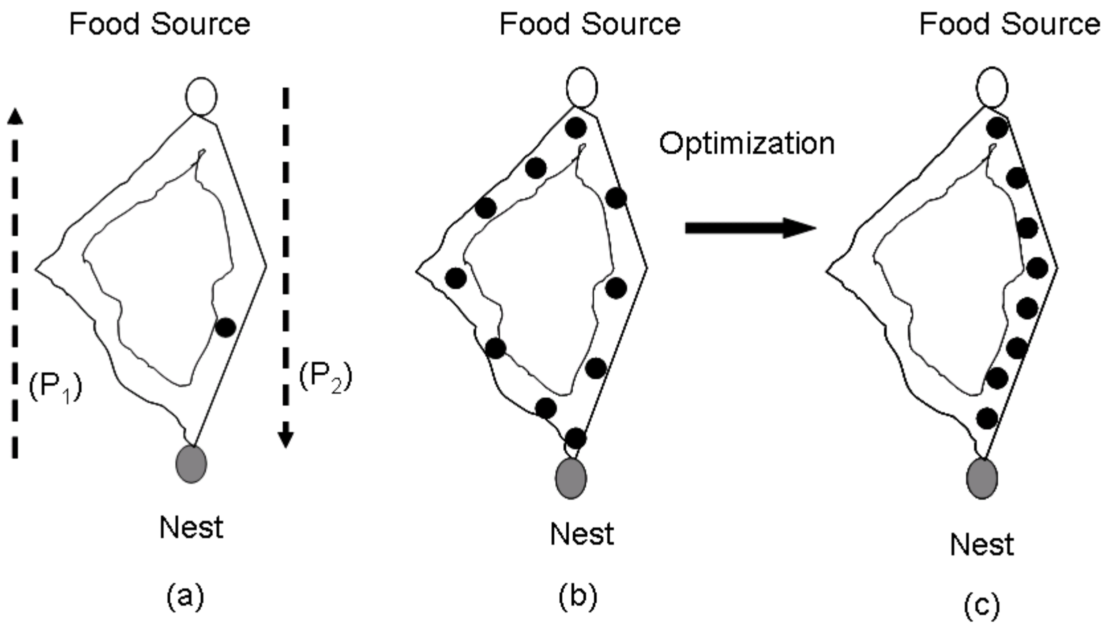


Figure 3.15: *Foraging behaviour of ants in nature.*

1. When an ant finds a source of food (white node), it evaluates the quantity and quality of the food. Then, it carries some food to the nest; meanwhile, it deposits a trail of chemical pheromone (P_2) on the way home. The chemical pheromone will guide other ants to the food source.

2. The first batch of ants to find the food source (P_1), return to the nest, leaving behind a pheromone trail. Suppose that all ants move at the same speed. Since a round trip on a shorter path costs less time, the intensity of pheromone on the shorter path is higher than on the other path.

3. After a certain time, almost all ants choose the shortest path between the food source and the nest, which is shown in Figure 3.15.(c).

The pseudo code of the basic steps of the ant colony optimisation algorithms (ACO) [55] is shown in Algorithm 3.

Algorithm 3 : the pseudo code of an ant colony optimisation algorithm

```
1:  Objective function $f(x)$, $x = (x_1, ...x_n)^T$
2:  Define pheromone evaporation rate $\gamma$
3:  while $t$ ¡ maximum number of iterations
4:      for loop over all $n$ nodes
5:          Generate new solutions
6:          Evaluate the new solutions
7:          Mark better routes with pheromone $\delta\phi_{ij}$
8:          Update pheromone: $\phi_{ij} \longleftarrow (1-\gamma)\phi_{ij} + \delta\phi_{ij}$
9:      end for
10:     Select the current best solution best.route
11: end while
12: Output the best solution best.route and pheromone distribution
```

3.5.1 ACO algorithms versus traditional routing algorithms

The advantages of ACOs are self-adaption, self-organisation, flexibility, robustness, parallel computing, no need of prior information, and ACOs have been successfully applied to combinatorial optimisation problems such as the Internet routing problem, the travelling salesman problem, assignment problems, scheduling problems and the vehicle routing problems as described in [55][57][70].

The main differences between ant colony algorithms and traditional routing algorithms, including distance vector routing (e.g., routing information protocol (RIP)) and link state routing (e.g., open shortest path first (OSPF)), are the mechanism for adapting to topology change and the method for updating routing. RIP and OSPF need transmit the routing table or flood link-state-packets at regular intervals to adapt to topology change, and update the whole routing table when the changes happening. Ant colony algorithms provide only need update one entry in a pheromone table when this situation happening.

3.5.2 Factors influencing the performance of ACO algorithms

The main directions of the researches are the integration of ACOs and other clustering algorithms, or use different overall strategies and different pheromone attributes to solve clustering problems in the literature [70][71].

For example, in paper [72], Liao *et al.* replaced distance with number of hops in ACO global pheromone increment formula in Equation (3.16). The pheromone update formula is as follows:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij} \tag{3.12}$$

$$\Delta\tau_{ij} = [\zeta + (h_i - h_j)] \times \Delta\omega_j \tag{3.13}$$

where

1. ρ is the pheromone decay parameter,
2. ζ is a predefined positive number,
3. h_i is the number of hops from node i to the sink, h_j is the number of hops from node j to the sink.
4. $\Delta\omega_j$ is the number of hops from some sources to the sink via node j.

In Equation (3.13), if $(h_i - h_j) \geqslant 0$, it indicates that node j is closer to the sink than do node i, thus the node j will be chosen as the next hop node; otherwise, it reverses.

There are three important issues which affected the performance of ACO algorithms, the population size, the probability of choosing a route and the evaporation rate of the pheromone [55].

- **The probability of choosing a route** p

For a routing problem, Equation (7.6) indicates that ants would follow the paths with higher pheromone concentrations. The probability of ants at a node i choosing the route from node i to node j (among n_d nodes) is given by

$$p = \frac{\phi_{ij}^{\alpha} d_{ij}^{\beta}}{\sum_{i,j=1}^{n_d} \phi_{ij}^{\alpha} d_{ij}^{\beta}} \tag{3.14}$$

where

1. α, β ($\alpha, \beta > 0$) are predefined parameters that control the influence of heuristic information. Their typical values are $\alpha \approx \beta \approx 2$ [55].
2. ϕ_{ij} is the pheromone concentration on the route between i and j;
3. s_{ij} is the distance of the same route;
4. d_{ij} is the desirability of the same route, $d_{ij} \propto 1/s_{ij}$, which implies that shorter routes will be selected due to their shorter travelling time. Thus, the pheromone concentrations on these routes are higher. This is because the shorter the travelling time, the less the amount of the pheromone that evaporates during this period.

- **The evaporation rate of pheromone decay or evaporation** γ

The pheromone concentration (ϕ_{ij}) varies with time exponentially for a constant rate (γ) of pheromone decay or evaporation as follows:

$$\phi(t) = \phi_0 e^{-\gamma t} \tag{3.15}$$

If $\gamma t \ll 1$, $\phi(t) \approx (1 - \gamma t)\phi_0$. For the unitary time increment $\Delta t = 1$, we approximate the evaporation as $\phi_{ij}^{t+1} \leftarrow (1 - \gamma)\phi^t$. Thus we have the pheromone update formula as follows:

$$\phi_{ij}^{t+1} = (1 - \gamma)\phi_{ij}^{t} + \delta\phi_{ij}^{t} \tag{3.16}$$

where

1. ϕ_0 is the initial concentration of pheromone,
2. t is the current time,

3. $\gamma \epsilon$ [0,1],

4. $\delta\phi_{ij}^{t}$ is the increment of the amount of pheromone deposited at time t at route i to j when an ant travels a distance L. Usually $\delta\phi_{ij}^{t} \propto 1/L$, only if there are no ants on a route, $\delta\phi_{ij}^{t}$ will be equal to 0.

However, the above algorithms cannot solve the main shortcoming of ant colony algorithms. This shortcoming is presented in paper [59]: ant colony algorithms have a slow speed of convergence if there is little or no information in the initial searching.

3.6 Limitations of existing routing algorithms

The routing strategies used in the algorithms described in this chapter can be classified into two broad types, namely globalised strategies and localised strategies. If using a globalised strategy, the protocol makes routing decisions based on the global state of the system, in so far as it is known. In localised routing algorithms, the nodes make routing decisions on the basis of the location of their neighbours and the destination.

Global routing algorithms can exploit the state information available to them in order to maximise network lifetime. However, it is necessary to transmit relevant energy information between nodes in order to gather this state information; a global addressing scheme is thus required and the overhead of ID maintenance rises rapidly with the number of nodes. It is impractical to deploy such algorithms in networks containing a large number of sensor nodes.

Localised routing algorithms have better scaling properties, but solutions which feature direct communication with the base station show serious network lifetime performance degradation when the base station is far away from the nodes. Lindsey *et al.* (PEGASIS) used a hierarchical chain-based binary aggregation scheme to achieve a high degree of parallelism and so reduce the energy-delay product. The SEP protocol and Heinzelman *et al.* (LEACH) used clustering methods to improve the network lifetime performance. Kumar *et al.* (EECHE) considered the residual energy in each node in deciding the weighted election probabilities of each node to become a cluster head.

All of these protocols (i.e., PEGASIS, LEACH, SEP, EECHE [73]) reduce energy consumption by using cluster heads to aggregate data.

The goal of maximising network lifetime in large-scale sensor networks is further complicated in resource-constrained heterogeneous sensor networks where traditional homogeneous routing techniques are inefficient. New techniques are needed to effectively exploit the hardware capabilities of the multiple types of mote present in such networks.

One approach to extending the network lifetime is by using topology control to keep the transmission distance low. Two new algorithms (TinyReg and TSEP) will be described in the following chapters that take this approach; the other novel algorithm (GAAC) is intended to improve the convergence speed in the process of searching for the optimal routing solution. All of the proposed algorithms are scalable to meet the needs of real-world applications of wireless sensor networks.

3.7 Summary

In this chapter, I have discussed three issues: the primary routing challenges, the important routing protocols and clustering technology in sensor networks. I also provide a brief introduction to genetic algorithms (GA), ant colony algorithms (ACOs) and their applications in the routing problem in wireless sensor networks.

In short, new routing strategies are required for sensor networks that are capable of effectively managing the trade-off between optimality and efficiency.

CHAPTER 4

FRAMEWORK AND ARCHITECTURE

In this chapter, I propose a software framework which already used in three protocols I designed) and a three-tier architecture (used by the TinyReg and TSEP protocols).

4.1 Motivation

During the development of my routing algorithms, I think there exist common features between routing algorithms in wireless sensor networks no matter these algorithms fall into which class of routing protocols. Therefore, I propose a software design framework (GSF) for routing protocols in future works.

Given the various applications of sensor networks, we need to consider a range of constraints on the system architecture in designing routing protocols. In a sensor network, nodes can be dynamic or fixed; the node deployment in wireless sensor networks can be deterministic (i.e., manual) or self-organising (i.e., randomly); the network can be heterogeneous (i.e., including different hardware and software) or homogeneous (so that all nodes are equipped with the same reserves of energy and consume it at equal rates); the data delivery model to the sink can be continuous, event-driven, query-driven or hybrid; nodes may or may not have identical roles; a data aggregation function can be performed either partially or fully in each node.

Based on the above studies, I present a generic software framework (GSF) to feasibly and efficiently meet the needs of the users and applications (e.g., reliability, energy efficiency) from a code reuse point of view. The components in the GSF framework are designed to work cooperatively or separately to efficiently meet the customised needs of the users and applications.

4.2 Proposed software framework

The generic software framework (GSF), which consists of the following modules, is shown in Figure 4.1.

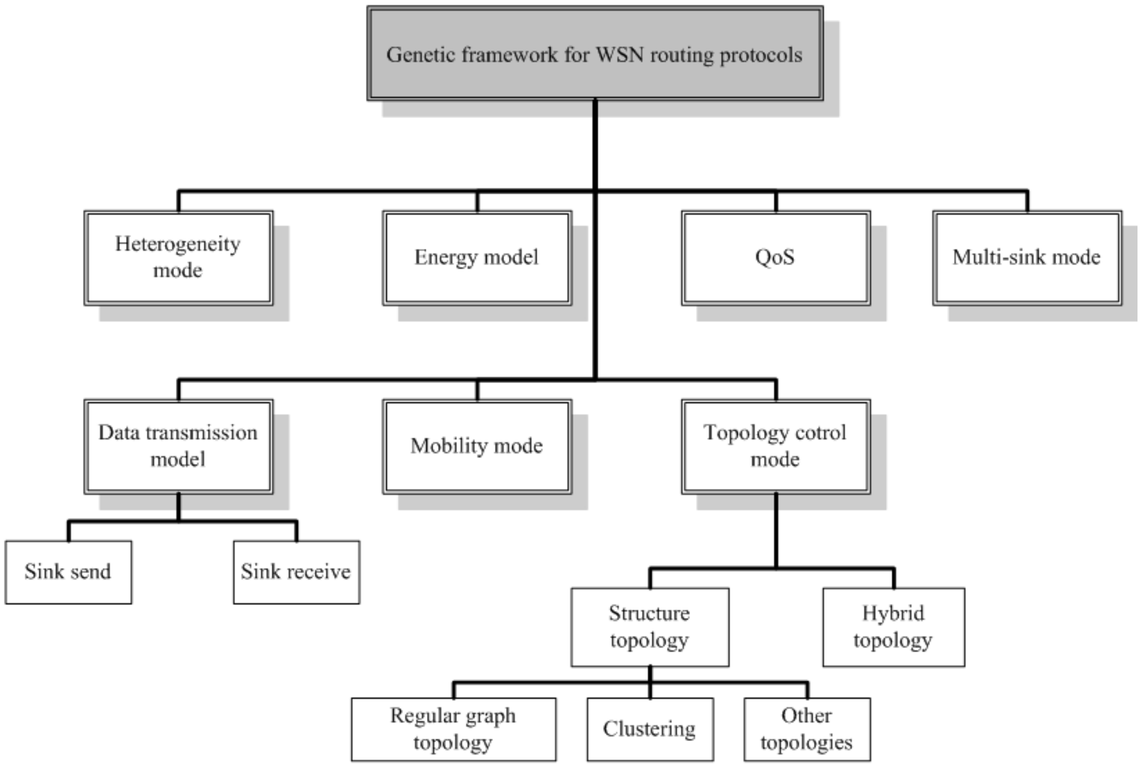


Figure 4.1: *Generic software framework for WSN routing protocols*

- **Data transmission model component.** It provides two options: *sink-send mode* and *source-send mode*. These two modes are respectively an active mode and a passive mode, i.e., either the sink is the sender or the sink is the receiver. Further, the data delivery model to the sink can be flexible and be set as one of continuous mode, event-driven mode, query-driven or hybrid mode. In the *continuous* mode, each sensor sends data periodically. In *event-driven and query-driven* modes, the data transmission is triggered by an event or a query generated at the sink. In the *hybrid* mode, any combination of the above three data delivery models are achieved depending on the needs of applications.

- **Energy model component.** Depending on the hardware and application needs, different radio energy dissipation models should be used. For example, if data aggregation function is not performed in a routing protocol, the energy consumption of nodes for aggregate packets should be excluded in the energy model. Therefore, it is convenient to reset parameters of the energy model in this component.

- **Topology control component.** This component dynamically adjusts to the needs of an application by selecting an appropriate topology. If the routing protocols and infrastructures use uniform topology-constrained techniques (or based on a single topology structure), it may not suit some applications. Thus, I provide two subcomponents whose codes can be reused in the future.

  - **Structured topology.** It consists of multiple topologies, including clustering structure, chain based and regular structure.
  - **Hybrid topology.** In an effort to meet application needs (e.g., QoS), it dynamically combines a variety of topologies.

- **Heterogeneity mode component.** The choice of this mode depends on application requirements. In many early studies, all the network nodes were assumed to be *homogeneous*. Node homogeneity indicates that nodes have equal power, computation and communication capacities. In a complex application, it might have at least two or more types of sensors to achieve multiple sensing functions. From the economic point of view, when we add a new type of sensors or update hardware in the network, we can update associated codes in this component and do not need recode the whole routing protocol.

- **Mobility component.** This mobility depends on the application scenarios. The routing protocol can adjust between a static environment or mobile scenario by flexibly adding or removing a mobile component. At times support for the mobility of sinks or cluster-heads is required.
- **Multi-sink mode component.** Sometimes it is necessary to switch to a multi-sink mode.
- **Quality of service component.** This depends on application requirements. As energy is depleted, the network may be required to reduce the quality of results in order to reduce energy dissipation in the nodes and accordingly lengthen the total network lifetime.

The mobility component, multi-sink mode component and quality of service component will be finished in future work.

4.3 Proposed three-tier architecture

Considering a multi-sensor system, I design a three-tier architecture which consists of a regular graph network and a clustering network. Proposed two algorithms (TinyReg, TSEP) are based on this architecture. The three-tier architecture is shown in Figure 4.2. From a functional point of view, it consists of a *Sensor Tier*, a *Relay Tier* and a Base *Station Tier*.

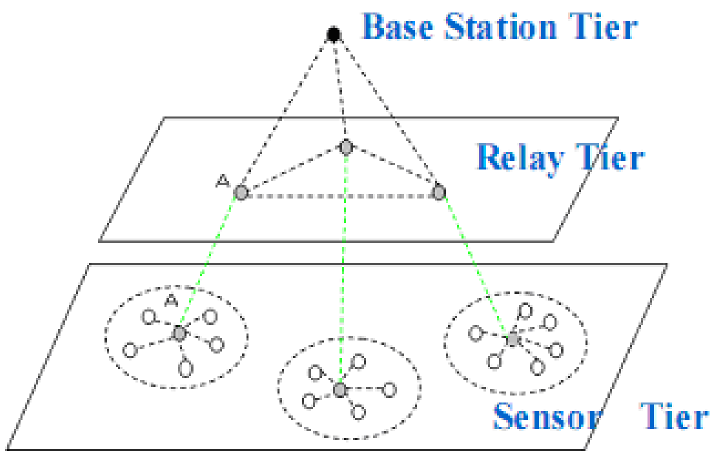


Figure 4.2: *The three-tier sensor architecture*

- **Sensor Tier.** This is composed of tens to thousands of sensors. Its purpose is to collect data from the sensing field, and to hand over that data to the Relay Tier. It features a star topology. Compared to mesh and other structures, the star topology incurs minimal overhead to maintain the infrastructure. Therefore, when a node fails by energy exhaustion or other reasons, it is easy to maintain the underlying network.

- **Relay Tier.** It consists of cluster heads. Cluster heads incorporate their own data and all sensing data received from their clustering field, and transfer that data to upper-layer cluster heads. The Relay Tier structure is a regular triangle structure. I use regular graph theory to maintain the connectivity and balance load among cluster heads in networks. This ensures a stable backbone tier with high transmission reliability.

- **Base Station Tier.** The base station receives all sensing data in the whole field via multi-hop transfer. The base station is configured at a user-selected location to connect to the Internet. Thus users can access the whole wireless monitoring system from off site.

The three-tier architecture has two technological features:

- **Multi-hop routing.** Multi-hop route shortens the distance of message transmission of cluster-heads, so it decreases the energy consumption of cluster-heads for inter-network communication. In general, the transmission radii of sensor nodes are similar to each other in homogeneous wireless sensor networks; for a heterogeneous set of sensor nodes, variance of the transmission radii is higher. The transmission range depends on the power available at the node. Another reason for using multi-hop routing is to extend the range of a sensor network and decrease the radio-frequency interference of sensor nodes. The longer the transmission radii is, the stronger the radio-frequency interference among neighbours.

- **Hybrid topology.** Using self-organisation of complex structures in the first level network, it reduces the impact of a single node failure and removes the intra-node performance bottlenecks. If we only use the completed one-type structured topology, it will dramatically limit scalability. Since the distance that the data can transmitted from the wireless device to the gateway is limited to a range of 30-100 metres. In regular graph topology, rigid structure constraint weakens the scalability and heterogeneity of aggregation topology model. In star topology (a line-of-sight architecture), data may be lost if any environment changes that may interrupt that line-of-sight transmission. To eliminate these limitations, it is better to keep more than one transmission path between device and gateway in a topology.

4.4 Summary

In this chapter, I propose a software framework, called GSF and a three-tier architecture. The framework is used in design of all three algorithms proposed.

CHAPTER 5

A HYBRID ROUTING ALGORITHM

In this section, I propose a novel global clustering routing algorithm (TinyReg) for the heterogeneous environment. Wireless sensor networks pose a lot of unique technical challenges due to their distinctive factors such as insufficient network bandwidth, limited transmit power, and large numbers of nodes supporting under short-range communication.

