Chapter 1

Energy Restoration in Mobile Sensor Networks

Nicola Santoro and Elio Velazquez
*School of Computer Science, Carleton University,
Ottawa, Canada*

Restoration of sensors' energy is crucial to ensure continuous operations of a sensor network. In static sensor networks, energy restoration requires the use of special mobile entities (e.g., robots, actuators). In mobile sensor networks, the sensors themselves can solve the problem by moving to recharge facilities deployed throughout the sensing area. An efficient energy restoration strategy is one in which the losses are limited in scope and time; that is, with the minimum number of sensor losses, it allows the system to reach a state of equilibrium guaranteeing continuous operation of the network without any further sensor losses. In this Chapter we describe and examine the basic strategies for energy restoration, passive and active, and analyze their performance.

1. Introduction

1.1. *Energy Management in Sensor Networks*

Regardless of the specific application being addressed, the ultimate goal of any sensor network is to achieve accurate sensing and maximize lifetime while maintaining an acceptable level of coverage. Since every sensor operation consumes energy, in any wireless sensor network deployment, the sensors' batteries will eventually deplete, and loss of coverage will occur.

The most simplistic approach to cope with the eventual loss of coverage has been to deploy a very large number of sensors to compensate for the loss of the depleted ones. In this approach, the spare sensors must detect when a sensing hole is created by the battery depletion of a sensor, and take its place. A large number of studies have indeed been dedicated to these problems (e.g., see [1, 2]). However, for obvious environmental reasons, these kind of solutions are not really sustainable; furthermore, regardless of the number of spares, a loss of coverage over time is inevitable since there are no provisions to recharge or replace sensors in the long run.

The study of *energy management* has mostly focused on balancing the energy levels among all sensors (e.g., [3, 4]), rather than on energy restoration. For example, since sensors closer to the base station have to route/aggregate data flowing from remote parts of the network towards the base station, they tend to deplete their batteries much faster than other sensors; to overcome this disparity, a *mobile* base station could be used so to extend network operating life [3].

The crucial and constant concern, in every aspect of a sensor network design, on the minimization of energy consumption only ensures to delay the time of loss of coverage. But, in absence of provisions to recharge sensors, the network will eventually fail. The focus of this chapter is rather on *energy restoration* and providing *continuous operation* of the network.

1.2. *Energy Restoration in Static Sensor Networks*

To achieve *energy restoration* in static sensor networks, one approach is to have the sensors extracting energy from the environment (e.g., [5, 6]). The alternative approach to energy restoration in static sensor networks is to employ special entities to recharge the depleted sensors; these special entities could be particularly equipped sensors or dedicated robots; this approach is advocated also for network repairs where the special maintenance entities replace depleted sensors (e.g., [7–10]). For example, in [11], static sensors are recharged by chargers or actuators carrying solar panels.

In general, these energy restoration strategies can be categorized into two groups with some degree of overlap: *cluster-based* approaches (e.g., [12–15]) or *mobility-based* approaches (e.g., [3, 4, 7, 16, 17]). Indeed, most of these solutions rely on some kind of clustering or partitioning of the network with the special entities as cluster heads; they operate by either creating a fixed partition of the field or by constructing and maintaining dynamic clustering structures which depend on the current position of the cluster heads (e.g., [8, 10]). In each cluster, the special entity must obtain information about the energy distribution in the cluster by collecting additional information embedded in each communication with the static sensors (e.g., maximum energy, remaining energy, location, etc.) to decide when and where to intervene.

Other examples of these approaches can be found in research papers in robotics [11, 18, 19]. In all cases, a charger robot is responsible for delivering energy to a swarm of sensors. The recharging strategy is completely reactive (i.e., sensors are only recharged when they become out of service and cannot move). In the scenario described in [18], the charger robot is equipped with several docking ports. However, the charger robot can travel to recharge a robot in need only if none of the docking ports are occupied, assuming that several depleted robots need to be close by in order to be recharged simultaneously. The solution presented in [19], where a team of mini-robots (our sensors) are deployed along with more powerful docking station robots, is based on the creation of clusters of mini-robots.

1.3. *Energy Restoration in Mobile Sensor Networks*

All the above mentioned approaches for energy restoration are tailored for traditional sensor networks, that is where the sensors are *static*. The situation is drastically different in the case *mobile* sensor networks, that is where the sensors are endowed with motor capabilities. Mobility clearly enhances the system allowing to perform tasks impossible in static sensor networks such as *self-deployment* [20–22], *aerial and maritime unmanned applications* [23–25].

In the case of mobile sensor networks, the sensors themselves can solve the energy restoration problem by moving towards recharge facilities deployed throughout the sensing area; in other words, the responsibility for maintaining the overall health of the network can be shifted to the sensor side, whereas the service facilities can play a passive role.

To ensure a *continuous* operation of a mobile sensor network, the mobile sensors are responsible for managing their own energy levels and for coming up with strategies to extend their operating life beyond one battery charge. The standard method to decide when to recharge has been based on fixed thresholds [17]. In this case, the service stations take a more passive role and the sensors should be able to compute their remaining operational time and coordinate the use of the service stations [16]. Furthermore, for instances where the sensors have to visit a predefined number of points of interests, [17] describes threshold vs. non threshold-based solutions where robots decide to visit the service stations depending on their proximity and the nature (locations) of the points of interests.

The problem of achieving continuous operation by refuelling or recharging mobile entities has been the focus of attention in recent research papers in robotics. In particular, a general version of this problem is the *Frugal Feeding Problem* (FFP), so called for its analogy with occurrences in the animal kingdom [26, 27]. The FFP consists to find energy-efficient routes for a mobile service entity, called "tanker", to rendezvous with every member of a team of mobile robots. The FFP has several variants depending on where the "feeding" or refuelling of the robots takes place: at each robot's location, at a predefined location (e.g., at the tanker's location) or anywhere. Regardless of which variant is chosen, the problem lies in ensuring that the robots reach the rendezvous location without "dying" by energy starvation during the process. The context of mobile sensors and static recharge facilities deployed throughout the sensing area corresponds to the "tanker absorbed" version of FFP: the "rendezvous" between the recharging facility (the tanker) and the mobile sensors (the robots) takes place at the location of the recharging facility.

The problem of where to place a service facility is examined by [28] for mobile robots. In this case, a team of mobile robots have the specific task of transporting certain items from a pick-up to a drop-off location. To be able to work for a prolonged period of time, the robots should interrupt their work and visit the recharge station periodically (i.e., tanker-absorbed FFP). Their solution is to place the charger station close enough to the path followed by the robots but without causing interference to the robots' movements.

In all the aforementioned scenarios there are some necessary conditions for the sensors to be able to recharge themselves [18]. First of all, the sensors must be able to monitor their energy levels and detect when it is time to recharge. Second, they must be able to locate and move towards a charging station. Finally, there must be a mechanism for the energy transfer either by docking or plugging into the charging station or via wireless recharging at short distances (e.g., [16, 29–31]).

The *perfect* energy restoration strategy should be able to guarantee a continuous operation of the network without any losses; however in reality some sensor losses will occur. A *successful* energy restoration strategy is one in which the losses are limited in scope and time; that is, the strategy allows the network to reach a state of equilibrium, where no further sensor losses will occur (thus guaranteeing a continuous operation of the network), with the minimum number of sensor losses. The basic strategies for energy restoration in mobile sensor networks using static recharge facilities deployed throughout the sensing area have been introduced in [32, 33]. In the rest of this Chapter we will describe and analyze these strategies.

2. Basic Terminology and Assumptions

The system is composed of a set $S = \{s_1, ..., s_N\}$ of N mobile sensors distributed in an area of unspecified shape and a set $F = \{f_1, ..., f_K\}$ of K static recharge facilities distributed throughout the area. Each facility is equipped with a fixed number of recharging ports or sockets; this represents the maximum number of simultaneous sensors at the facility.

The placement of the service facilities can be achieved using any of the clustering algorithms shown in [34–38]. Once the clustering creation is finalized, there will be exactly one recharging station for each sensor in S. The sensors will know the location of their recharging facility, but the facilities are not required to know the number of sensors that will use their resources. With a clustering structure already in place, we can focus on the interactions within a particular cluster. Therefore, without loss of generality, our strategies will be presented in the context of one facility and the subset of mobile sensors assigned to its cluster.