5.1 TinyReg review

In a multi-hop communication, the failure of one relay node can lead to the disconnection of a number of nodes from the base station. Algorithms [43][28], which feature direct communication with the base station, also cause the disconnection a significant number of nodes from the base station when key nodes failed (such as cluster-heads). That is why I consider regular graph structure to keep a redundant link amongst communication links among key nodes. That is the difference between TinyReg and other existing algorithms. To ensure energy-effective and self-adaptive operations of sensor networks, the design takes into account the energy-level E and the distance to cluster-heads D in the network as the key parameters.

The notion of *distance* D used above is the Euclidean distance between nodes. This distance metric is used in all of the algorithms (TinyReg, TSEP and GAAC) that I have proposed. Because energy consumption is proportional to communication distance, energy consumption can be reduced by keeping the transmission distance low. In TinyReg, the distance metric is considered during the cluster construction phase, when nodes join the nearest cluster-head i, the decision of which to join being made on a node-by-node basis. This feature means that the algorithm is less centralised, and thus has better scalability properties, than algorithms that require extensive state information during cluster construction.

5.2 Energy analysis of the algorithm

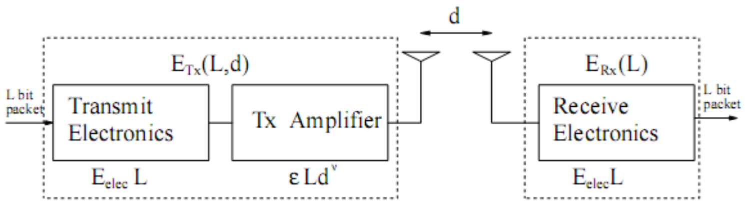


Figure 5.1: *Energy dissipation model*

Figure 5.1 shows a common energy model proposed by Heinzelman *et al* in [28]. Optimal clustering depends highly on the energy model they used. Many routing algorithms for wireless sensor networks (e.g [39]) either use the whole Equation (5.1) or one part of this equation for the energy model (see Figure 5.1). Because this energy model is a common model, we use it for message transmitting and receiving.

In order to achieve an acceptable Signal-to-Noise Ratio (SNR) in transmitting an L-bit message over a distance d, the energy E_{Tx} expended by the radio is given by:

$$E_{Tx}(L,d) = \begin{cases} L \times E_{elec} + L \times \varepsilon_{fs} \times d^2, & d < d_0 \\ L \times E_{elec} + L \times \varepsilon_{mp} \times d^4, & d \geq d_0 \end{cases} \tag{5.1}$$

where

- E_{elec} is the energy dissipated per bit to run the transmitter or receiver circuit;
- ε_{fs} and ε_{mp} depend on the transmitter amplifier model;
- d is the distance between the sender and the receiver. We obtain the cross-over distance $d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}(= 86.2m)$ as the cut-off point by equating the two expressions at d = d_0.

Besides, the radio expends an amount of energy E_{Rx} in receiving an L-bit message as follows:

$$E_{Rx}(L) = L \times E_{elec} \tag{5.2}$$

According to the radio energy dissipation model (in Figure 5.1) and Equations (5.1) and (5.2), the energy consumption for message transmitting depends highly on the distance between the transmitter and the receiver using the free space (d_2 power loss) and the multi-path fading (d_4 power loss) channel models. In most small-scale sensor networks, such as a 100m×100 m field, the distance between nodes is seldom greater than d_0 while in large scale sensor networks the reverse is true.

I must reduce the energy consumption by keeping the transmission distance low. TinyReg can keep the transmission distance (from cluster-heads to the sink) low using multi-hop relaying between the cluster-heads and the base station.

5.3 Topology structure analysis

Network layer structure directly affects the upper layer routing protocols' performance. The current most common wireless network topologies are star, mesh

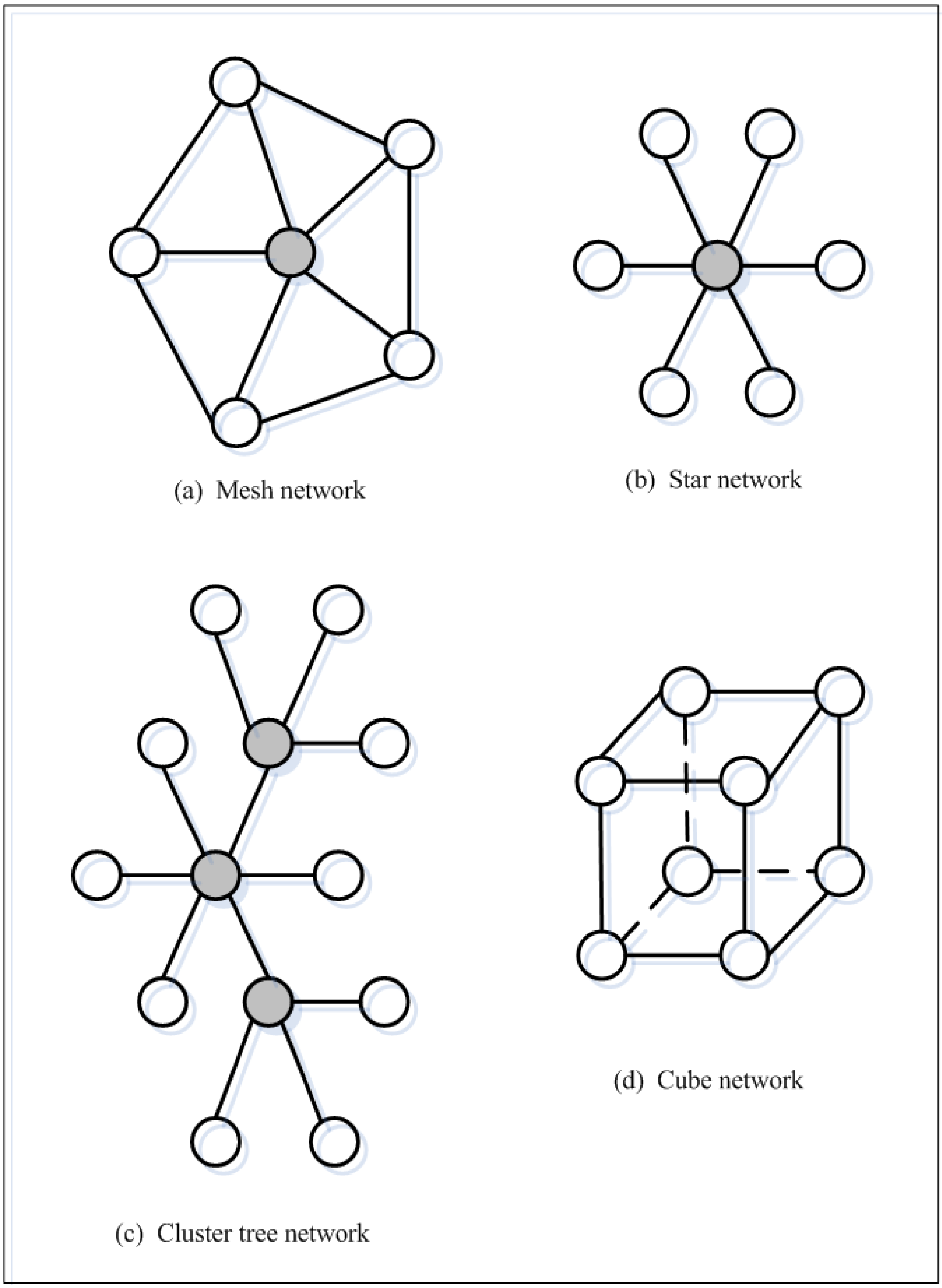


Figure 5.2: *Typical topology networks*

and cluster-tree topology as shown in Figure 5.2.

Mesh topology. Generally, there are multiple routing paths between two non-adjacent nodes. These networks are robust to failure of individual nodes and links as well. Compared to star topology, mesh topology has a longer range distance and decreases the data loss.

Star topology. A star topology is a point-to-point architecture in which all nodes are considered peers and every sensor node has the equal role and function. The star topology has minimal overhead to maintain the infrastructure compared to other structures.

Cluster-tree topology. Cluster-tree topology blends the advantages of star topology and mesh topology. The cluster-head plays a more prominent role than other members in the network.

Regular graph. A cubic graph is shown in Figure 5.2.(d), which is a classic structure of k-regular graph. A k-regular graph topology has regular degree in which all nodes are of degree k. Regular graph has a special robust feature, which is proposed by Nash Williams (1969) in [74]. This robust feature keeps average control overheads in networks as a constant.

5.4 Implementation of TinyReg

A pseudo-code description of TinyReg for the initial network construction is as follows (see Algorithm 4):

I consider a two-tier model and a wireless sensor network consisting of a base station and n sensors. It is assumed that the initial energy of the nodes in the network is known.

As shown in Figure 5.3, the two-tier model consists of two parts: Tier 1 is a 3-regular graph topology; Tier 2 is a clustering topology.

In my design, a clustering topology is used as the underlying network. Messages are transmitted along a multi-hop route from sensor nodes to the base station. All cluster-heads are connected with each other and form a hierarchy. The path length of communication tree in this topology will be shortened by parallel transmission.

In Tier 1, TinyReg forms a tree-like hierarchy and the tree is ”balanced”. Tier 1

Algorithm 4 : TinyReg

```
1: Generate a set $U(1,2,..,n)$ of cluster-heads in a hierarchy
2:     Compute function $M = k_1 \times E + k_2 \times D$. $E$: the residual energy of a node, $D$: the distance between a node (outside the network) and cluster-heads in the network.
3:     Reorder all nodes according to their value of $M$ in ascending order
4:     $t = 0$
5:     **while** ($t <$ Max number of cluster-heads)
6:         Select a node with the minimum value of $M$ as a cluster-head, and save the result in $U$
7:         $t = t + 1$
8:     **end while**
9: The rest of nodes in the network will be allocated to the closest clusters based on measuring transmission distance between them and the cluster-heads
10: Compute $distance = |U(1,2,..,n) -$ the base station$|$
11: $U.d \leftarrow distance$
12: Sort $[U.d]$ in ascending order
13: Set $V(1,2,..,n) \leftarrow \emptyset$, $V$ is a set of cluster-heads in the network
14: $V(1) \leftarrow U(d_{min1}), V(2) \leftarrow U(d_{min2}), V(3) \leftarrow U(d_{min3})$
15: $U(1,2,..,n) \leftarrow U(1,2,..,n) - U(d_{min1}) - U(d_{min2}) - U(d_{min3})$
16: **while** ($U \neq \emptyset$)
17:     For every node $i$ ($i \in U$), compute the distance between $i$ and each node in set $V$, save the value of the distance as a distance attribute of $i$
18:     $V(4) \leftarrow U(d_{min1}), V(5) \leftarrow U(d_{min2})$,
19:     $U \leftarrow U - U(d_{min1}) - U(d_{min2})$
20:     For every node $i$ in set $U$, compute the distance between $i$ and each node in set $V$, save the value of the distance as $i.d$
21:     $V(6) \leftarrow U(d_{min1}), V(7) \leftarrow U(d_{min2})$
22:     $U \leftarrow U - U(d_{min1}) - U(d_{min2})$
23:     ...
24: **end while**
25: Output $V$
```

consists of multiple levels (Level 1, Level 2, ... Level $\lceil \log_3^n \rceil$, where n is the number of nodes). In Figure 5.4, nodes (1), (2) and (3) are the leaders for Level 1. They will transmit the message to the base station. Since node (1) is in Level 1 in the hierarchy, nodes (4) and (5) which are in the same level (Level 2) will aggregate their self-generated data with their received data and send the aggregated data to node (1). Similarly, at the Level 3, nodes (10) and (11) will combine their current data with that received from the lower level and transmit the message to node (4). The dashed lines in Figure 5.4 indicate redundant links. These links are used only when the shortest link (as shown by the solid line) has failed. The maximum de-

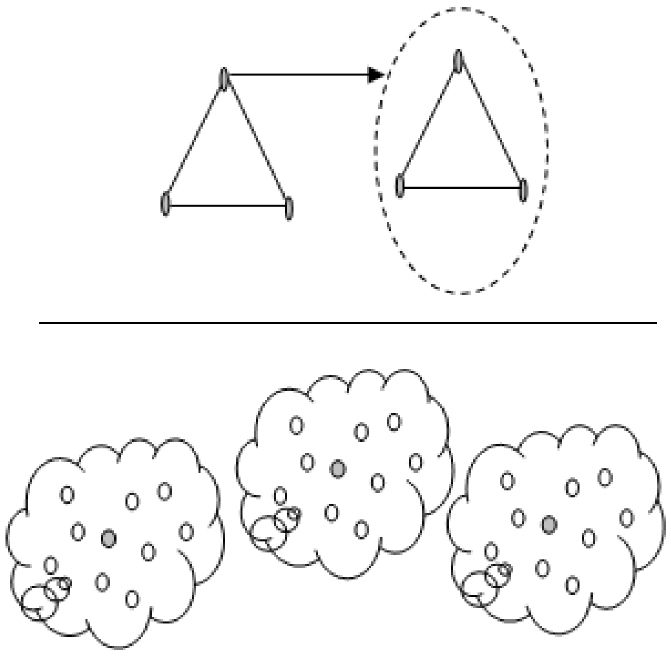


Figure 5.3: *The two-tier model. The upper sub-figure is Tier 1 (a 3-regular structured topology) where the arrow points to a complex node. The lower one is Tier 2 (a clustering topology). The shadow nodes in figure are cluster-heads and together comprise Tier 1.*

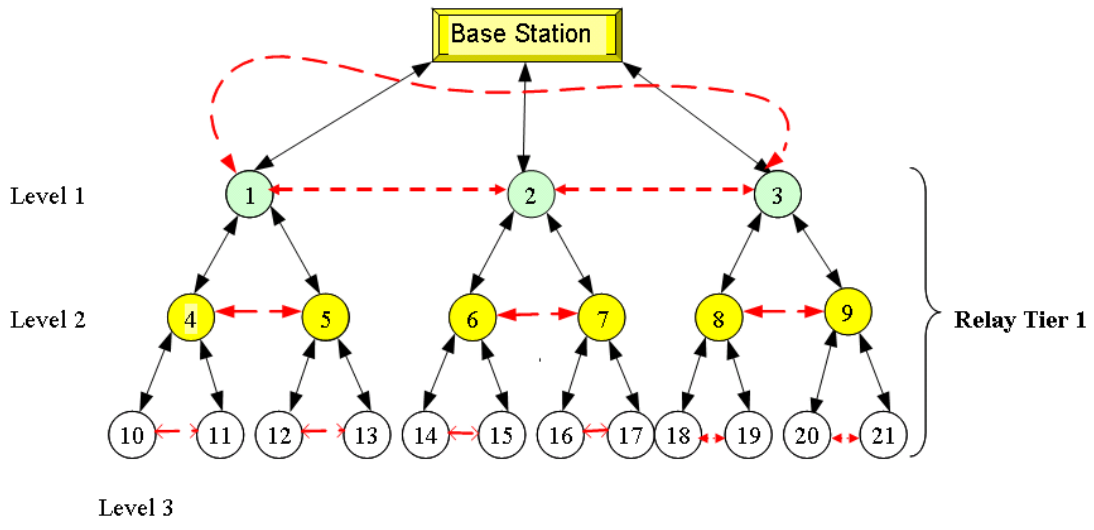


Figure 5.4: *Data transmission path in tree*

gree of a node (other than the cluster-head itself) is three in Tier 1. I choose more than $2k+1$ nodes to form the k-regular graph. The path length of communication tree in this topology will be shortened by parallel transmission.

In Tier 2, when the nodes join the nearest cluster-head i, the member count of cluster i is incremented where $i \epsilon [1, 2, ..n]$), and where n is the number of sensor nodes distributed across the target region.

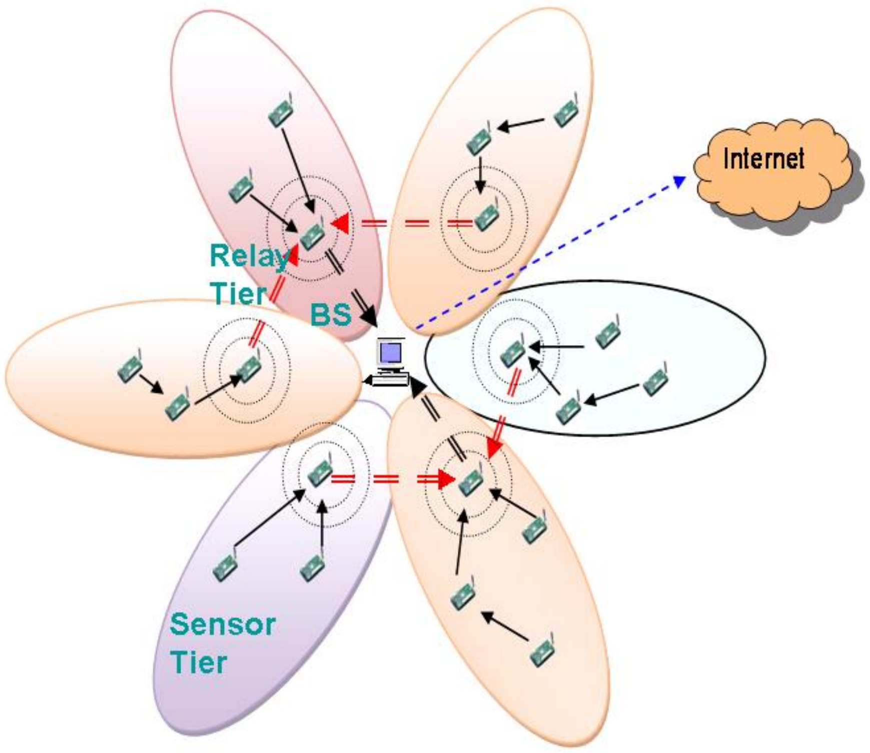


Figure 5.5: *Practical wireless sensor networks*

Based on the two-tier model, a self-organising network is constructed as shown in Figure 5.5. There are three tiers: a sensor tier, a relay tier and a base station tier. These tiers are hierarchical, and most nodes belong to the sensor tier. The nodes in the relay tier communicate with each other and act as ”bypass” connections across the sensor tier. The base station tier sits on the top of the system.

In the initial construction of the network, the routing problem is broken into two phases: a global planning phase in which the cluster-heads are determined and an architecture is constructed using global information; and a local phase in which the nodes joining clusters are formed on a node-by-node basis. In this phase, sensors are grouped into clusters. TinyReg employs cluster-heads to maintain the states of the sensors and to set up multi-hop routes for collecting sensor data.

In the initial construction of the network, each cluster-head is elected based on its energy reservoir and its position in the network. Clusters are formed and merged by constructing a three-tier hierarchy. The cluster-head selection scheme uses the following metric value M to calculate and select a potential cluster-head in the initial network construction:

$$M = k_1 \times E + k_2 \times D \tag{5.3}$$

where the quantities are normalised so that $E \in (0, 1)$, $D \in (0, 1)$, $k_1 \in (0, 1)$ and $k_2 \in (0, 1)$. E is the residual energy of a node, D is the distance between a node (outside the network) and cluster-heads in the network, k_1 and k_2 are weights.

The base station constructs broadcast trees and it broadcasts control packet from the sink to sources. After the first round of network operation, the sink receives the energy-based metric E of nodes according to the remaining energy of nodes. The TinyReg control packet is a type of packet that is used by both nodes and the base station to send control messages to each other. The routes to all the possible destinations are saved in the routing table of the Base Station (BS)). The base station broadcasts control packets to cluster-heads and normal nodes via multi-hop routing. Messages in TinyReg are transmitted along multi-hop routes from cluster-heads to the base station. Routes are determined after the initial construction of the network by the base station, and are rebuilt only when the BS receives notification of the death of a cluster-head.

To maintain the existing network, the base station and cluster-heads respectively send control messages to cluster-heads and normal nodes (i.e., regular nodes) for every time slot (round) to check whether they are still alive. Based on the received information, the base station updates its routing table and the energy level information. Then, it broadcasts the information to cluster-heads and sources. The cluster-heads receive these control messages and delete the dead cluster member.

Group reorganisations are performed in response to node failures and partition. When cluster-head failures occur in the network of Tier 1, communication interruptions may happen. This may lead to a dynamic load imbalance among sensor nodes if the optimal path is blocked due to node failures. TinyReg triggers the mechanism to select a cluster-head for the entire network, thereby replacing the dead one and restoring the interrupted communication.

5.5 Experimental results

I simulated the TinyReg and, for comparison the LEACH and SEP protocols. I compared the energy-based metrics (e.g., network lifetime) obtained from our algorithm to other algorithms. All simulations were conducted in randomly generated, static networks. I implemented the protocol in custom code using C++ and the glib-2.0 library. A key performance metric measured in these simulations is the *operational time* defined as the number of rounds for which 50% of the nodes are active. A *round* is a time interval where all the cluster members have to transmit their data to the associated cluster-head.