The location of the sensors is assumed to be final in terms of their sensing assignment; in other words, from the point of view of the application (i.e., functional requirements), the sensors are already placed in the desired positions, e.g. though an initial self-deployment phase [20, 22, 39]. This means that if a sensor decides to move (e.g., to go to a service station to recharge its battery), it might create a sensing hole.

It is assumed that sensors can determine their own positions by using some localization method (e.g., GPS). Sensors can communicate with other sensors within their transmission range R and they all move at the same speed. The distance to the assigned facility should be within the sensors' mobility range to guarantee a successful round-trip to the station with one battery charge.

All communications are asynchronous; there is no global clock or centralized entity to coordinate communications or actions. The communication environment is assumed to be contention and error free (i.e., no need to retransmit data) and there is no interference produced by receiving simultaneous radio transmissions (i.e., ideal MAC layer).

To receive service from a recharge station $f \in F$, a sensor $s \in S$ must request the station f for a free socket. Once the request is granted, the sensor can then move to the station and recharge at the assigned socket.

There are two basic types of strategies for the energy restoration problem in our context: passive and proactive.

In a *passive* strategy, each sensor will monitor its energy level at periodic intervals and after any operation (e.g., send, receive, etc.). When its battery reaches a critical level, it goes to its recharging station.

In a *proactive* strategy, a sensor may decide to move before its battery reaches a critical level; the general idea is that sensors will try to get closer to their service stations in order capture the so called "front seats" for when their time comes to make a trip to recharge their batteries.

3. Passive Approach to Energy Restoration

In a passive strategy, the sensors operate in two basic states: BATTERY_OK and BATTERY_LOW. Once the battery level falls below a predefined threshold (state BATTERY_LOW), which is not necessarily the same for all the sensors and depends on their distance to the station, the sensor must recharge its battery.

If the recharge station is within the sensor's transmission range, the sensor can send a recharge request directly and, once the request is granted, it moves to the assigned dock of the recharging station. If the recharging station is outside the sensor's transmission range, the request can be sent using some routing mechanism to forward the recharge request message to the service station. Alternatively, the sensor could start its journey towards the recharge station and once it gets there (or at least within range) it requests an available socket.

Regardless of the mechanism chosen, the sensor-facility interactions are implemented based on the service station communication pattern shown in Figure 1. For simplicity, the pattern shows the case of a service station with only one recharge socket. The recharging process is initiated with a RECHARGE_REQUEST sent by a low battery sensor. The service station will keep a queue of received requests and a ranking based on the sensors' energy levels. When a socket becomes available, the service station sends a RECHARGE_ACCEPT to the smallest ranked sensor (i.e., lowest energy sensor). Every time a sensor recharge is completed, the sensor sends a RECHARGE_DONE message to the service station and travels back to its assigned position in the network. This process is repeated continuously.

The effectiveness of this method depends on several factors such as: number of sensors in the cluster, distance to the station, number of recharging sockets, etc. Since our ultimate goal is to achieve a point of equilibrium with minimal or no sensor losses, a new question arises: will this approach work, and if it does, at what cost? The experimental analysis section provides some of these answers.

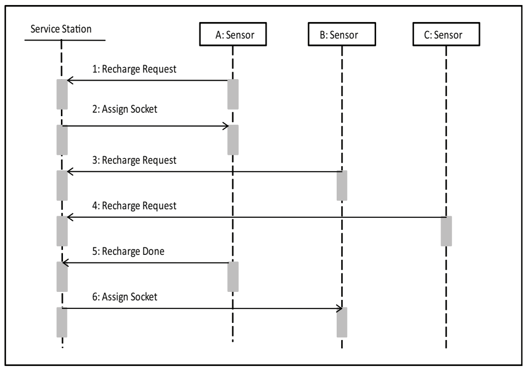


Fig. 1. Communication pattern for a mutex service station.

It is important to point out that, when sensors travel to the stations, they create temporary coverage holes. If temporary loss of coverage is an issue of paramount importance for the network, there are solutions to overcome this limitation. For instance, the service stations could be equipped with spare sensors. The number of spare sensors should be equal to the number of recharging sockets and every time a sensor is accepted (i.e., a socket becomes available), a spare is dispatched to the sensor's location to take its place. The low battery sensor is now free to travel to the base station and will eventually become a spare after its battery has been recharged.