To evaluate the performance of TinyReg, I simulated a network where nodes are randomly deployed and where there are two types of nodes, as a representative heterogeneous network within a rectangular field. In the network, n node locations are chosen randomly and independently. The sink is located at the centre of the network area. Sensor nodes are heterogeneous in the region and have different energy constraints.

In order to compare the performance of TinyReg to other algorithms, the tunable parameters in TinyReg are set as follows which is the same to the parameters in [39][28][62]:

- E_{Tx} = 50 nJ/bit;

- E_{Rx} = 50 nJ/bit;
- ε_{fs} = 10 $pJ/bit/m^2$;
- ε_{mp} = 0.0013 $pJ/bit/m^4$;
- E_{DA} is the processing cost of a bit report to the sink and E_{DA} = 5 $nJ/bit/$signal;

Initial energy of low energy sensor nodes is $0.5J$ and initial energy of high energy sensor nodes is $1J$. The size of the data packets that a cluster-head node sends to the sink is 512 bytes and the size of the control packets that a sink sends to cluster-head nodes is 16 bytes.

I compared the performance of TinyReg to LEACH and SEP in the same heterogeneous setting. No two sensors can be positioned at the same point on the grid. I simulated TinyReg, LEACH and SEP in a multi-cast environment where all nodes receive a control packet every time the sink sends this packet, once per round.

I consider three performance metrics in the proposed mechanism: the number of alive nodes, the throughput and the energy dissipation. Two scenarios (1 and 2) in [39] are considered as follows:

- Scenario 1: a heterogeneous sensor network in a 100 $\times$100 metre field, where 20% of the nodes are advanced nodes (i.e., $m = 0.2$) equipped with a 100% great energy reservoir than other (normal) nodes (i.e., $a = 1$);
- Scenario 2: a heterogeneous sensor network in a 100 $\times$100 metre field, where 20% of the nodes are advanced nodes (i.e., $m = 0.2$) and equipped with 300% great energy reservoir than other (normal) nodes (i.e., $a = 3$).

The total number of sensors in the network is 100. The proportion of cluster-heads (set to 0.1) are input parameter of the simulation.

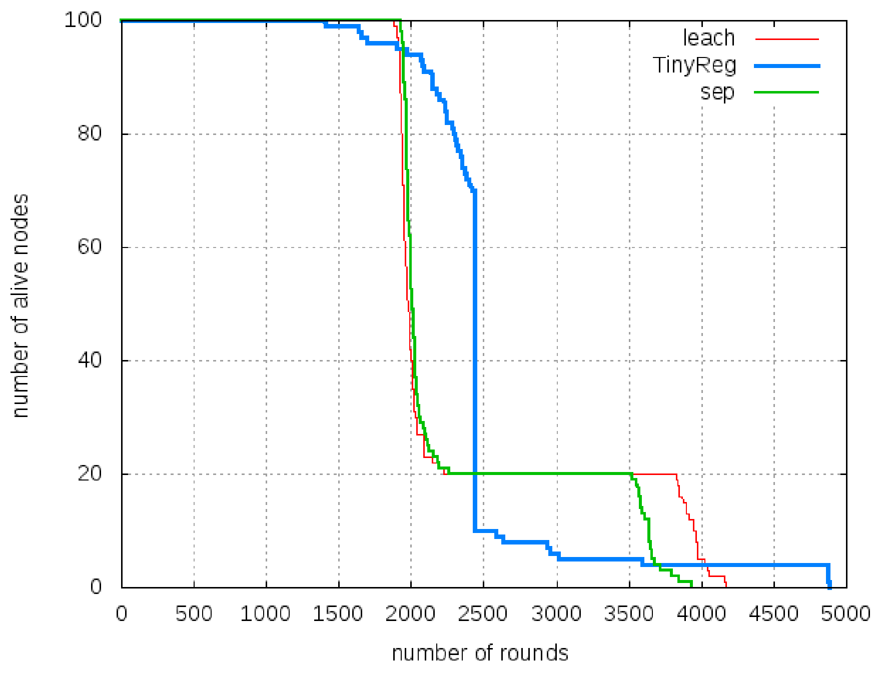


Figure 5.6: *Comparison of the number of alive nodes per round per epoch between LEACH, SEP and TinyReg in the presence of heterogeneity (a=1).*

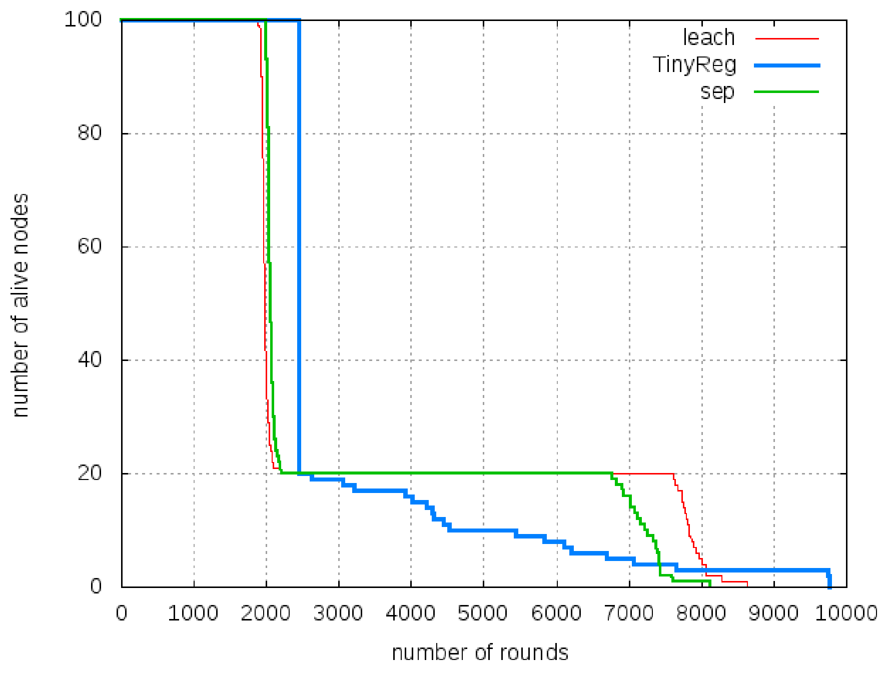


Figure 5.7: *Comparison of the number of alive nodes per round per epoch between LEACH, SEP and TinyReg in the presence of heterogeneity (a=3).*

Table 5.1: *The parameters of MICAZ motes*

Parameter	Value
Node initial energy	3.6J
Traffic interval	0.0969
Traffic packet size	70 bytes

Table 5.2: *Comparison of network lifetime (0% nodes active) in rounds*

Model	Scenario 3	Scenario 4	Scenario 5
LEACH	119414	15771	2171
TinyReg	128568	17576	2437

5.5.1 Number of alive nodes

I compared the performance of TinyReg to LEACH and SEP in terms of the number of alive nodes for the case of $a = 1$ and $a = 3$. Figure 5.6 shows the results for the case of $m = 0.2$ and $a = 1$. Though TinyReg's first two nodes died earlier than with the other two protocols, as seen in Figure 5.6, the remaining nodes of TinyReg live longer than with LEACH and SEP. It can be seen from Figure 5.6 to 5.7 that, in the case of all these algorithms, the network degrades rapidly once 40% of the nodes die. However, this occurs later in the case of TinyReg.

Figure 5.7 shows the results of the simulation of scenario 2 where $m = 0.2$ and $a = 3$. It is obvious that the operational time of TinyReg is extended compared with that of LEACH (by 29%) and SEP (by 22%). Moreover, the lifetime of the network in our method is the longest in all three protocols and the operational time of TinyReg is significantly longer than that of LEACH and SEP.

- Scenario 3: node initial energy (J) is 3.6 J and traffic packet size is 70 Bytes;
- Scenario 4: node initial energy (J) is 3.6 J and traffic packet size is 512 Bytes;
- Scenario 5: node initial energy (J) is 0.5 J and traffic packet size is 70 Bytes.

Table 5.3: *Comparison of operational time (50% nodes active) in rounds*

Model	Scenario 3	Scenario 4	Scenario 5
LEACH	104287	14107	1927
TinyReg	126738	15390	2140

The performance benefits of TinyReg are also evident in homogeneous networks. We verify TinyReg using our simulator based on the parameters of MICAZ sensors in Table 5.1 in [75].

The lifetime in each case is extended through the use of TinyReg as seen in Table 5.2 and Table 5.3. The lifetime performance of LEACH and TinyReg change proportionally as the node initial energy (3.6 J) and traffic packet-size (70 bytes) change from scenario 3 to scenario 5. When the initial energy increases 7.2 times ($3.6J \div 0.5J = 7.2$), the network lifetime increased about 7 times. On the other hand, when traffic packet size decrease from 512 bytes to 70 bytes, the network lifetime decreased about 7 times.

5.5.2 Throughput

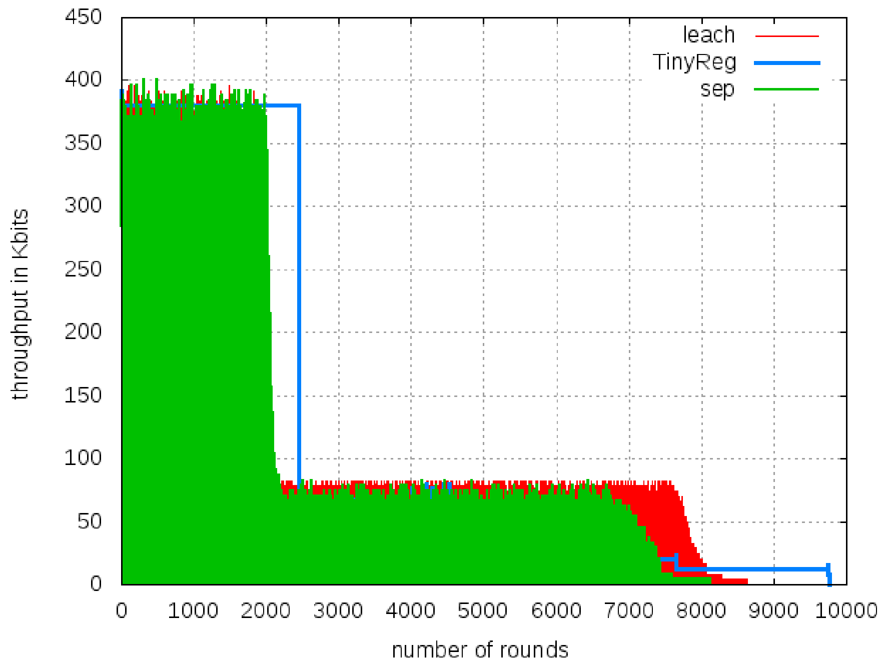


Figure 5.8: *Comparison of throughput from leaves to cluster-heads per round per epoch between LEACH, SEP and TinyReg in the presence of heterogeneity (a=3).*

Next, I compared the performance of our proposed method to LEACH and SEP in terms of throughput. Two types of throughput are considered: the throughput from cluster members to cluster-heads, and the throughput of the network (i.e., the sum of the above).

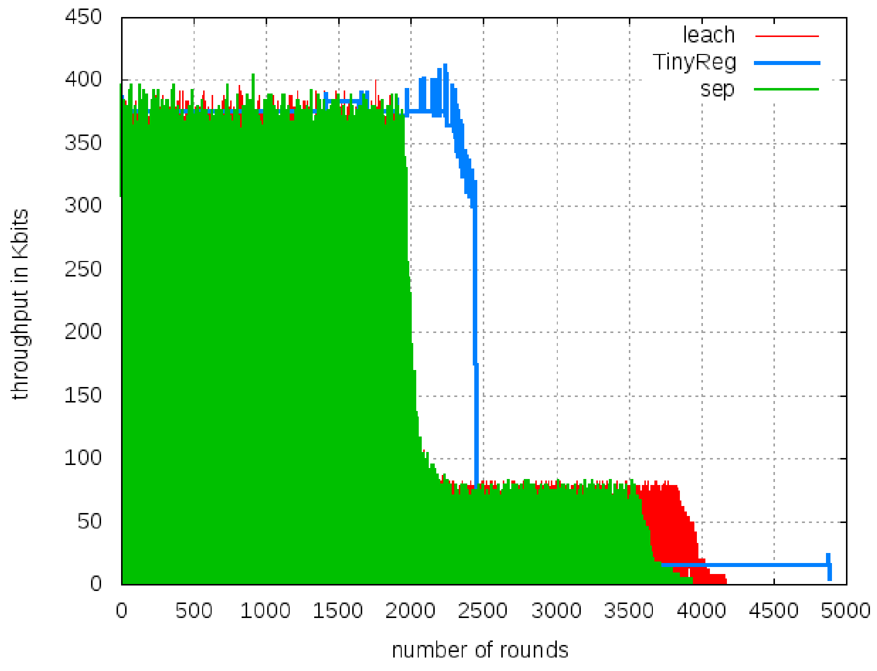


Figure 5.9: *Comparison of throughput from leaves to cluster-heads per round per epoch between LEACH, SEP and TinyReg in the presence of heterogeneity (a=1).*

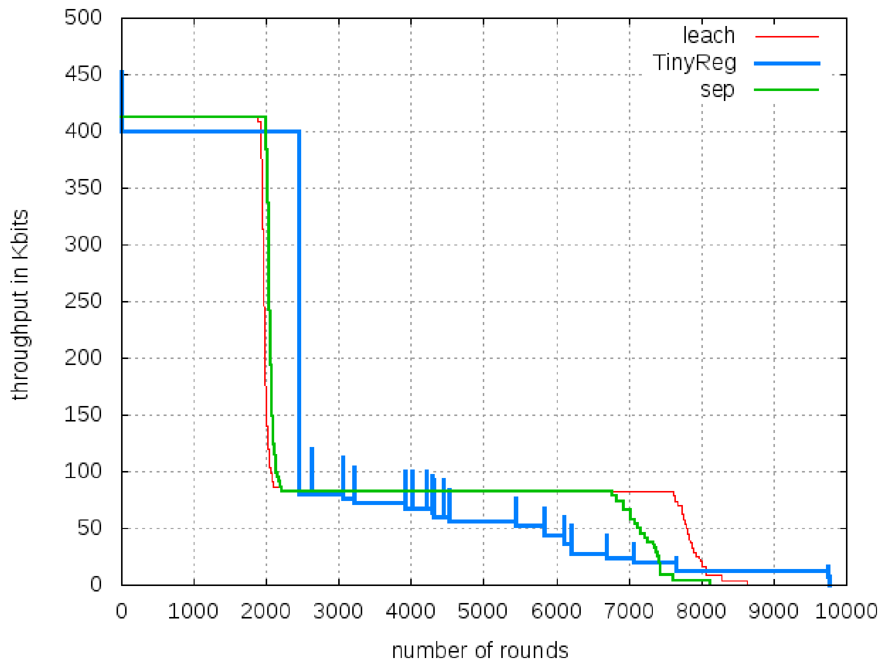


Figure 5.10: *Comparison of throughput of the network per round per epoch between LEACH, SEP and TinyReg in the presence of heterogeneity (a=3).*

The throughput of SEP is larger than that of LEACH and SEP previously reported by Heinzelman *et al* and Smaragdakis *et al*.

Figure 5.8 through 5.9 show respectively the throughput from leaves (i.e., clus-

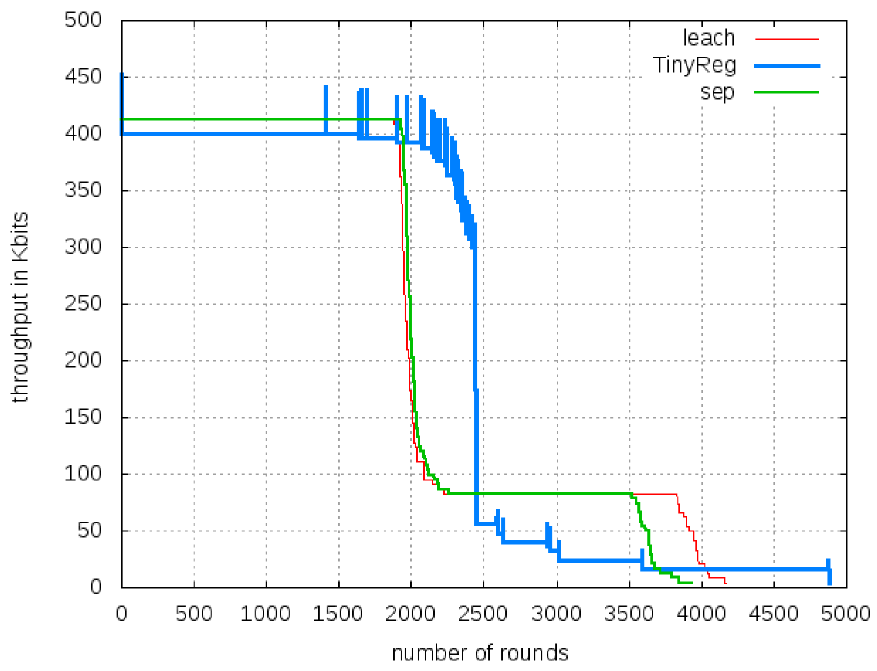


Figure 5.11: *Comparison of throughput of the network per round per epoch between LEACH, SEP and TinyReg in the presence of heterogeneity (a=1).*

ter members) to cluster-heads per round per epoch in scenario 1 and scenario 2. The throughput of LEACH and SEP dramatically fluctuate up and down during the period of every two rounds. This is because 1) the number of cluster-heads in LEACH and SEP is oscillating, 2) the maximum number of cluster-heads is the number of alive nodes in the last round per epoch. Figure 5.8 through 5.9 show that the throughput of nodes to their cluster-heads in TinyReg is greater than those of LEACH and SEP during the operational time.

Figure 5.10 and 5.11 show the throughput of the network per round per epoch in scenario 1 and scenario 2. The throughput of the network of TinyReg is significantly greater than that of SEP and LEACH. Due to multi-hop routing handover, cluster-heads in the case of TinyReg are elected in a more stable fashion during the operational time. The overall throughput of TinyReg is greater than those of LEACH and SEP during the operational time.

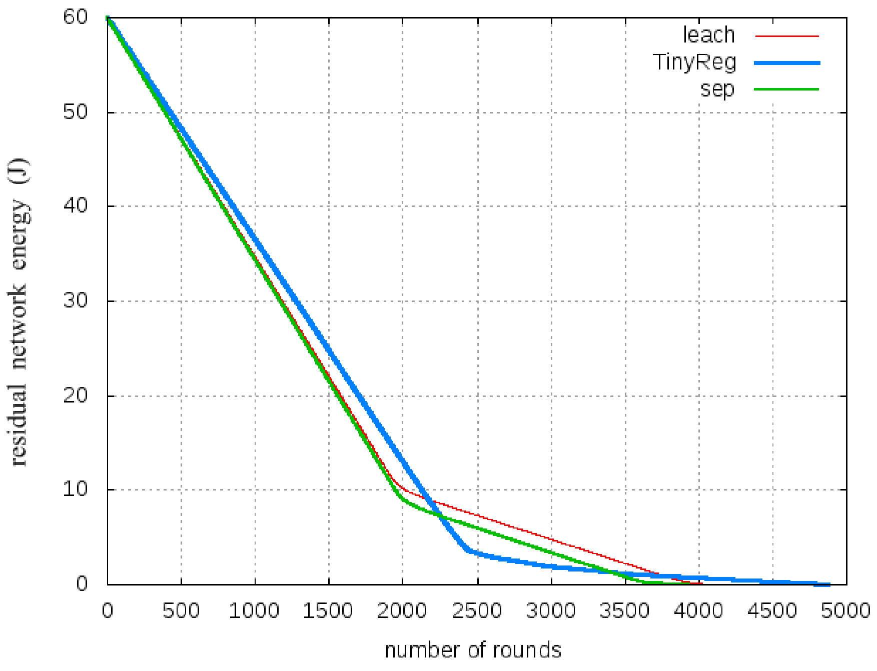


Figure 5.12: *Comparison of residual energy of the network per round per epoch between LEACH, SEP and TinyReg in the presence of heterogeneity (a=1).*

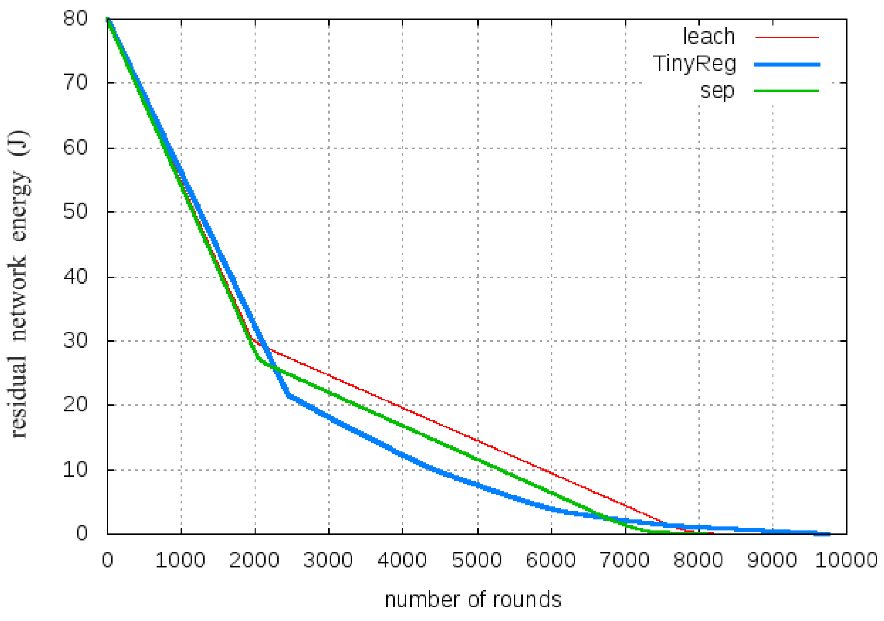


Figure 5.13: *Comparison of residual energy of the network per round per epoch between LEACH, SEP and TinyReg in the presence of heterogeneity (a=3).*

5.5.3 Residual network energy

Figure 5.12 and Figure 5.13 show the residual energy of the network per round per epoch in LEACH, SEP and TinyReg. It clearly indicates the advantages of TinyReg over LEACH and SEP in terms of network lifetime in scenario 1 and scenario 2 (where $a = 1$ and $a = 3$ respectively); and TinyReg achieves a longer network lifetime with the same initial network energy during the operational time.