4. Proactive Approach to Energy Restoration

In this section we examine the case when the mobile sensors decide to act before their batteries reach a critical level and a trip to the recharging station is needed. The general idea is that sensors will try to get closer to their service stations in order capture the so called "front seats" (i.e., sensor locations within one-hop distance to the station). However, the number of front seats is limited and, since the sensors have functional responsibilities in their assigned locations and any movement can create a coverage hole, changing location cannot be a unilateral decision.

r!

4.1. *Position Based Movements*

To minimize coverage holes (due to movements or to total energy depletion), a sensor with low energy will perform a gradual approach towards the recharge station, and it will do so by swapping position with higher energy sensors closer to the recharge station. The operating life of a sensor is now divided in three stages depending on its battery status: 1) BATTERY_OK or normal operation, 2) BATTERY_LOW or energy-aware operation, and 3) BATTERY_CRITICAL or recharge-required operation.

A sensor in a BATTERY_OK state will perform its regular sensing functions as well as accept any swapping proposal from other sensors with less energy. When the battery level falls below a fixed threshold, the sensor switches its state to a more active BATTERY_LOW state. In this state, the sensor will start its migration towards the recharge station, proposing a position swap to sensors with higher energy levels. Finally, a sensor in the BATTERY_CRITICAL state will contact the recharge station and wait until a socket or docking port has been secured, then it will travel to the station and recharge (see Figure 2).

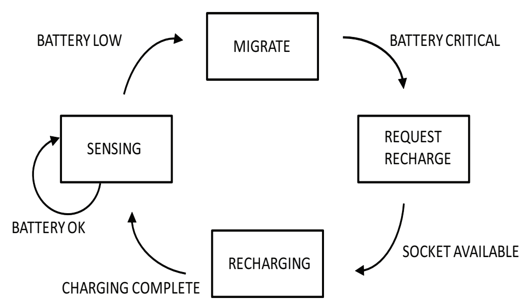


Fig. 2. A sensor's life cycle.

In this life cycle, it is the migration behaviour that is of interest. The objective of the sensor during migration is to reach the recharge facility in an effective, timely manner, while relying solely on local information.

This can be done by allowing the sensor to explore energy-aware routes leading to the recharge facility. We propose to make use of position-based routing strategies (e.g., see [40, 41]). However, instead of sending a packet that needs to be routed until it reaches the intended target, the sensors have to "route themselves" until they reach the service stations. In particular, we propose to reduce the problem of coordinating the recharging of mobile sensors to the problem of finding energy-aware routes in a logical Compass Directed Unit Graph (CDG), defined below, built on top of the original topology. The proposed graph incorporates ideas from forward progress routing techniques [40–44] and the directionality of compass routing [45] in an energy-aware unit sub-graph.

Definition 1. Given the set of sensors S in $\mathbb{R}^2$, its Unit Disk Graph (or Unit Graph) is the graph $G = (S, E)$ where $\forall s_i, s_j \in S$, $(s_i, s_j) \in E$ if and only if $d(s_i, s_j) \leq R$, where d denotes the Euclidean distance and R is the transmission range.

Definition 2. Let $G = (S, E)$ be the unit disk graph of S, and let F be a recharge facility in $\mathbb{R}^2$. The Compass Directed unit Graph (CDG) of S with respect to F is the directed graph $G' = (V', E')$, where $V' = S \cup F$ and $\forall\ v_i, v_j \in V'$, $\overrightarrow{(v_i, v_j)} \in E'$ if and only if the following conditions are satisfied:

(1) Unit graph criterion: $d(v_i, v_j) \leq R$.
(2) Proximity criterion: $d(v_j, F) < d(v_i, F)$ and $d(v_i, v_j) < d(v_i, F)$
(3) Directionality criterion: $\exists v_{jp}$ such that $v_j\vec{v}_{jp} \cdot v_i\vec{F} = 0$ and $d(v_i, v_{jp}) + d(v_{jp}, F) = d(v_i, F)$

Routing algorithms use the hop count as the metric to measure effectiveness. In our case, the hop count would be equivalent to the number of swapping operations between sensors in our CDG. Our solution to the FFP can be divided into two main stages: 1) the construction of the CDG and 2) the incremental swapping approach (i.e., migration) towards the rendezvous location.

4.2. *Creating the CDG*

An example of the proposed CDG for three sensors A,B,C and a facility F is shown in Figure 3. In this first stage, it is assumed that all sensors have the required levels of energy to construct the CDG. The process is rather simple and can be summarized by the following actions:

(1) Sensors position themselves at some initial fixed location that depends on the task at hand.
(2) Sensor A sends a NEIGHBOUR_REQUEST broadcast message inviting other sensors to participate.
(3) Upon receiving a NEIGHBOUR_REQUEST message from sensor A, immediate neighbours verify the neighbouring criteria according to the following rules:
   a) Proximity: $d(A, F) > d(B, F)$ and $d(A, B) < d(A, F)$.

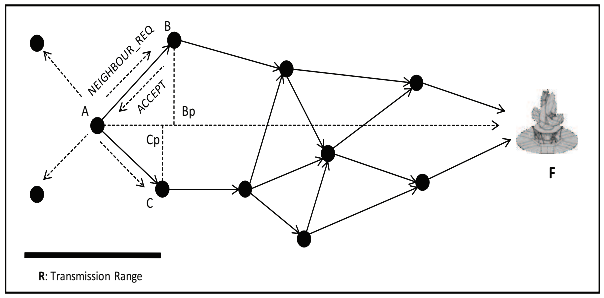


Fig. 3. Compass Directed Graph

b) Directionality: For example, B and C are neighbours of A if the corresponding projections B_p and C_p on line $\overline{AF}$ intersect the line segment $\overline{AF}$.

(4) If both conditions a) and b) are met, then sensors B and C send a NEIGHBOUR_ACCEPT message. Otherwise they send a NEIGHBOUR_DENY message.

In order to save energy, sensor A will then try to deviate as little as possible from the direction of the recharge station F. That is, sensor A will try to minimize the angle $\angle BAB_p$. Therefore, all the sensors that satisfy the conditions a) and b) are ranked according to the following function: $f(s_i, s_j) = \left\{ d(s_i, s_j) + \frac{d(s_j, s_{jp})}{d(s_i, s_j)} \right\}$ where s_i, s_j are the neighbouring sensors, d is the Euclidean distance, F is the recharge station and s_{jp} is the projection of s_j on the line segment $\overline{s_i F}$.

At the end of this phase, each sensor will have two routing tables: one containing its children (i.e., sensors from which NEIGHBOUR_ACCEPT messages were received) with their corresponding rank and a second table containing its parents (i.e., sensors to which NEIGHBOUR_ACCEPT messages were sent). The routing tables are just partial maps of the network indicating the position of the children and parents.

4.3. *Migration Strategy*

The second stage starts when sensors change their state from BATTERY_OK to BATTERY_LOW as a result of their battery levels falling below the first threshold. Once a sensor enters this state, it will try to get closer to the facility by making a series of one-hop swaps with its graph neighbours.

The swapping operation is initiated with a sensor sending a SWAP_REQUEST message to its lowest ranked neighbour. Neighbors could be ranked based on their distance (closest to farthest) and their direction relative to the target station. Another option of ranking includes the energy levels of neighbours as a metric as well as the energy levels of 2-hop neighbours (i.e., children of my children). If the current energy level of the child sensor is larger than the parent sensor, the sensor replies with a SWAP_ACCEPT message and travels to the position of the parent sensor. If its energy level is lower, it replies with a SWAP_DENY message. Once a requesting sensor has initiated the swapping process it will not entertain any SWAP_REQUEST messages until the swapping operation is completed. The swapping operation is considered atomic and once completed both sensors will send a SWAP_COMPLETE message that will be used by current and new neighbours/parents to update their routing ! tables.