5.6 Summary

In this chapter, I presented a novel algorithm (called TinyReg) that is especially proposed for use in heterogeneous wireless sensor networks. It combines a regular-graph-based approach to routing with the clustering method. The key idea in TinyReg is to form a communication tree among cluster-heads to the base station so that each cluster-head will receive from and transmit to a close neighbour. The algorithm significantly extends the network lifetime beyond that achieved with existing protocols.

CHAPTER 6

TOPOLOGICAL STABLE ELECTION PROTOCOL (TSEP)

6.1 TSEP review

In this chapter, I discuss a localised routing algorithm called Topological Stable Election Protocol (TSEP) for heterogeneous sensor networks. TSEP combines the K-regular method with routing based on geographical location information.

TSEP uses a distance-based metric value D (the Euclidean distance from node itself to the sink) to calculate and select a potential location in the transmission tree for each cluster-head node on a node-by-node basis. All cluster-heads are elected using a weighted probability (i.e., every node independently generated a random number uniformly distributed in the range 0 and 1; if the number is greater than its threshold, the node is elected as a cluster-head). Thus, it is a distributed, scalable and localised energy balancing strategy. I also use this cluster-head election method in the design of the GAAC algorithm.

In localised routing algorithms [28][38][39], the energy consumption of cluster-heads increases with increasing network scale, since the transmission distance from cluster-heads to the base station is increased as the scale of network is increased. Solutions [28][43], which feature direct communication (with the base station), show serious network lifetime performance degradation when the base station is far away from the nodes. In that case, direct communication will require a large amount of transmit power from each node.

My work differs from other studies in that I consider the optimal energy consumption of different types of nodes and use multi-hop routing and topology control to keep the transmission distance low. This solution makes no assumptions about delay and jitter, or the sleep function. The presented algorithm (TSEP) adopts the metrics and energy constraints of the SEP protocol with additional regular-balanced constraints.

6.2 Implementation of TSEP

TSEP elects cluster-heads using the SEP method [39] where high energy nodes have a higher probability to be elected than that of normal energy nodes.

TSEP aims to reduce the energy consumption of the network and to distribute energy consumption evenly among its nodes so as to extend the network lifetime. This leads to the following design considerations:

1. Direct transmission to the base station should be avoided because in general such transmissions will be over long distances.

2. The number of cluster-heads in the network should determined value with a deviation as small as possible.

There are two types of nodes distinguished by the initial energy level in the node battery: the advanced nodes and the normal nodes. The nodes with a higher initial energy should have a higher probability of being cluster-head nodes because they have much energy reservoir and thus can have longer lifetime than that of the normal nodes.

This motivates the design of an algorithm inspired by SEP but with a different implementation. TSEP generates a hierarchical network graph of cluster-heads where the nodes in all but the lowest tier of the hierarchy have a degree of 4. The value 4 is chosen as a compromise between resilience and energy efficiency. A pseudo-code description of TSEP for the graph construction is as follows (see Algorithm 5):

Algorithm 5 : graph construction algorithm

```
Require: Array $U(1, 2, .., n)$
Ensure: Array $V(1, 2, .., n)$
 1: Compute $distance = |U(1, 2, .., n) - thebasestation|$
 2: $U.d \leftarrow distance$
 3: Sort $[U.d]$ in ascending order
 4: $V(1) \leftarrow U(d_{min1}), V(2) \leftarrow U(d_{min2}), V(3) \leftarrow U(d_{min3})$
 5: $U(1, 2, .., n) \leftarrow (U(1, 2, .., n) - U(d_{min1}) - U(d_{min2}) - U(d_{min3}))$
 6: while ($U \neq \emptyset$)
 7:     For every node $i$ ($i \in U$), compute the distance between $i$ and each node
        in set $V$, save the value of the distance as a distance attribute of $i$
 8:     $V(4) \leftarrow U(d_{min1}), V(5) \leftarrow U(d_{min2}), V(6) \leftarrow U(d_{min3})$
 9:     $U \leftarrow U - U(d_{min1}) - U(d_{min2}) - U(d_{min3})$
10:     For every node $i$ in set $U$, compute the distance between $i$ and each node
        in set $V$, save the value of the distance as $i.d$
11:     $V(7) \leftarrow U(d_{min1}), V(8) \leftarrow U(d_{min2}), V(9) \leftarrow U(d_{min3})$
12:     $U \leftarrow U - U(d_{min1}) - U(d_{min2}) - U(d_{min3})$
13:     ...
14: end while
15: return Array $V$
```

There are two stages in tree formation in a TSEP network. In the first stage, we elect cluster-head nodes from amongst the nodes. TSEP picks sensor nodes as cluster-heads and periodically reassigns these roles to evenly distribute the energy load among the sensors in the network in each round. Once all cluster-head nodes are elected, the first stage of the algorithm is completed. In the second stage of tree formation, the remaining nodes in the network are allocated to the closest clusters based on measuring the transmission distance between them and the cluster-head.

In this second stage, we use the graph construction algorithm (Algorithm 5) to construct a novel cluster-head transmission tree (see Tier 1 in Figure 6.1) where the sink communicates to nodes CH_1, CH_2 and CH_3. CH_1 has the smallest distance (d_{min1}) to the sink, CH_2 has the second smallest distance (d_{min2}) to the sink, etc. Every round TSEP re-elects all cluster-heads in the network. Then, TSEP uses a distance-based metric value D (the distance from node itself to the sink) to calculate and select a potential location in the transmission tree for each node (see Figure 6.1). All cluster-heads communicate via a regular-graph topology.

In Figure 6.1, nodes (1), (2) and (3) in Level 1 are the leaders for Level 2. They will transmit the message to the base station. Since node (1) is in Level 1 in the hierarchy, nodes (4), (5) and (6) which are in Level 2 will aggregate their locally generated data with their received data and send it to node (1). At Level 3, nodes (13), (14) and (15) will similarly combine their current data with that received from the lower level and transmit the resulting message to node (4).

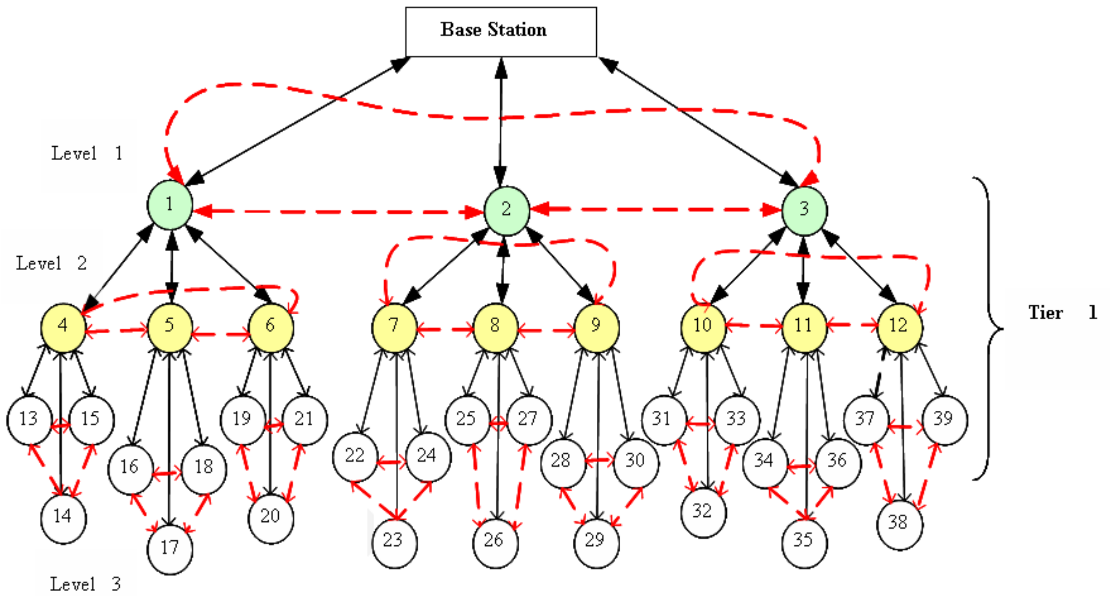


Figure 6.1: *TSEP tree of cluster-heads*

In TSEP, the cluster-heads are selected using a weighted probability. The two types of nodes with different initial energy, have different thresholds, respectively $T(S_{nrm})$ and $T(S_{adv})$. The initial energy of each normal node is E_0 while the energy of each advanced node is $E_0 \cdot a$ ($a \geq 1$). Then every node independently generates a random number per round uniformly distributed in the range 0 to 1.

If this number is greater than its threshold, the node is elected as a cluster-head. The threshold $T(S_{nrm})$ for a normal node to be elected as a cluster-head is:

$$T(S_{nrm}) = \begin{cases} \frac{p_{nrm}}{1-p_{nrm}\times(r \mod \frac{1}{p_{nrm}})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

The threshold $T(S_{adv})$ for an advanced node to be elected as a cluster-head is:

$$T(S_{adv}) = \begin{cases} \frac{p_{adv}}{1-p_{adv}\times(r \mod \frac{1}{p_{adv}})} & \text{if } n \in G \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

where p_{nrm} and p_{adv} is the desired percentage of cluster-heads which is given prior to constructing the network; r is the current round; G is the set of nodes that have not been cluster-heads in the last $\frac{1}{p}$ rounds.

The weighted probability for normal nodes (p_{nrm}) is:

$$p_{nrm} = \frac{\rho}{1 + (a-1)\cdot m} \tag{6.3}$$

The weighted probability for advanced nodes (p_{adv}) is:

$$p_{adv} = \frac{\rho}{1 + (a-1)\cdot m} \times a \tag{6.4}$$

where a is the energy factor between advanced and normal nodes (namely, regular nodes); m is the proportion of advanced nodes in the network; TSEP first assigns a weight to the optimal probability ρ, and ρ is the target ratio of the number of cluster-heads to the total number of nodes.

6.3 Experimental results

I have evaluated the proposed algorithm with different node density and network scale parameters and compared it with LEACH and SEP, using Matlab in randomly generated, static networks. Since LEACH, SEP, generic algorithms and ant colony algorithms are simulated in Matlab, I write my algorithmic codes in Matlab as well.

There are n nodes whose locations in the rectangular field of deployment are chosen randomly and independently. The only sink is in the centre of the network area. A heterogeneous sensor network consists of two types of nodes. I use the remaining energy metrics and geographical transmission distance as the target metrics. Four scenarios are considered:

- Scenario 1: a 300m×300m heterogeneous network with 900 nodes. The node density of the network is 0.01 (900÷(300×300) = 0.01).
- Scenario 2: a 400m×400m heterogeneous network with 1600 nodes. The node density of the network is equal to 0.01.
- Scenario 3: a 300m×300m heterogeneous network with 450 nodes. The node density of the network is 0.005.
- Scenario 4: a 400m×400m heterogeneous network with 800 nodes. The node density of the network is 0.005.

The notation used in all scenarios is as follows:

- Initial network parameters are set: $a = 2$ and $m = 0.2$. This means that advanced nodes are equipped with 200% energy than other (normal) nodes and 20% of all nodes are advanced nodes. Whether or not a given node is advanced is determined at random.
- The proportion of cluster-heads ρ is set to 0.1.
- The characteristics of the energy model are set as follows: E_{Tx} = 50 nJ/bit, E_{Rx} = 50 nJ/bit, E_{DA} = 5 $nJ/bit/$signal, ε_{fs} = 10 $pJ/bit/m^2$, and ε_{mp} = 0.0013 $pJ/bit/m^4$. These parameters are set to common values as shown in many literatures such as [31][39][62].
- The initial energy of nodes are set for normal nodes at $0.5J$ and for advanced nodes at $1J$.

- The size of the data packets L that a cluster-head node sends to the sink is set to 512 bytes.
- The *stable time* and the *operational time* are defined as the number of rounds for which 100% of the nodes and 50% of nodes respectively are active in the network. A *round* is defined as a time interval where all the cluster members have to transmit their data to the associated cluster-head.

6.3.1 Number of alive nodes

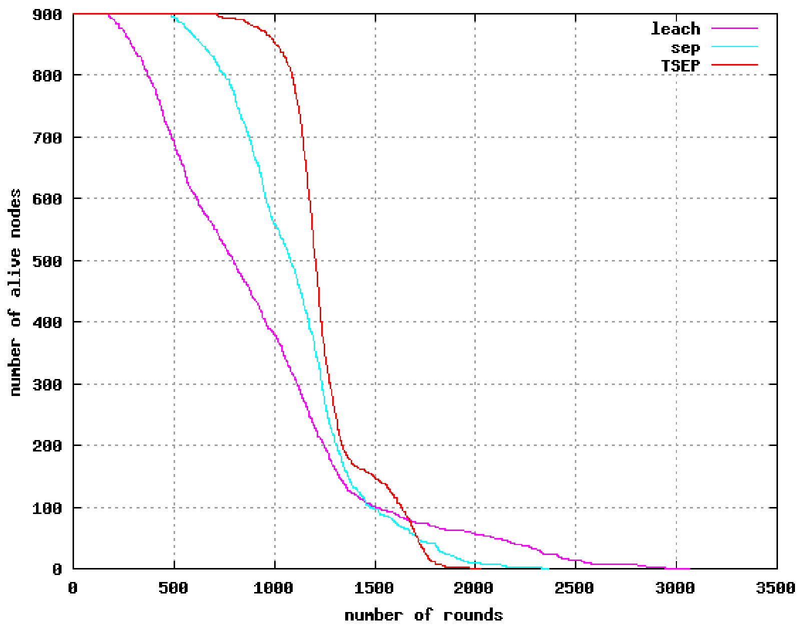


Figure 6.2: *Comparison of the number of alive nodes between LEACH, SEP and TSEP in scenario 1.*

I compared the number of alive nodes of TSEP to LEACH and SEP for all of the above scenarios. Figure 6.2 through Figure 6.5 show that the stable time and the operational time in TSEP is the longest of all three protocols. The stable time is crucial for many applications where feedback from a sensor network must be reliable.

Although the time that no node active in TSEP is shorter than that of LEACH and SEP, the descending curve in TSEP is steeper than that of LEACH and SEP.

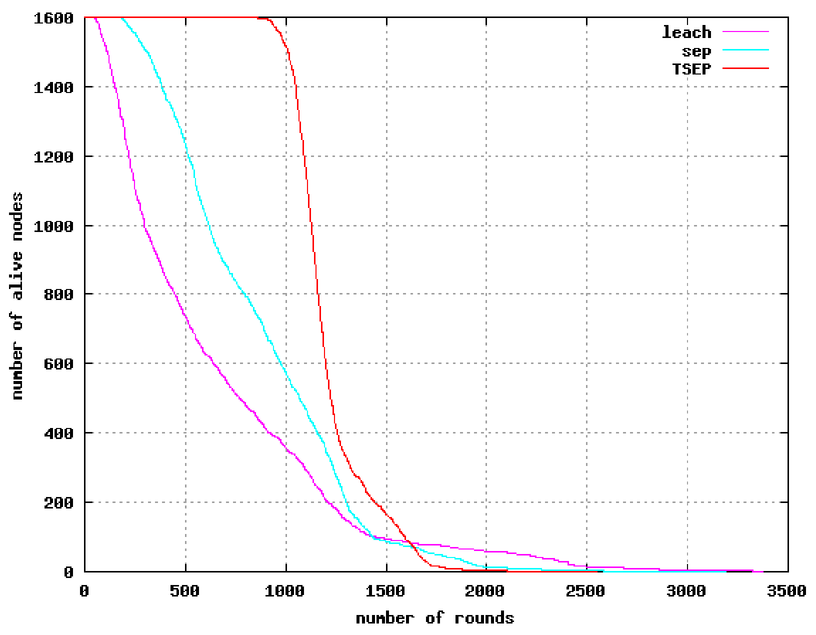


Figure 6.3: *Comparison of the number of alive nodes between LEACH, SEP and TSEP in scenario 2.*

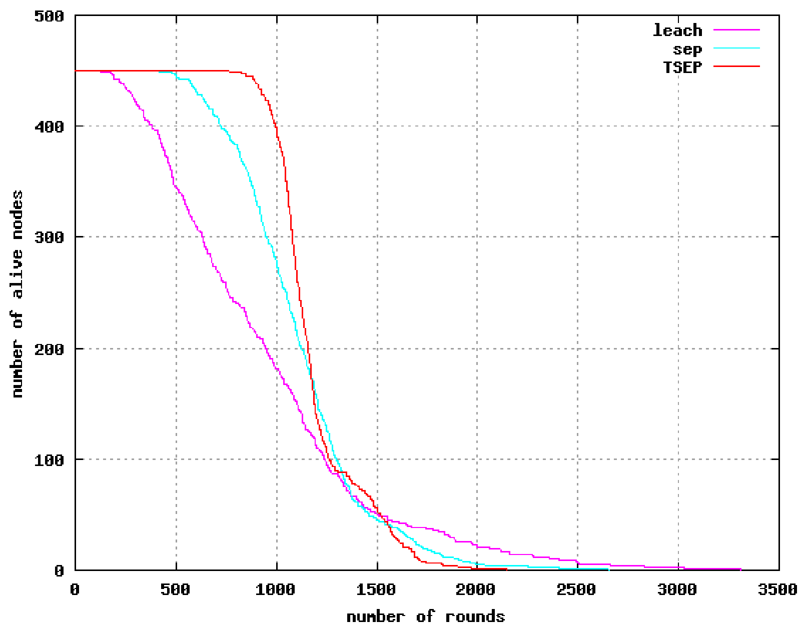


Figure 6.4: *Comparison of the number of alive nodes between LEACH, SEP and TSEP in scenario 3.*

This is because TSEP balances energy consumption, and all nodes will fail at roughly the same time.

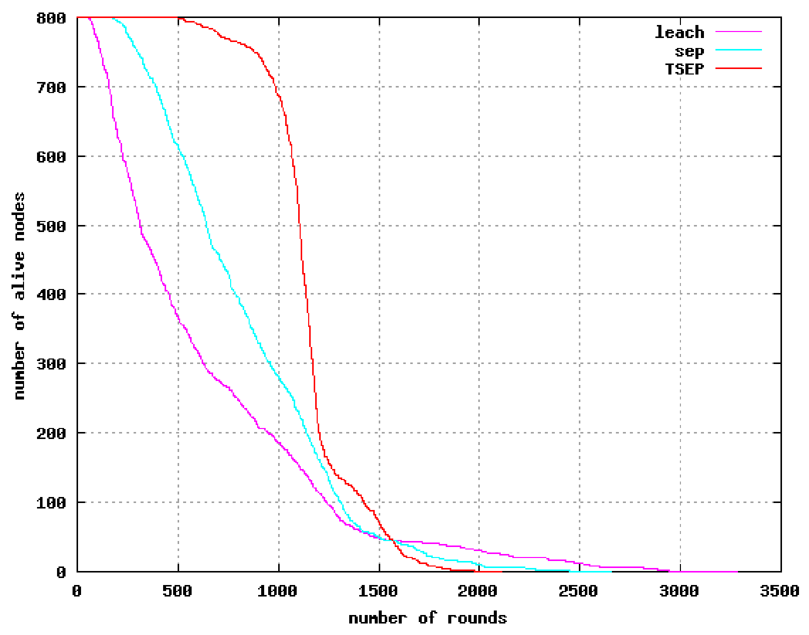


Figure 6.5: *Comparison of the number of alive nodes between LEACH, SEP and TSEP in scenario 4.*

6.3.2 Transmission distance

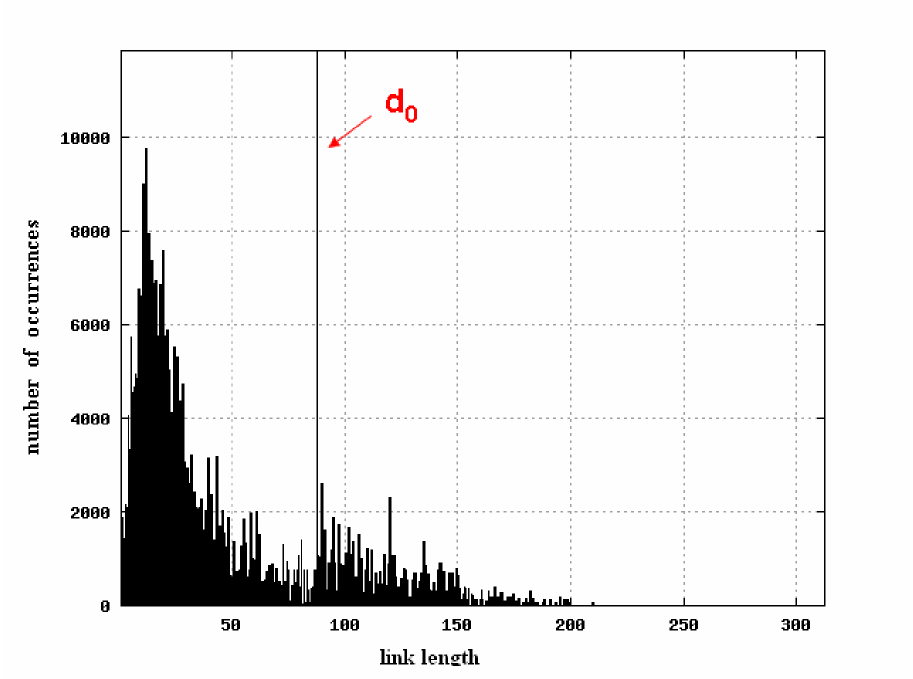


Figure 6.6: *Distance (in metre) distribution between transmitter node and receiver node in the network in LEACH in scenario 3.*

Figures 6.6 through Figure 6.8 show the key advantage of TSEP: due to the use

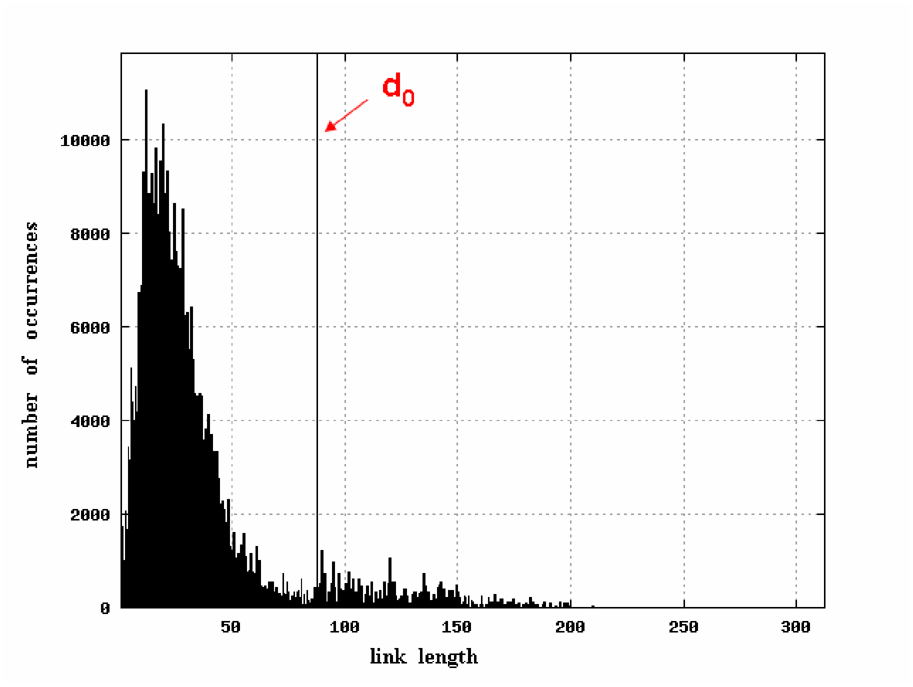


Figure 6.7: *Distance (in metre) distribution between transmitter node and receiver node in the network in SEP in scenario 3.*

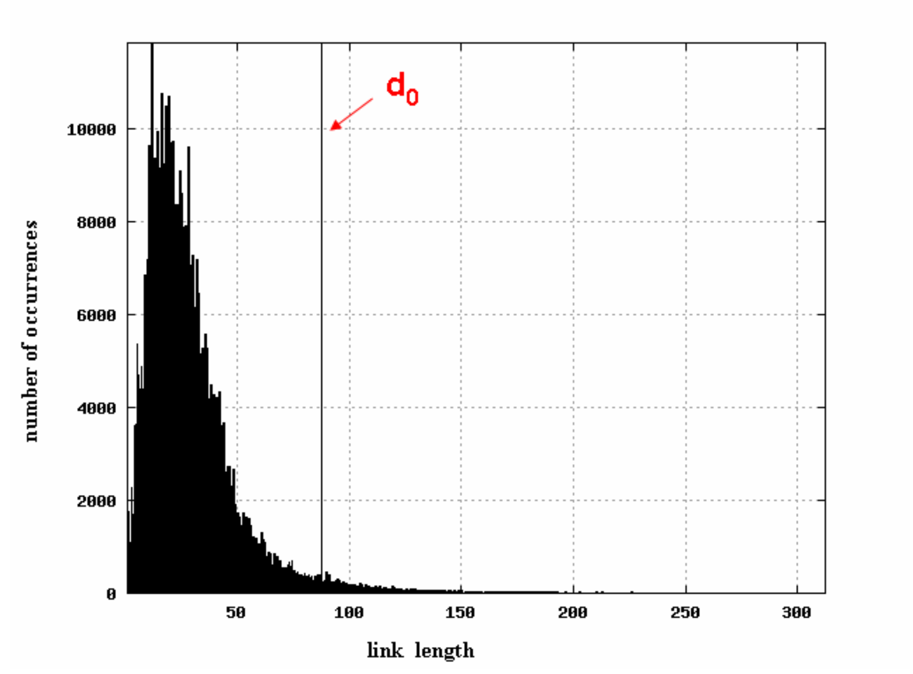


Figure 6.8: *Distance (in metre) distribution between transmitter node and receiver node in the network in TSEP in scenario 3.*

of multi-hop routing in TSEP, the transmission distances between cluster-heads in the case of TSEP are significantly lower on average than in SEP and LEACH. The black vertical line in these figures indicates the critical value of link length, i.e.,

d_0. Link lengths shorter than this will incur significantly less energy consumption (as is clear from Equation (5.1), which was previously described in Section 4.2). Relatively few links exceed d_0 in length, so that our algorithm will be more energy efficient than existing ones.

Figures 6.6 through Figure 6.8 compare the distribution of link lengths for TSEP, LEACH and SEP in an identical scenario. The distribution of link lengths means the frequency of occurrence of inter-node distances for all pairs of directly communicating nodes in the network. For example, in Figure 6.6, there are 1487 links where the inter-node distance is 136 metres.

6.3.3 Energy consumption

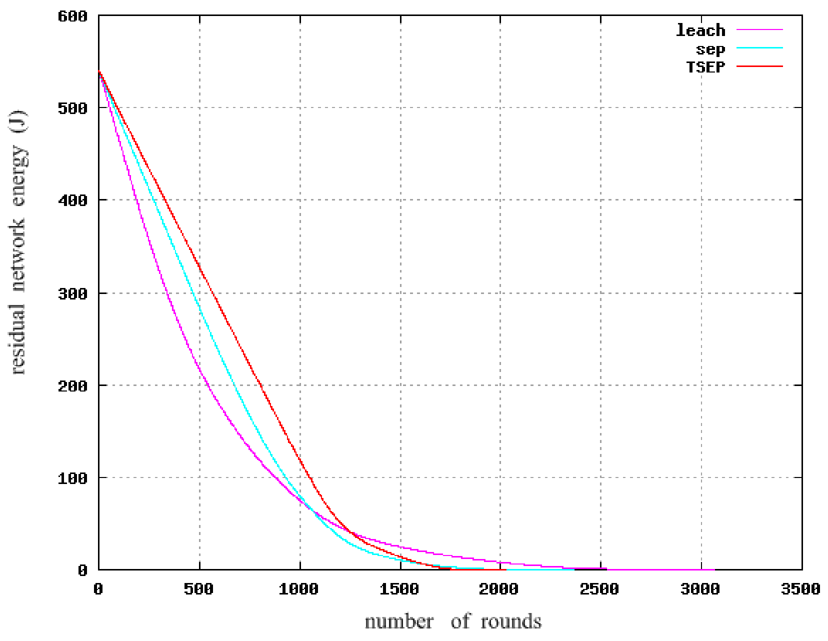


Figure 6.9: *Residual network energy as a function of the number of rounds in scenario 1.*

Figure 6.9 through Figure 6.12 show the per-round energy consumption of the network. In all cases, TSEP has a higher level of remaining energy in the network compared to LEACH and SEP during the stable time and the operational time.

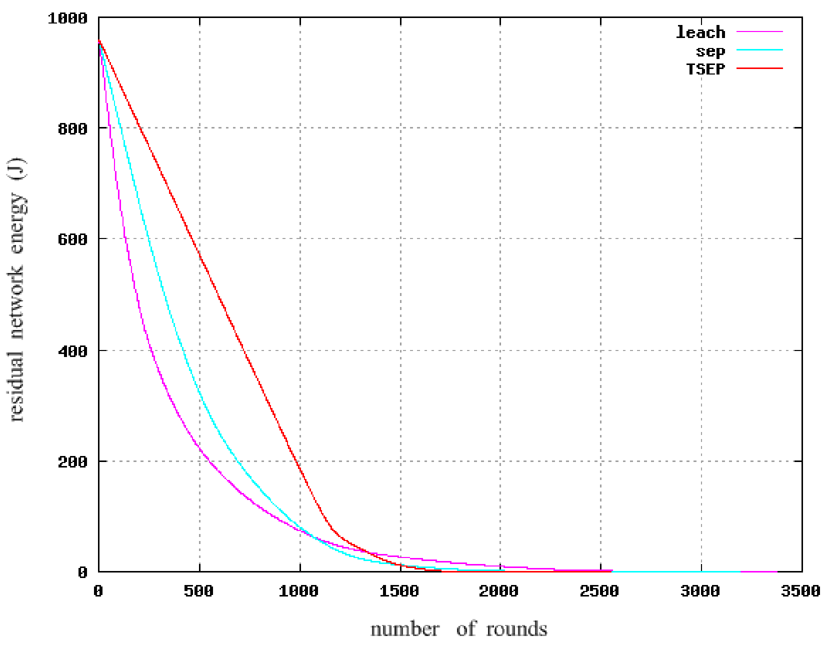


Figure 6.10: *Residual network energy as a function of the number of rounds in scenario 2.*

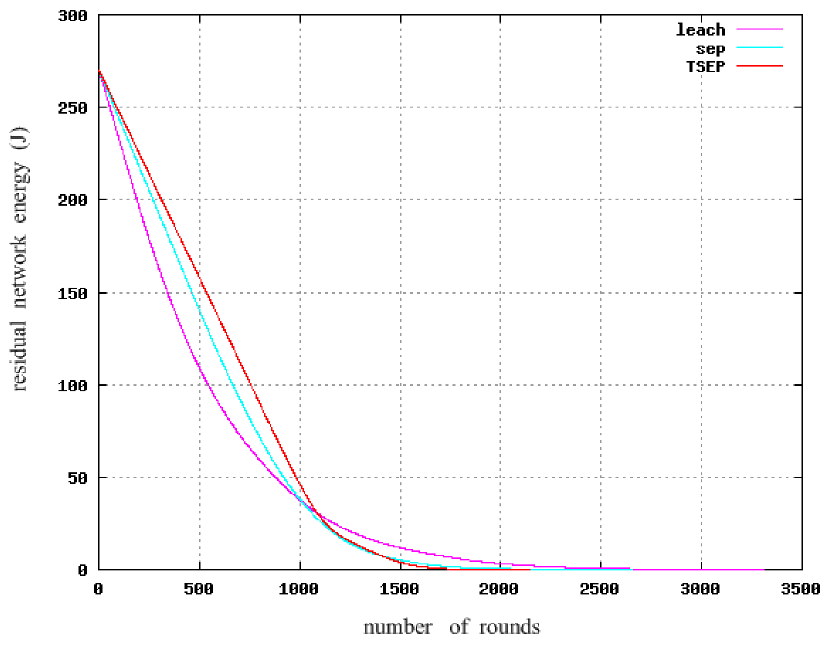


Figure 6.11: *Residual network energy as a function of the number of rounds in scenario 3.*

6.4 Summary

TSEP is proposed as an energy-efficient routing algorithm for heterogeneous networks. TSEP does not require any global knowledge of energy at every elec-

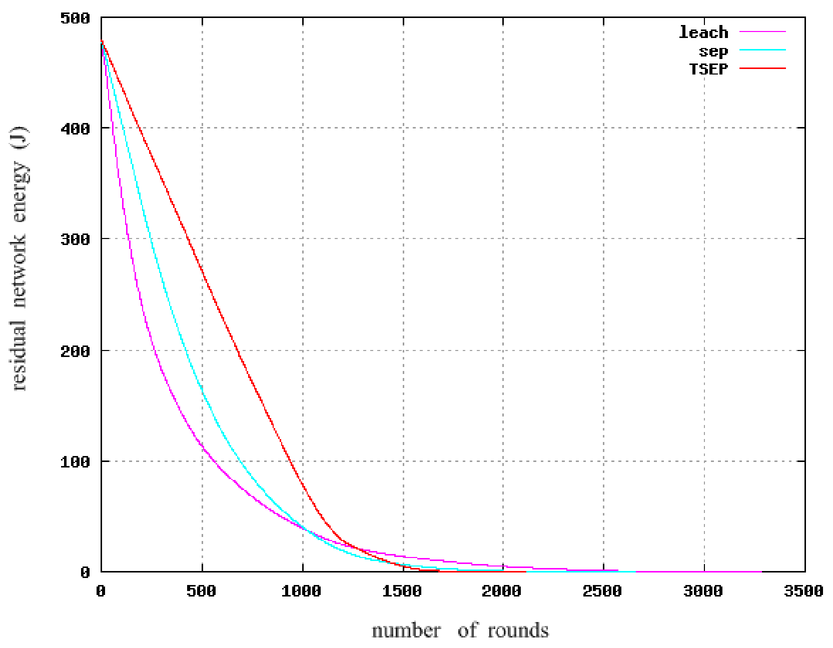


Figure 6.12: *Residual network energy as a function of the number of rounds in scenario 4.*

tion round. It keeps transmission distances low in order to extend the lifetime of cluster-head nodes. The vast majority of link lengths are kept shorter than the cross-over distance d_0, the critical distance whose significance is discussed in Section 4.2.

In the next chapter, we will introduce an evolutionary routing algorithm GAAC which seeks to emulate human intelligence in selecting the best route.

CHAPTER 7

IMPROVED EVOLUTIONARY ALGORITHM

In this chapter, an energy-efficient evolutionary algorithm (GAAC) for heterogeneous sensor networks is proposed. It combines the genetic algorithms and ant colony algorithms to solve the multi-constraint optimal routing problems in sensor networks.

7.1 Introduction

Yao *et al.* [59] used genetic algorithms and ant colony algorithms to improve the searching performance in speed and accuracy in virtual enterprise partner selection. Figure 7.1 ([59]) shows the difference between the speed-time curves of the genetic algorithm (GA) and ant colony algorithm (ACO). In the early search stages between 0 to time t, the genetic algorithm has a higher rate of convergence to the optimal solution. ACO's speed of convergence improves rapidly because the feedback information's pheromone are applied into its route selection scheme after time t. However, they don't identify the best integration time for starting the

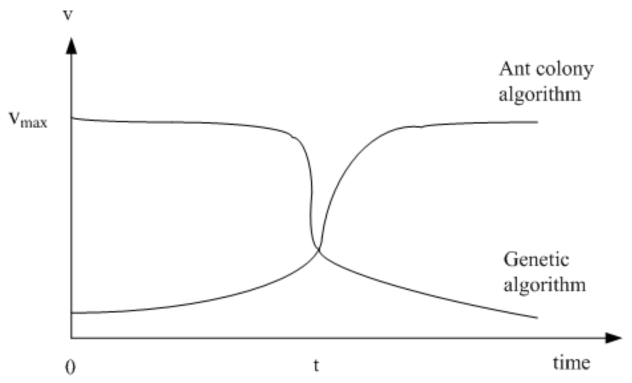


Figure 7.1: *Comparison of the convergence speed of GA and ACO algorithms*

ant colony algorithm and ending the genetic algorithm.

Cluster-heads experience high energy consumption and exhaust their energy resources more quickly than do other nodes [76]. Previous work in literatures [51][57][58][59] consider how to use GA to generate cluster-heads and clusters, thereby minimising total energy consumption. This would result in a routing algorithm with low flexibility and high memory requirements. On the other hand, since GA is a centralised algorithm [57], the GA algorithm complexity increases when the number of nodes increases.

The work described below differs from the above studies in that I consider the optimal energy consumption of different types of nodes using genetic algorithms, ant colony algorithms and clustering methods to keep the transmission distance low. Dead loops are avoided since spanning tree algorithms are used to guarantee that no circles in the network. The goal of this work is to design a heuristic routing algorithm which addresses the tradeoff between algorithm complexity and transmitted energy for heterogeneous networks.

Algorithm 6 : the pseudo code of GAAC

```
1: Encode the solution into chromosomes (binary strings) in step 1, define Objective function $E$, initialise initial probabilities of crossover ($p_c$) and mutation ($p_m$), pheromone evaporation rate $b$.
2: Generate the initial population
3:     Elect cluster-heads using methods in Step 2
4:     The rest of nodes in the network will be allocated to the closest clusters based on measuring transmission distance between them and the cluster-head
5:     **while** ($t <$ Max number of generations)
6:         Generation new solution by crossover and mutation
7:             if $p_c > rand \in (0, 1]$, $crossover(x)$; end if
8:             if $p_m > rand$, $mutation(x)$; end if
9:         Accept the new solutions if the function value decreases
10:        Select the current best for new generation (elitism)
11:    **end while**
12: Decode the results to mark the best route of ACOs with pheromone $\tau_{ij}$
13: **for** loop over all nodes
14:     Compute the best solution using methods in Step 4
15: **end for**
16: Output the best solutions and pheromone distribution
```

7.2 Implementation of GAAC

The algorithm is shown in Algorithm 6. As the beginning the algorithm, the control parameters including E, b, p_c and p_m are given. I design an energy consumption function (E), which is the energy consumed in all nodes in the network. GAAC chooses the best route which is with the minimal value of E. b is the evaporate coefficient of the pheromone. The definitions of *crossover* operation and *mutation* operation are presented in Section 2.5. The probabilities of p_c and p_m are predefined parameters. A bernoulli random variable with parameter p_c is used to determine whether the crossover operation happening; and so does p_m.

All nodes elect themselves as cluster-heads, and "advanced nodes" (having higher energy levels than the normal nodes) have preference to become cluster-heads. This allows the algorithm to be less centralised.

The new algorithm retains the merits of genetic algorithms, namely a higher rate of convergence to the optimal solution, and of ant colony algorithms. The primary features of GAAC are: 1) a novel clustering routing method based on genetic algorithms and ant colony algorithms for heterogeneous networks; 2) an improved generating strategy for the initial population which consists of spanning tree algorithms.

It assumed that a heterogenous network consists of two types of nodes with different initial energy, normal nodes and advanced nodes. It can be extended into three types of nodes, four types of nodes, etc. There are three roles performed by nodes in the network: cluster-heads, normal nodes and one sink.

There are four steps in the operation of the GAAC algorithm, as documented below:

Step 1 Genetic encoding. A chromosome of the proposed GAAC consists of sequences of positive integers that represent the IDs of nodes. A routing path passes through these nodes. Each sensor node is represented by a 1-bit binary number as follows:

0 - the sensor is dead; 1 - the sensor is alive. E.g., the sequence

```
s1 s2 s3 s4 s5 s6 s7
 1  1  1  0  0  1  0
```

indicates that sensors s4, s5, s7 are dead while sensors s1, s2, s3, s6 are alive.

Step 2 Initialisation clusterings. In GAAC, each node decides whether or not to become a cluster-head for the current round. The solution makes no assumptions about sleep status, delay and jitter. The algorithm retains the cluster-head selection schedule proposed by Smaragdakis *et al* in [39]. According to [39], an $epoch$ of the clustered sensor network is defined as the number of rounds ($\frac{1}{p}$) (where p is the weighted probability to obtain the threshold T that is used to elect the cluster-head in each round). This decision is made by each node $i \in G$ (where G is the set of nodes that have not become cluster-heads) independently at the beginning of each round. A bernoulli random variable with parameter T is used to determine whether the node becomes a cluster-head for the current round.

$$T = \begin{cases} \frac{p}{1-p\times(r \bmod (\frac{1}{p}))}, & \text{if } i \in G \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

where p is the weighted probability to obtain the threshold $T(i)$ that is used to elect the cluster-head in each round; r is the current round; G is the set of nodes that have not become cluster-heads within the last $\frac{1}{p}$ rounds of the epoch. The probability (p) of node i is set as:

$$p = \begin{cases} \rho/(1+(a-1)\cdot m), & \text{if } i \in G_N \\ a\cdot\rho/(1+(a-1)\cdot m), & \text{if } i \in G_A \end{cases} \tag{7.2}$$

where ρ is the desired percentage of cluster-heads which is determined a priori; m is the proportion of advanced nodes in the network; a is the energy factor between advanced and normal nodes. For example, the initial energy of each normal node is E_0 while the energy of each advanced node is $E_0 \cdot a$ (where $a \geq 1$). G_N is the set of normal nodes and G_A is the set of advanced nodes.

In a heterogeneous environment, each normal node will become a cluster-head exactly once every $\frac{1}{\rho}\cdot(1+(a-1)\cdot m)$ rounds per epoch. For advanced nodes, it will become a cluster-head exactly once every $\frac{1}{\rho}\cdot\frac{(1+(a-1)\cdot m)}{a}$ rounds. This period is defined as a *subepoch* in [39]. It is clear that each epoch has a subepochs for advanced nodes. Therefore, an advanced node becomes a cluster-head exactly a times more frequently than a normal node within an epoch.

After all cluster-heads are determined, every cluster-head chooses its close neighbours as its cluster members. When all clusters are decided, the GAAC generates the initial genetic population using spanning tree algorithms. The new genetic population is generated as far Step 3 and 4 below.

Step 3 Objective function value calculation. The objective function (E) is used for evaluating whether this solution (namely, a chromosome) is better than other solutions.

The objective function (E) in GAAC is calculated as:

$$E = \sum_{i \in C} (E_i + \sum_{j \in CT_i} E_j) \tag{7.3}$$

where C is the set of cluster-heads and CT_i is the set of members in cluster i, E_i is the energy consumed by cluster-head i in transferring a message to its cluster-head at the next level of the hierarchy. This energy is calculated as $E_i = E_i^t + E_i^r$ where E_i^t is the transmission energy consumption of node i, E_i^r is the energy consumption for receiving messages of node i (i=1,2,..C_k), C_k is the number of cluster-heads in the network. $\sum_{j \in CT_i}(E_j)$ is the energy consumption for sending and receiving messages between cluster-head i and its members. The route function (E) represents the energy consumption of all nodes in the network. To find a good solution, GAAC tries to minimise the function and thus the energy consummation.

Step 4 New population. To produce a new population, GAAC uses three operations: node selection, crossover and mutation. Node selection is the identification of cluster-heads and member nodes. The crossover exchanges one solutions's two partial chromosomes (partial-routes); the mutation introduces new partial chromosomes by choosing one gene locus in the individual string and reverses the value with the mutation probability p_m.

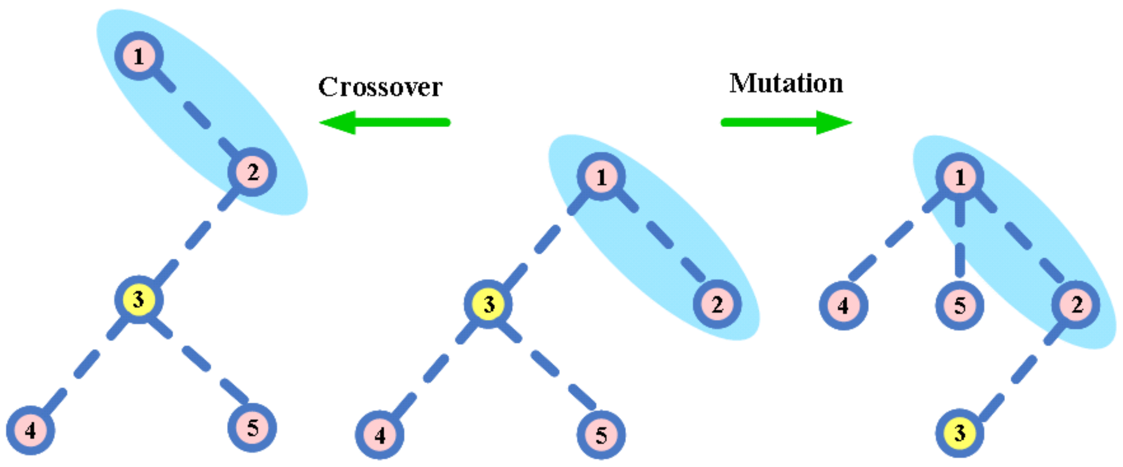


Figure 7.2: *Crossover and mutation operations of the spanning tree.*

Sometimes applying the mutation operation on the individuals leads to the algorithm being interrupted because of dead loops or infeasible chromosomes. To avoid this situation happening, a spanning tree of all nodes is computed; to further improve searching accuracy and efficiency, we implement crossover operation and mutation operations which work on a cluster-head tree. Figure 7.2 indicates how crossover and mutation operations work on a cluster-head tree where node 3 is the crossover point and mutation point. In the crossover operation, the solution exchanges its three portions (nodes 3, 4 and 5) in the tree (a portion shows the node location in the tree) and get a new solution; in the mutation operation, the solution exchanges one portion (node 3) to a new location to get a new solution.

The crossover and mutation operations are applied to each bit of an individual with a probability of crossover rate of p_c and a probability of mutation rate of p_m. After the three steps (step 2, 3 and 4), the best solution is produced. Then the candidates produced by the genetic algorithm are used as the seeds of the pheromone to update the initial pheromone parameter τ_{ij} in Equation (7.4).

The algorithm makes two passes. The first one pass searches for the possible route. For every node, there are m ants that move forward to the destination. The second pass records the shortest route to the destination. These ants reverse route from the destination to source and updates the node potential which records the shortest route to the destination.

In the reverse pass, the pheromone trails of GAAC for the optimal solution path are updated as follows:

$$\tau_{ij} = (1-b) \cdot \tau_{ij} + \Delta\tau_{ij} \tag{7.4}$$

where τ_{ij} is the pheromone information for the route between i and j; b is the evaporate coefficient of the pheromone; $\Delta\tau_{ij}$ is the increment of pheromone τ_{ij}.

If the ant travelled the route between node i and node j, then the pheromone will be evaporated according to Equation (7.5) as follows:

$$\Delta\tau_{ij} = \sum_{k=1}^{ma} \frac{1}{L_k} \tag{7.5}$$

where ma is the number of ants in the current location; L_k is the length of the path from the current node to the sink that ant k has token. For other nodes not visited in the first pass, $\Delta\tau_{ij}$ is equal to 0, thus, the pheromone evaporates more rapidly.

The ant located at node i hops to node j, selected from among the neighbours that have not yet been visited in the ant colony algorithm of GAAC, with probability p_{ij} where

$$p_{i,j} = \begin{cases} \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{s\in G_i}\tau_{is}^{\alpha}\eta_{is}^{\beta}}, & \text{if } j \in G_i \\ 0, & \text{otherwise} \end{cases} \tag{7.6}$$

where τ_{ij} is the pheromone information for the route between i and j; $\eta_{ij} = \frac{1}{d_{ij}}$, d_{ij} is the distance between node i and node j. The parameters α and β are strictly positive and dictate the relative influence of the pheromone and distance on the probability of p_{ij}. G_i indicates the set of neighbours of node i which can be chosen as the next hop.

Each ant updates the pheromone trails and node potential in the reverse path using Equation (7.4).

The termination condition. Repeat Step 3 and Step 4 until the variation of the function value between the current solution and the best solution is less than 0.001 or the predefined maximum number of iterations is met, output the best solution.

7.3 Experimental results

I evaluate the proposed algorithm in Matlab in randomly generated networks. Existing open source codes of genetic algorithms and ant colony algorithms and the protocols of LEACH and SEP are all written in Matlab, thus I write my code in Matlab as well so that it is convenient to compare them with the proposed algorithm.

The scenario is a square heterogeneous network which consists of 100 nodes. The locations of nodes are chosen randomly and independently in the field of deployment, and the sink is at location (0,0) in the field. The initial energy of normal nodes is $0.5J$ and that of advanced nodes is $1.5J$. The proportion of cluster-heads ρ is set to 0.2, m = 3, E_{Tx} = 50 nJ/bit, E_{Rx} = 50 nJ/bit, E_{DA} = 5 $nJ/bit/$signal, ε_{fs} = 10 $pJ/bit/m^2$, and ε_{mp} = 0.0013 $pJ/bit/m^4$. The size of the data packets L that a cluster-head node sends to the sink is 512 bytes. p_c = 0.95, p_m = 0.05, the population size is set to 20, ma = 5, ρ = 0.85, α= β= 2.

I compared the performances of GAAC to LEACH [28], SEP [39], GA, and ACO in terms of the number of alive nodes and the energy consumption for $100m \times 100m$ network.

7.3.1 Convergence speed

The completion is defined as finding the optimal solutions. I compared the convergence speed of the sub-algorithms in the proposed algorithm. The results are shown in Figure 7.3 through Figure 7.4.

For various protocols in my experiment, the maximum number of iterations is set to 100 in each round. As I observed, the network lifetime for various protocols in my experiment is more than 3000 rounds (see Figure 7.10). Then, the total number of iterations for various protocols is more than 300,000 times in all scenarios.

Clearly, the genetic algorithm converges more rapidly than that of the ant colony algorithm in the early search rounds in Figure 7.4; after that round, the ant

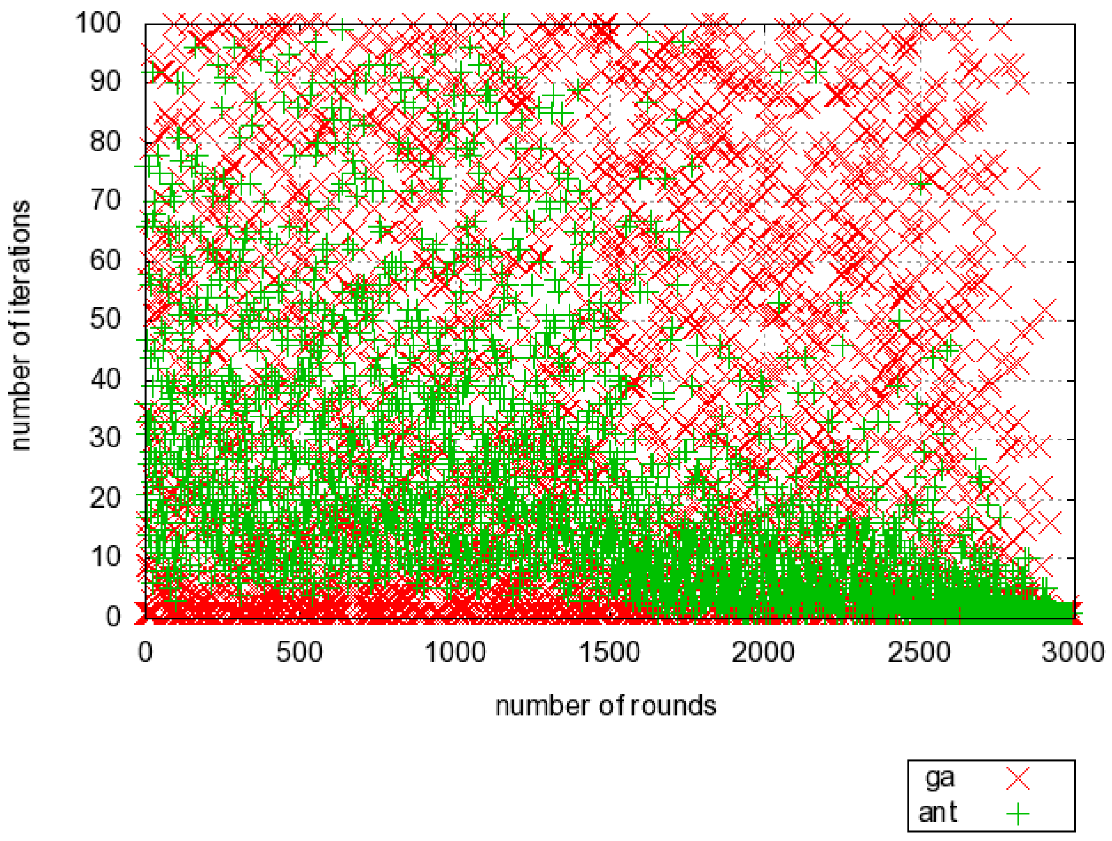


Figure 7.3: *Comparison of the number of iterations for finding optimal routing solutions of GAAC's ant colony sub-algorithm and genetic sub-algorithm.*

algorithm converges more rapidly than the genetic algorithm through Figure 7.3 to Figure 7.4. This feature explains that why I combine the genetic algorithm and ant colony algorithm to the GAAC protocol (i.e., in order to converge rapidly).

7.3.2 Number of alive nodes

Figure 7.5 shows that the stable time (the time until the first node dies) and the time until 60% of nodes die is the longest in GAAC of the four protocols. The stable time is crucial for many applications where feedback from a sensor network must be reliable. The 60% mortality level metric was chosen because simulations indicate that networks where the number of dead nodes exceed 60% of the original population are typically very inefficient. Although network connectivity may be maintained at such a mortality level, this will be at the expense of increased energy consumption as the average link length rises.

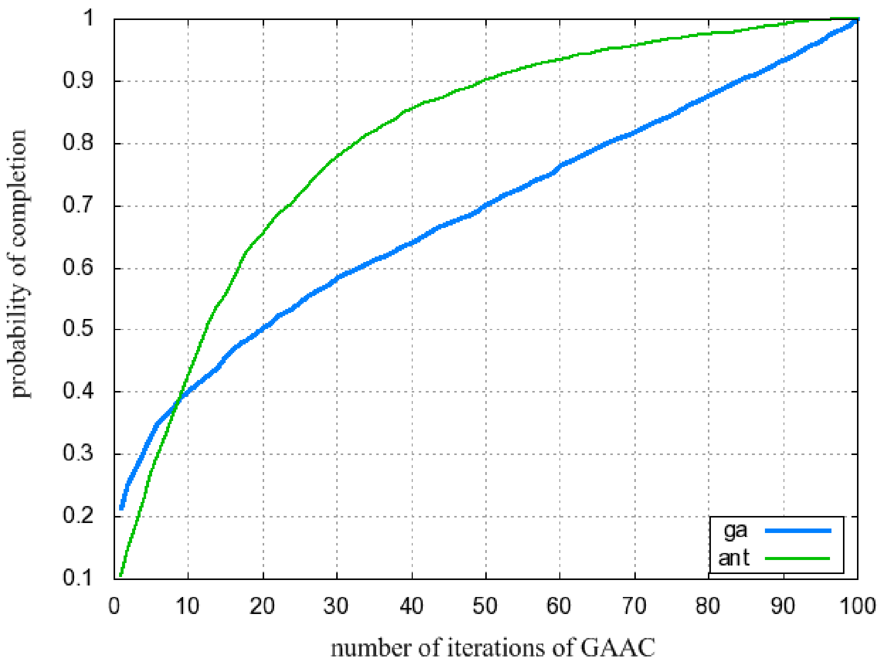


Figure 7.4: *Comparison of the probability of the completion for GAAC's genetic algorithm and GAAC's ant colony algorithm.*

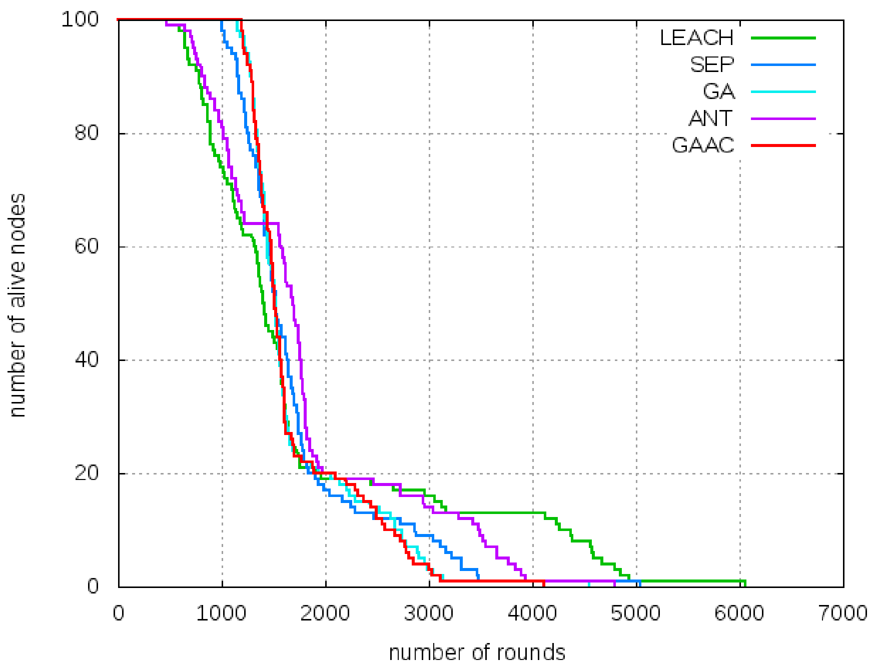


Figure 7.5: *Comparison of the number of alive nodes.*

In the interval from round 0 to round 2000, the GAAC and GA methods have similar performance regarding the number of alive nodes. This is to be expected as they both run a genetic algorithm during this phase of operation. Similar observations apply in Figures 7.6 through 7.10. After round 2000, the convergence speed of GA drops compared with that of the ACO algorithm, and so the number of alive nodes in GAAC between rounds 2000 and 2500 is larger than in the case of GA. After round 2500, although the network lifetime for GAAC is shorter than with other protocols, the descending curve in GAAC as the network fails is steeper than with other protocols. This shows that GAAC is achieving the goal of balancing energy consumption, so that all nodes will fail at roughly the same time.

It shows that GAAC always prolongs the stable time (the time until the first node dies) and the time until 60% nodes die in GAAC is longer than compared to other protocols as well. I found that GAAC yields longer stability region for higher values of extra energy brought by more powerful nodes.

7.3.3 Energy consumption

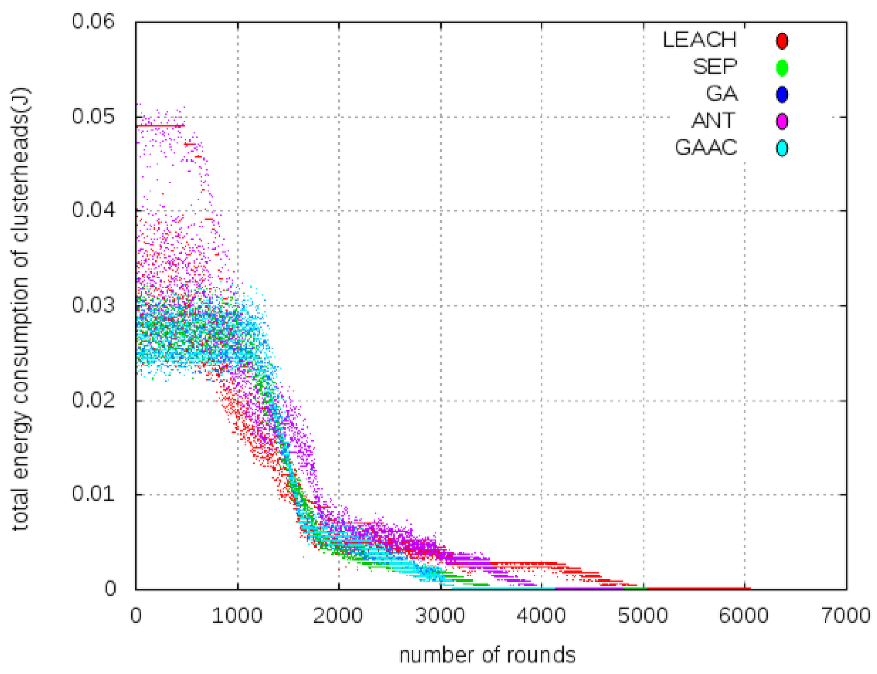


Figure 7.6: *Comparison of the energy consumption of cluster-heads for various protocols.*

Figures 7.7 and 7.9 show the average energy consumption of cluster-heads and the network for the various protocols. Figures 7.6 and 7.8 show the energy

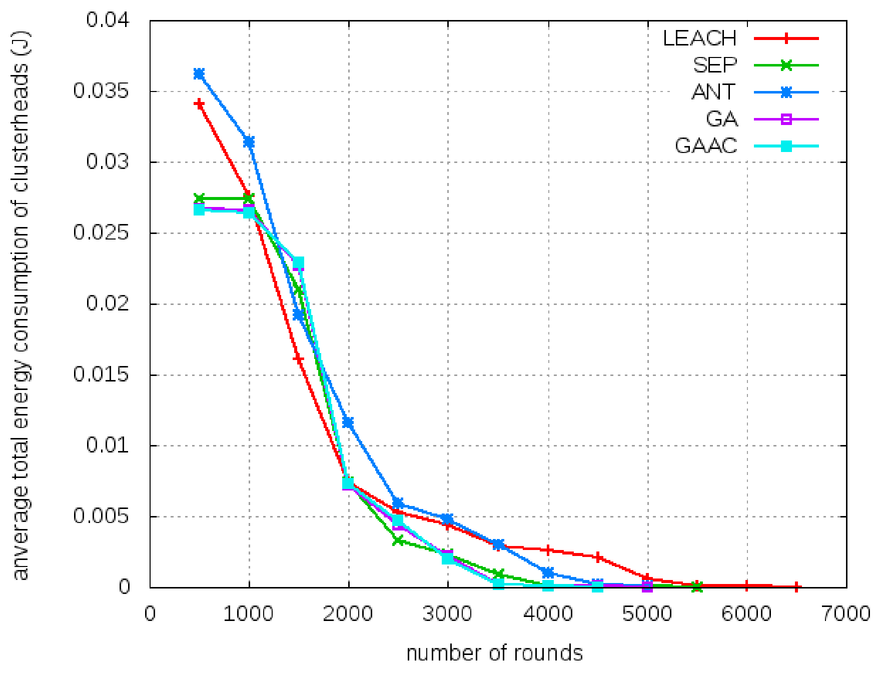


Figure 7.7: *Comparison of the average energy consumption of cluster-heads for various protocols.*

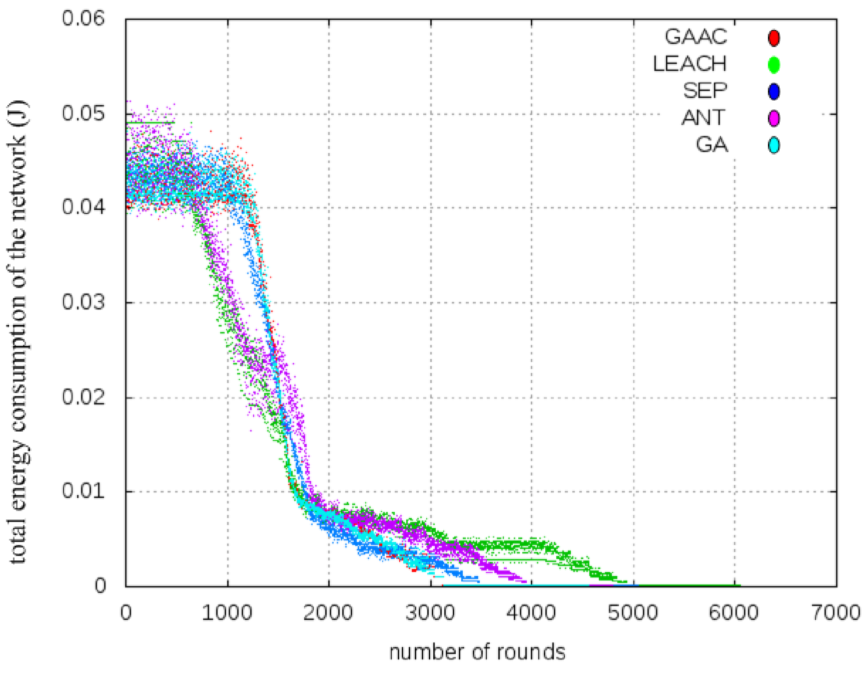


Figure 7.8: *Comparison of the energy consumption of the network for various protocols.*

consumption of cluster-heads and the energy consumption of the network for the various protocols. The result for GAAC are the lowest of all four protocols during the time until 60% nodes die. Compared to the other three protocols, the transmission distances between cluster-heads in the case of GAAC are significantly

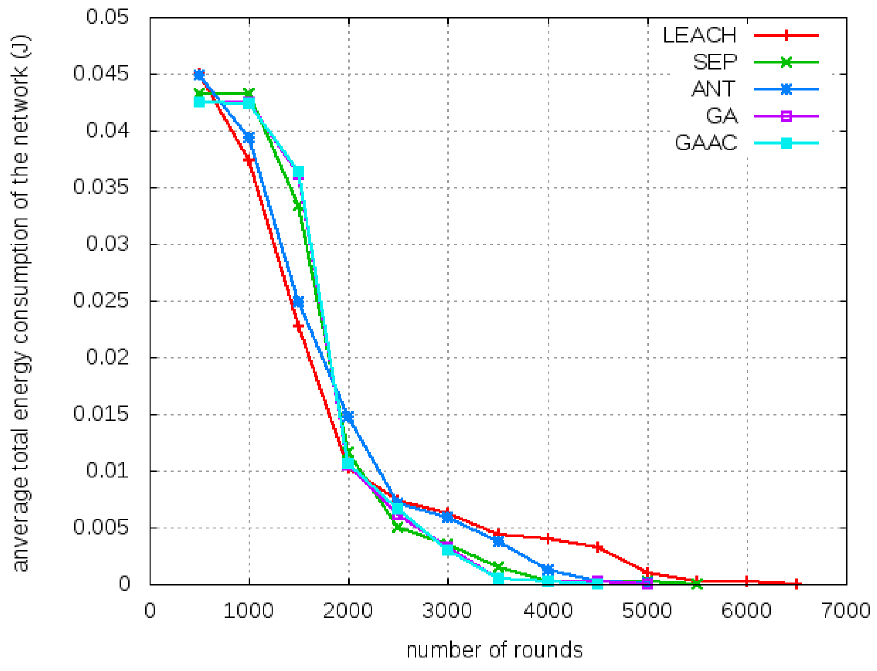


Figure 7.9: *Comparison of the average energy consumption of the network for various protocols.*

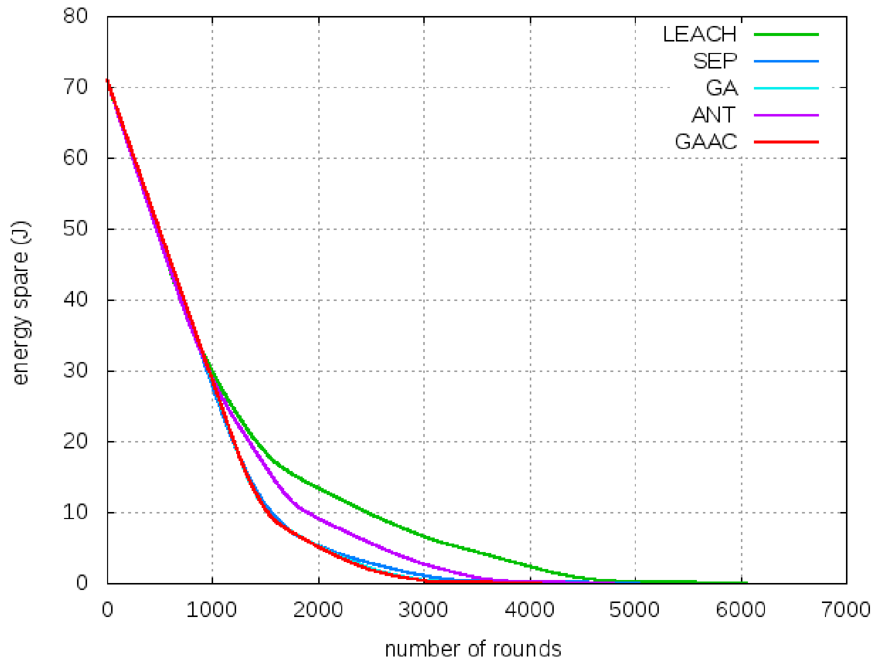


Figure 7.10: *Comparison of residual energy of the network for various protocols.*

shorter. Figure 7.10 illustrates the energy capacity of the network per round per Epoch in all protocols. It clearly indicates that, GAAC has a higher level of remaining energy in the network compared to other protocols during the stable and operational time.

7.4 Comparison to related algorithms and complexity analysis

The contributions of the EECHE protocol (proposed by Kumar et al. in [73]) are:

1. Extending two types of sensor nodes (namely, high energy nodes and normal nodes) of the SEP protocol [39] to three types of nodes (namely, the few type-3 and type-2 nodes of α and β times more energy than the type-1 nodes),

2. Assigning three types of nodes with three weighted probabilities (p_1, p_2, p_3) of cluster-head election.

In SEP, it is assumed that there are two types of sensor nodes, where the high energy nodes have α times more energy than normal nodes. The key contribution of the SEP protocol in [39] is the proposed cluster-head election scheme for a heterogeneous network. The experimental results proved that it prolongs the network lifetime compared to LEACH in a heterogeneous network with two types of nodes. This cluster-head election scheme is used in the EECHE protocol.

The GAAC algorithm can be readily extended to accommodate more than two types of node. In the general case where there are n types of nodes, this requires assigning each type of node a weighted probability (p_1, p_2, .. p_n) of cluster-head election. The extension of the SEP cluster-head election method to deal with this is straightforward. The algorithm as described in this chapter supports just two types of node only for clarity of exposition and not because of an algorithmic constraint.

I evaluated and analysed the performance of the proposed algorithm for heterogeneous networks with two types of nodes. I have shown that GAAC offers better performances than SEP in this case. The properties of GAAC in networks with more than two types of node has been left as a topic for future study. Although its performance could be compared to EECHE in the case of networks with three types of nodes, no similar algorithm can be extended to an arbitrary number of node types, as GAAC can.

GAAC is a scalable algorithm. Its ability to accommodate multiple types of node is one aspect of that scalability. Another is that, in its cluster-head election method, it randomly picks nodes as cluster-heads on a node-by-node basis and rotates this role among nodes. This helps to balance energy consumption across a large number of nodes.

The complexity of the proposed heuristic algorithm is evaluated below.

According to Cayley's formula [77], for any positive integer n, the number of trees on n vertices is $T_n = n^{(n-2)}$. Using an enumeration method to generate a minimum spanning tree results in a large amount of calculation.

Using the genetic sub-algorithms of GAAC, the best solution is a tree and the probability p_{ga} of finding the optimal solution is:

$$p_{ga} = \frac{k \times (q+1)}{n^{(n-2)}}, \text{ where } q \in [1, +\infty), k \in [1, +\infty),$$

and where k is the initial population size (so that initially we have k trees on n vertices) and q is the number of iterations per round in the GA. The initial population size k is not set to a large value. Thus, relatively few evaluations of the objective function are needed, reducing the amount of computation required. For example, if we set k to 20 and q to 100, we have $p_{ga} = \frac{k \times (q+1)}{n^{(n-2)}} = \frac{2020}{n^{(n-2)}}$.

Using the ACO sub-algorithm of GAAC, the best solution is also a tree. During the initial algorithm searching period, there is no path feedback information in the ACO algorithm. In a directed complete graph, the number of edges is $n \times (n-1)$ where n is the number of vertices. The probability p_{ant} of finding the optimal solutions is $\frac{Q}{n^{(n-2)}}$ where Q is the number of edges that all ants pass. For example, if we set the number of ants on each node to 5, then Q is $5 \times (n-1) \times n \simeq 5 \times n^2$, so $p_{ant} = \frac{Q}{(n^{(n-2)})} = \frac{5 \times n^2}{n^{(n-2)}}$.

Both expressions have the same denominator, but the numerator of p_{ga} is lower in value than that of p_{ant}. Therefore, $p_{ga} < p_{ant}$. As shown in Figure 7.1, both the genetic algorithm and the ACO algorithm will ultimately find the optimal solution. However, the pure ACO algorithm expends much computational effort to generate enough possible solutions in order to identify the optimal solution in each iteration - my simulations of this algorithm have sometimes required more than one week to execute. The pure genetic algorithm can get the result quickly. The trade-off of these two algorithms in the hybrid solution of GAAC results in a faster convergence to the optimal solution than with either acting alone.

7.5 Summary

In this chapter, I have described a sensor-routing algorithm called GAAC, based on genetic algorithms and ant colony algorithms, for heterogeneous wireless sensor networks. I assumed that a heterogeneous network consists of two types of nodes with different initial energy, normal nodes and advanced nodes. I have shown its excellent performance and flexibility in comparison to similar algorithms in the literature.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

This thesis presents novel algorithms for heterogeneous routing within wireless sensor networks. Algorithms which are energy-efficient in a homogeneous environment will behave suboptimally in a heterogeneous environment, because the nodes have different energy consumptions and reserves.

This thesis provides a survey of the researches in this field to date and identifies new directions in wireless sensor network routing protocols.

The main goal of this research is to develop a new routing model, framework that will solve many of the shortcomings of existing mechanisms in terms of energy efficiency. It proposes a three-tier architecture and a flexible framework, called GSF - generic software framework. Three new algorithms called TinyReg - a routing algoirthm based on regular-graph theory , TinyReg - topological stable election protocol and GAAC - an evolutionary algorithm based on genetic algorithms and ant colony algorithms respectively have been described.

The choice of topology is important because it has been shown heavily influence in the energy consumption of the network in the literatures [24][2]. Regular topologies can help to efficiently save energy and achieve long network lifetime [78][79]. This thesis combines regular graph theory and hierarchical clustering routing in the implementation of wireless sensor network routing. TinyReg is a global algorithm with less overheads on cluster-heads (compared to TSEP); meanwhile, TSEP is a localised algorithm but more robustness for lager scale sensor networks (compared to the former).

It also applies three techniques, namely, clustering, genetic algorithms and ant colony algorithms, to the routing problem in wireless sensor networks. This is the first time that all three of these approaches have been combined in a single algorithm (GAAC). I have shown that this extends the useful lifetime of the network.

8.2 Future work

This thesis presents novel methods for routing strategies in heterogeneous sensor networks. In the future, a number of interesting research problems require further investigation.

It will be interesting to apply the proposed methods in this thesis to some real world networks. It is meant to apply our methods (TinyReg, TSEP and GAAC) in a live wireless sensor network using the variety of sensors available in my laboratory. However, the amount of sensor nodes available for my use in the laboratory does not allow me to create a network composed of more than 15 nodes.

The algorithms (TinyReg, TSEP and GAAC) proposed in this thesis are developed on the software framework GSF. It will be interesting to rewrite other researchers's prototype routing protocols to enrich the components of this framework. Another possibility is to test every possible combined strategies of components from existing prototype routing protocols under this framework depending on application requirements. I would like to perform tests of these methods to examine the performance of these techniques when sensors have different initial energy and energy consumptions.

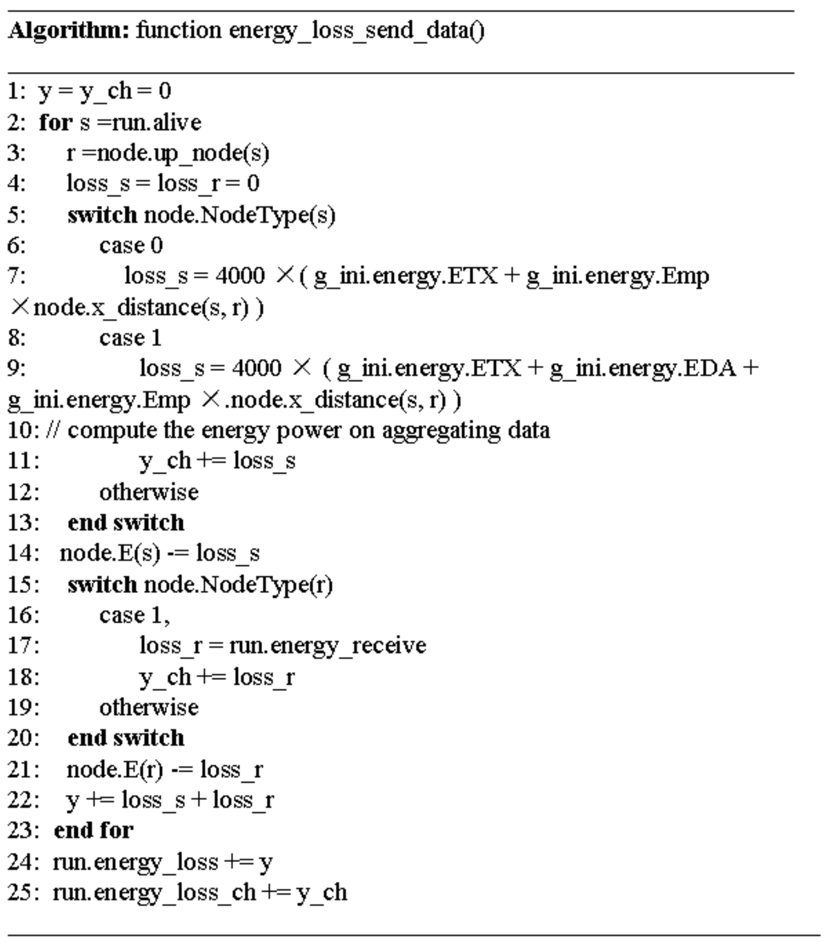
APPENDIX A

THE INCORPORATION OF AGGREGATION INTO LEACH

The LEACH protocol is designed for a homogeneous network where all nodes are equipped with the same initial battery energy.

In my experiments, I improve LEACH by adding a data aggregation function to the protocol. In the original LEACH protocol, the implicit assumption is that nodes do not support an aggregation function and thus the reduction in energy consumption that is obtained by data aggregation is absent. This makes for an unfair comparison between LEACH and other algorithms (which may perform poorly against it were it not for their use of data aggregation). Accordingly, I have modified the original LEACH algorithm to incorporate aggregation and it is this modified algorithm that has been used to obtain the baseline LEACH results throughout this thesis.

The Matlab code used to incorporate aggregation in LEACH is shown below:



Algorithm: function energy_loss_send_data()

```
y = y_ch = 0
for s =run.alive
    r =node.up_node(s)
    loss_s = loss_r = 0
    switch node.NodeType(s)
        case 0
            loss_s = 4000 × ( g_ini.energy.ETX + g_ini.energy.Emp ×node.x_distance(s, r) )
        case 1
            loss_s = 4000 × ( g_ini.energy.ETX + g_ini.energy.EDA + g_ini.energy.Emp ×.node.x_distance(s, r) )
// compute the energy power on aggregating data
            y_ch += loss_s
        otherwise
    end switch
    node.E(s) -= loss_s
    switch node.NodeType(r)
        case 1,
            loss_r = run.energy_receive
            y_ch += loss_r
        otherwise
    end switch
    node.E(r) -= loss_r
    y += loss_s + loss_r
end for
run.energy_loss += y
run.energy_loss_ch += y_ch
```

Figure A.1: *Incorporating aggregation in LEACH*

BIBLIOGRAPHY

[1] K. Sohraby, D. Minoli, and T. Znati. *Wireless Sensor Networks - Technology, Protocols, and Applications*. John Wiley & Sons, Inc, Publications., 2007.

[2] X. F. Li, Y. C. Mao, and Y. Liang. A survey on topology control in wireless sensor networks. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 251–255, 2008.

[3] C. E. Perkins. *Ad Hoc Networking*. Addison-Wesley, New York, 2000.

[4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 263–270, 1999.

[5] E. M. Royer and C. K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications Magazine*, 6(2):46–55, Apr. 1999.

[6] J. Elson and D. Estrin. An address-free architecture for dynamic sensor networks. Technical report, UCLA, 2000.

[7] J. Broch, D. A. Maltz, and D. B. Johnson *et al*. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of*

the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 85–97, 1998.

[8] G. Asada, M. Dong, T. S. Lin, F. Newberg, and G. Pottie *et al*. Wireless integrated network sensors: Low power systems on a chip. In *Proceedings of the 24th European Solid-State Circuits Conference (ESSCIRC)*, pages 9–16, Sep. 1998.

[9] K. Sohrabi, V. Ailawadhi J. Gao, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications Magazine*, 7(5):16–27, 2000.

[10] A. Ephremides. Energy concerns in wireless networks. *IEEE Wireless Communications Magazine*, 9(4):48–59, Aug. 2002.

[11] M. Ilyas and I. Mahgoub. *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press, USA, 2005.

[12] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[13] J. Adams. Designing with 802.15.4 and zigbee. In *Proceedings of Industrial Wireless Applications Summit, San Diego, CA*, Mar. 2004.

[14] M. Welsh. Sensor networks for medical care. Technical Report TR-08-05, Harvard University Technical Report, April. 2005.

[15] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005.

[16] A. Manjeshwar and D. P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS)*, page 189, Washington, DC, USA, 2001. IEEE Computer Society.

[17] S. Fedor and M. Collier. On the problem of energy efficiency of multi-hop vs one-hop routing in wireless sensor networks. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, pages 1–5, 2007.

[18] Q. Qang, H. Hassanein, and K. Xu. A practical perspective on wireless sensor networks. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, 2004.

[19] L. Subramanian and R. H. Katz. An architecture for building self-configurable systems. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, pages 63–73, Piscataway, NJ, USA, 2000.

[20] E. J. Duarte-Melo and M. Liu. Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pages 21–25, Nov. 2002.

[21] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of IEEE/ACM International Conference on Information Processing in Sensor Networks*, 2005.

[22] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *International Journal on Ad Hoc Networks*, 3(3):325–349, 2005.

[23] J. Pan, L. Cai, and Y. T. Hou *et al*. Optimal base-station locations in two-tiered wireless sensor networks. *IEEE Transactions on Mobile Computing*, 4(5):458–473, 2005.

[24] J. Pan, Y. T. Hou, and L. Cai *et al*. Topology control for wireless sensor networks. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 286–299, 2003.

[25] F. Ye, H. Luo, and J. Cheng *et al*. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of ACM International*

Conference on Mobile Computing and Networking (MobiCom), pages 148–159, 2002.

[26] M. Gupta and G. Younis. Fault-tolerant clustering of wireless sensor networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1579–1584, March. 2003.

[27] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. In *Proceedings of the 23rd IEEE International Conference on Computer Communications (INFCOM)*, pages 629–640, 2004.

[28] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Journal of IEEE Transactions on Wireless Communications*, 1:660–670, 2002.

[29] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (Mobicom)*, pages 70–84, 2001.

[30] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th ACM annual international conference on Mobile computing and networking (MobiCom)*, pages 56–67, New York, NY, USA, 2000.

[31] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 174–185, New York, NY, USA, 1999.

[32] S. Lindsay, C. S. Raghavendra, and K. M. Sivalingam. Data gathering in sensor networks using the energy delay metric. In *Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS)*, Washington, DC, USA, 2001. IEEE Computer Society.

[33] S. Lindsey and C. S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *Proceedings of IEEE Aerospace Conference Proceedings*, pages 1125–1130, 2002.

[34] A. Manjeshwar and D. P. Agrawal. Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS)*, page 48, Washington, DC, USA, 2002. IEEE Computer Society.

[35] M. Younis, M. Youssef, and K. Arisha. Energy-aware routing in cluster-based sensor networks. In *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS)*, pages 129–136, 2002.

[36] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the 7th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 85–96, New York, NY, USA, 2001.

[37] A. Papadopoulos, A. Navarra, J. A. McCann, and C. M. Pinotti. Vibe: An energy efficient routing protocol for dense and mobile sensor networks. *Journal of Network and Computer Applications*, 35(4):1177–1190, 2012.

[38] A. Forster. Machine learning techniques applied to wireless ad-hoc networks: Guide and survey. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, pages 365–370, 2007.

[39] G. Smaragdakis, I. Matta, and A. Bestavros. Sep: A stable election protocol for clustered heterogeneous wireless sensor networks. In *Proceedings of the 2rd International Workshop on Sensor and Actor Network Protocols and Applications*, pages 1–11, 2004.

[40] S. M. A. Oteafy, H. M. AboElFotoh, and H. S. Hassanein. Dynamic election-based sensing and routing in wireless sensor networks. In *Proceedings of the 28th IEEE Conference on Global Telecommunications*, pages 5847–5852, 2009.

[41] S. A. Bermudez and S. B. Wicker. Taking advantage of data correlation to control the topology of wireless sensor networks. In *Proceedings of International Conference on Telecommunications*, pages 1–6, 2008.

[42] S. Sahni and X. Xu. Algorithms for wireless sensor networks. *International Journal on Distributed Sensor Networks*, 1(1):35–56, 2005.

[43] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, 11(1):2–16, 2003.

[44] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Select Areas in Communications*, 17(8):1333–1344, Aug. 1999.

[45] L. Li and J. Y. Halpern. Minimum-energy mobile wireless networks revisited. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 278 –283, 2001.

[46] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, 2001.

[47] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, page 46, Washington, DC, USA, 2003. IEEE Computer Society.

[48] Z. Marco, D. Waltenegus, and R. M. Johnathan. Localized power-aware routing in linear wireless sensor networks. In *Proceedings of the 2nd ACM International Conference on Context-awareness for Self-managing Systems (CASEMANS)*, pages 24–33, 2008.

[49] M. C. Huebscher and J. A. McCann. A survey of autonomic computing - degrees, models, and applications. *Journal of ACM Computing Surveys*, 40(3), 2008.

[50] U. Chandrasekhar and P. R. P. Naga. Recent trends in ant colony optimization and data clustering: A brief survey. In *Proceedings of IEEE International Conference on Intelligent Agent and Multi-Agent Systems (IAMA)*, pages 32–36, 2011.

[51] E. A. Khalil and B. A. Attea. Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks. *International Journal of Swarm and Evolutionary Computation*, 1(4):195–203, 2011.

[52] D. A. Coley. *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific, 1999.

[53] J. H. Holland. *Adaptation in Nature and Artificial Systems*. The University of Michigan Press, 1975.

[54] S. S. Iyengar and R. R. Brooks. *Distributed Sensor Networks*. Chapman and Hall, 2004.

[55] X. Yang. *Engineering Optimization-An Introduction with Metaheuristic Applications*. A John Wiley & Sons, Inc., Publication, 2010.

[56] P. H. Winston. *Artificial Intelligence*. Addison-Wesley Publishing Company, 1993.

[57] R. V. Kulkarni, A. Forster, and G. K. Venayagamoorthy. Computational intelligence in wireless sensor networks A survey. *IEEE Communications Surveys & Tutorials*, 13(1):68–96, 2011.

[58] X. M. Hu and J. Zhang *et al*. Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks. *IEEE Transactions on Evolutionary Computation*, 14(5):766–781, 2010.

[59] Z. Yao, J. Liu, and Y. G. Wang. Fusing genetic algorithm and ant colony algorithm to optimize virtual enterprise partner selection problem. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 3614–3620, 2008.

[60] A. J. Ramirez, D. B. Knoester, B. H. C. Cheng, and P. K. McKinley. Applying genetic algorithms to decision making in autonomic computing systems. In *Proceedings of IEEE Autonomic Computing and Communications Conference (ICAC)*, pages 97–106, 2009.

[61] P. Taejin and R. K. Ryel. A dual-population genetic algorithm for adaptive diversity control. *IEEE Transactions on Evolutionary Computation*, 14(6):865–884, 2010.

[62] D. C. Hoang, P. Yadav, R. Kumar, and S. K. Panda. A robust harmony search algorithm based clustering protocol for wireless sensor networks. In *Proceedings of IEEE International Conference on Communications Workshops*, 2010.

[63] O. Islam and S. Hussain. An intelligent multi-hop routing for wireless sensor networks. In *Proceedings of IEEE/WIC/ACM Workshops of International Conference on Web Intelligence and Intelligent Agent Technology Workshops*, pages 239–242, 2006.

[64] K. Dasgupta, K. Kalpakis, and P. Namjoshi. An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. In *Proceedings of IEEE Confernece on Wireless Communications and Networking (WCNC)*, volume 3, pages 1948–1953, 2003.

[65] S. Y. Jin, M. Zhou, and A. S. Wu. Sensor network optimization using a genetic algorithm. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics Proceedings*, volume 5, pages 257–262, 2003.

[66] S. R. Mudundi. A new robust genetic algorithm for dynamic cluster formation in wireless sensor networks. In *Proceedings of International Conferences on Wireless and Optical Communications*, pages 360–367, 2007.

[67] A. W. Matin and S. Hussain. Intelligent hierarchical cluster-based routing. In *Proceedings of International Workshop on Mobility and Scalability in Wireless Sensor Networks (MSWSN) of IEEE International Conference on Distributed Computing in Sensor Networks (DCOSS)*, pages 165–172, 2006.

[68] M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.

[69] M. Dorigo and T. Stutzle. *Ant colony optimization*. MIT Press, USA, 2004.

[70] Z. Gong and D. Li *et al*. An analysis of ant colony clustering methods Models, algorithms and applications. *International Journal of Advancements in Computing Technology*, 3(11):112–121, 2011.

[71] T. Camilo, C. Carreto, J. S. Silva, and F. Boavida. An energy-efficient ant-based routing algorithm for wireless sensor networks. In *Proceedings of the 5th ACM International Conference on Ant Colony Optimization and Swarm Intelligence*, pages 49–59, 2006.

[72] W. H. Liao, Y. C. Kao, and C. M. Fan. An ant colony algorithm for data aggregation in wireless sensor networks. In *Proceedings of International Conference on Sensor Technologies and Applications*, pages 101–106, Oct. 2007.

[73] D. Kumar, T. C. Aseri, and R. B. Patel. Eeche: energy-efficient cluster head election protocol for heterogeneous wireless sensor networks. In *Proceedings of the International Conference on Advances in Computing, Communication and Control*, pages 75–80, 2009.

[74] C. St. J. A. Nash-Williams. Valency sequences which force graphs to have hamiltonian circuits. Technical report, University of Waterloo Research Report, Waterloo, Ontario: University of Waterloo, 1969.

[75] Y. Y. Zeng, C. J. Sreenan, and L. Sitanayah. A real-time and robust routing protocol for building fire emergency applications using wireless sensor networks. In *Proceedings of the 8th IEEE Workshop on Pervasive Computing and Communications*, pages 358–363, 2010.

[76] K. P. Ferentinos and T. A. Tsiligiridis. Adaptive design optimization of wireless sensor networks using genetic algorithms. *International Journal of Computer Networks*, 51(4):1031–1051, 2007.

[77] A. Cayley. A theorem on trees. 23:376C378, 1889.

[78] H. Tian, H. Shen, and T. Matsuzawa. Random walk routing for wireless sensor networks. In *Proceedings of the 6th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pages 196–200, 2005.

[79] R. Iyengar, K. Kar, and S. Banerjee. Low-coordination topologies for redundancy in sensor networks. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 332–342, New York, USA, 2005.

[80] B. Krishnamachari, D. Estrin, and S. Wicker. Modelling data-centric routing in wireless sensor networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, June. 2002.

[81] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless micro-sensor network models. *Newsletter of ACM SIGMOBILE Mobile Computing and Communications Reviwe*, 6(2):28–36, 2002.

[82] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA)*, pages 88–97, Sep. 2002.

[83] C. R. Lin and J. S. Liu. Qos routing in ad hoc wireless networks. *IEEE Journal on Selected Area in Communications*, pages 1426–1438, Aug. 1999.

[84] A. Boukerche and A. Martirosyan. An energy-aware and fault tolerant inter-cluster communication based protocol for wireless sensor networks. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1164–1168, Nov. 2007.

[85] Y. Dong, Q. Quan, and J. Zhang. Priority-based energy aware and coverage preserving routing for wireless sensor network. In *Proceedings of IEEE Vehicular Technology Conference (VTC)*, pages 138 –142, May. 2008.

[86] A. Abdalkarim, G. Reinhard, and D. Falko. P2p-based routing and data management using the virtual cord protocol (vcp). In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, pages 443–444, 2008.

[87] N. Nicolaou, A. See, J. Cui, and D. Maggiorini. Improving the robustness of location-based routing for underwater sensor networks. In *Proceedings of MTS/IEEE OCEANS Conference*, 2007.

[88] D. Kumar, T. C. Aserib, and R. B. Patelc. Eehc: Energy efficient heterogeneous clustered scheme for wireless sensor networks. *International Journal on Computer Communications*, 32, 2009.

[89] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 1–10, 2000.

[90] D. L. Guidoni, A. F. R. Mini, and A. F. A. Loureiro. On the design of heterogeneous sensor networks based on small world concepts. In *Proceedings of the 11th ACM international symposium on Modeling, analysis and simulation of*

wireless and mobile systems (MSWiM), pages 309–314, New York, NY, USA, 2008.

[91] H. Y. Lee, W. K. G. Seah, and P. Sun. Energy implications of clustering in heterogeneous wireless sensor networks - an analytical view. In *Proceedings of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2006.

[92] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 70–84, New York, NY, USA, 2001.

[93] J. J. Lotf, M. Hosseinzadeh, and R. M. Alguliev. Hierarchical routing in wireless sensor networks: a survey. In *Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET)*, volume 3, pages 650–654, 2010.

[94] G. Z. Zheng and Q. M. Liu. A survey on topology control in wireless sensor networks. In *Proceedings of the 2rd International Conference on Future Networks (ICFN)*, pages 376–380, 2010.

[95] K. Pavai, A. Sivagami, and D. Sridharan. Study of routing protocols in wireless sensor networks. In *Proceedings of International Conference on Advances in Computing, Control, Telecommunication Technologies (ACT)*, pages 522–525, 2009.

[96] C. F. Jiang, D. M. Yuan, and Y. H. Zhao. Towards clustering algorithms in wireless sensor networks-a survey. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2009.

[97] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Inter-*

national Journal of High Performance Computing Applications, 16(3):293–313, 2002.

[98] G. Chartrand and O. R. Oellermann. *Applied and Algorithmic Graph Theory*. McGraw Hill, New York, 1993.

[99] A. Manjeshwar and D. P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS)*, 2001.

[100] K. Kowalik and M. Collier. Connectivity aware routing - a method for finding bandwidth constrained paths over a variety of network topologies. In *Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, 2003.

[101] D. Kumar, T. C. Aserib, and R. B. Patelc. Eehc: Energy efficient heterogeneous clustered scheme for wireless sensor networks. *Journal of Computer Communications*, 32, 2009.

[102] Y. Lin and J. Zhang. An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks. *IEEE Transactions on System, Man, and Cybernetics Part C: Applications and Reviews*, 2011.

[103] J. Zhong and J.Zhang. Energy-efficient local wake-up scheduling in wireless sensor networks. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 2280–2284, 2011.

[104] J. G. Yang and J. Q. Yang. Intelligence optimization algorithms: A survey. *International Journal of Advancements in Computing Technology*, 3(4):144–152, 2011.

[105] S. Hussain and A. W. Matin. Hierarchical cluster-based routingin wireless sensor networks. In *Proceedings of IEEE/ACM International Conferenceon Information Processing in Sensor Networks*, 2006.

[106] S. Hussain, A. W. Matin, and O. Islam. Genetic algorithm for hierarchical wireles sensor networks. *International Journal of Networks (JNW)*, 2(7):87–97, 2007.

[107] E. M. Shakshuki, H. Malik, and T. R. Sheltami. Multi-agent-based clustering approach to wireless sensor networks. *International Journal of Wireless and Mobile Computing*, 3(3):165–176, 2009.

[108] A. Lanjewar, V. N. Sahare, and N. Sahare. An approach based on clustering method for object finding mobile robots using aco. In *Proceedings of the 2nd ACM International Conference on Machine Learning and Computing*, pages 161–165, 2010.

[109] J. L. Deneubourg and S. Goss *et al*. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the 1st International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, pages 356–363, 1990.

[110] R. Misra and C. Mandal. Ant-aggregation: ant colony algorithm for optimal data aggregation in wireless sensor networks. In *Proceedings of IFIP International Conference on Wireless and Optical Communications Networks*, pages 1–5, 2006.

[111] G. Wang, Y. Wang, and X. Tao. An ant colony clustering routing algorithm for wireless sensor networks. In *Proceedings of IEEE International Conference on Genetic and Evolutionary Computing*, pages 670–673, 2009.

[112] J. Barbancho, C. Leon, J. Molina, and A. Barbancho. Giving neurons to sensors: Qos management in wireless sensors networks. In *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 594–597, 2006.

[113] I. Gupta, D. Riordan, and S. Sampalli. Cluster-head election using fuzzy logic for wireless sensor networks. In *Proceedings of the 3rd Annual Communication Networks and Services Research Conference*, pages 255–260, 2005.

[114] M. T. Islam, P. Thulasiraman, and R. K. Thulasiram. A parallel ant colony optimization algorithm for all-pair routing in manets. In *Proceedings of International Parallel and Distributed Processing Symposium*, 2003.

[115] S. Wazed, A. Bari, A. Jaekel, and S. Bandyopadhyay. Genetic algorithm based approach for extending the lifetime of two-tiered sensor networks. In *Proceedings of the 2nd IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, 2007.

[116] F. Xue, A. Sanderson, and R. Graves. Multi-objective routing in wireless sensor networks with a differential evolution algorithm. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 880–885, 2006.

[117] E. D. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the third international conference on Simulation of adaptive behavior : from animals to animats*, pages 501–508, 1994.

[118] S. M. Guru, S. K. Halgamuge, and S. Fernando. Particle swarm optimisers for cluster formation in wireless sensor networks. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 319–324, 2005.

[119] P. Arabshahi, A. Gray, I. Kassabalidis, and A. Das *et al*. Adaptive routing in wireless communication networks using swarm intelligence. In *Proceedings of the 19th Annual Satellite Communications System Conference*, 2001.

[120] S. Bashyal and G. K. Venayagamoorthy. Collaborative routing algorithm for wireless sensor network longevity. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, pages 515–520, 2007.

[121] R. Muraleedharan and L. Osadciw. A predictive sensor network using ant system. In *Proceedings of International Society for Optical Engineering*, 2004.

[122] A. Forster and A. L. Murphy. Clique: Role-free clustering with q-learning for wireless sensor networks. In *Proceedings of 29th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 441–449, 2009.

[123] Y. Zhang and M. P. J. Fromherz. A robust and efficient flooding-based routing for wireless sensor networks. *International Journal of Interconnection Networks*, 7(4):549–568, 2006.

[124] P. Wang and T. Wang. Adaptive routing for sensor networks using reinforcement learning. In *Proceedings of the 16th IEEE International Conference on Computer and Information Technology (CIT)*, page 219, 2006.

[125] B. Yu and P. Scerri *et al*. Scalable and reliable data delivery in mobile ad hoc sensor networks. In *Proceedings of the 5th ACM International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1071–1078, 2006.

[126] S. Dong, P. Agrawal, and K. Sivalingam. Reinforcement learning based geographic routing protocol for uwb wireless sensor network. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pages 652–656, 2007.

[127] A. Forster and A. L. Murphy. Froms: Feedback routing for optimizing multiple sinks in wsn with reinforcement learning. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, pages 371–376, 2007.

[128] J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: a reinforcement learning approach. In *Proceedings of Advances in Neural Information Processing Systems*, pages 671–678, 1994.

[129] P. Beyens, M. Peeters, K. Steenhaut, and A. Nowe. Routing with compression in wireless sensor networks: a q-learning approah. In *Proceedings of*

the 5th European Workshop on Adaptive Agents and Multi-Agent Systems (AA-MAS), 2005.

[130] S. Kumar and R. Miikkulainen. Dual reinforcement q-routing: an on-line adaptive routing algorithm. In *Proceedings of the Artificial Neural Networks in Engineering Conference*, pages 231–238, 1997.

[131] P. Stone. Tpot-rl applied to network routing. In *Proceedings of the 17th International Conference on Machine Learning*, pages 935–942, 2000.

[132] J. Dowling, E. Curran, R. Cunningham, and V. Cahill. Using feedback in collaborative reinforcement learning to adaptively optimize manet routing. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(3):360–372, 2005.

[133] R. Arroyo-Valles, R. Alaiz-Rodriguez, A. Guerrero-Curieses, and J. Cid-Sueiro. Q-probabilistic routing in wireless sensor networks. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, pages 1–6, 2007.

[134] F. Szymon. *Cross-layer energy optimisation of routing protocols in wireless sensor networks*. PhD thesis, Dublin City University, Ireland, 2009.

[135] P. Szczechowiak. *Cryptographic Key Distribution In Wireless Sensor Networks Using Bilinear Pairings*. PhD thesis, Dublin City University, Ireland, 2010.

[136] N. Sadagopan, B. Krishnamachari, and A. Helmy. The acquire mechanism for efficient querying in sensor networks. In *Proceedings of IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, pages 149–155, 2003.

[137] K. M. Sim and W. H. Sun. Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 33(5):560–572, 2003.

[138] J. Kulik, W. Heinzelman, and H. Balakrishnan. Negotiation-based protocols for disseminatiing information in wireless sensor networks. *Journal of Wireless Networks*, 8:169–185, 2002.