The final step of this phase takes place when battery levels fall enough to trigger a change to the BATTERY_CRITICAL state. In this state, the sensors behave exactly as in the passive approach and their interaction with the service station is defined by the pattern discussed earlier. A BATTERY_CRITICAL sensor sends a RECHARGE_REQUEST message to the recharge station and waits until an available socket is assigned. Similar to the passive approach, there are two cases to consider: 1) The recharge station is within the sensor's transmission range and 2) The recharging station is outside the sensor's transmission range and lowest ranked neighbours will forward the request towards the station. If there is no routing mechanism in place, the sensor can initiate its journey (i.e., panic situation) until the station is within range.

In an ideal system, all sensors will reach the BATTERY_LOW state when they are exactly at one-hop distance from the recharge station. When the trip to the recharge station is made from a one-hop position in the graph (i.e., there are no graph neighbours), we call this "*one-hop run*" or "*optimal run*". Contrarily, if the trip is made from any other location, it is called a "*panic run*". We will come back to visit this issue when we discuss the experimental analysis of the different strategies.

There are two important properties of the CDG (i.e., dynamic and self-correcting) that can be explained by the following scenarios. Both scenarios may cause situations where the information in the neighbouring tables is obsolete.

- Scenario 1: Simultaneous swapping. As part of the swapping process, the participating sensors exchange their neighbouring information, that is, their corresponding children and parent tables. However, since multiple swapping operations may occur at the same time, when a sensor finally arrives at the position occupied by its swapping partner, the information in its neighbouring tables may be out-of-date.
- Scenario 2: Sensor recharging. While this process takes place, other sensors may be swapping positions. Once the recharging process is finished, the sensor returns to its last known position. However, the structure of the network around it has changed. This situation is even more evident when trips to the facility are made from distances of more than one hop as a result of "panic runs".

The solution to these problems is to define the neighbouring information as position-based tables, where the important factor is the relative position of the neighbours and not their corresponding IDs. The information of the actual sensors occupying the positions is dynamic. In other words, a sensor in a given position (x, y) knows that at any given point in time it has n children at positions $(x_1, y_1)...(x_n, y_n)$ and p parents at positions $(x'_1, y'_1)...(x'_p, y'_p)$. This information is static with respect to (x, y) and will not be modified. However, the identity of the sensors occupying the positions is dynamic and will get updated every time a swapping operation occurs. The mechanism to detect changes in the routing tables is triggered by sending a SWAP_COMPLETE message. When two neighbouring sensors successfully complete a swapping operation, they will announce their new positions by sending SWAP_COMPLETE messages. Sensors within the transmission range that listen to this message will verify whether any of the positions involved in the exchange belong to their routing tables and update the appropriate entry with the ID of the new occupant of that position (i.e., self-correcting property).

On the other hand, a sensor returning from the service station (e.g., scenario 2) needs to re-discover the new occupants of its routing tables. This process is initiated by a SENSOR_RECHARGED message sent by the newly recharged sensor as soon it reaches its last known position on the network. Potential children and parents, upon receiving this message, will reply with CHILD_UPDATE and PARENT_UPDATE messages accordingly. This process is also used for parents to update their information about the energy levels of this newly recharged sensor.

These two important properties, along with a neighbouring criteria that

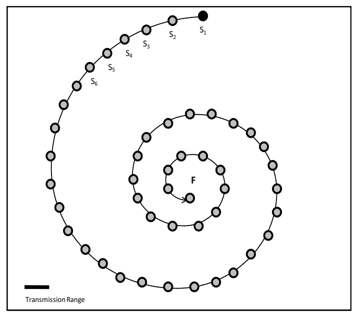


Fig. 4. Proactive strategy for Sensor s_1 in a spiral deployment.

incorporates ideas from forward progress and compass routing [40, 41, 45] in an energy-aware unit graph, ensure the following lemma:

Lemma 1. *The swapping-based proactive solution to the FFP guarantees that all sensors reach the rendezvous location within a finite number of swapping operations.*

Proof. Let $G' = (V', E')$ be the CDG of S with respect to the recharge facility F. to prove the lemma, it suffices to show that every path $P =< s_{i,1}, ..., s_{i,k}, F >$ in G' from $s_{i,1}$ to the recharge station F does not contain any cycles; that is all $s \in P$ are distinct.

By contradiction, let $s_{i,j} = s_{i,r}, j < r$; this means that $s_{i,(j+1)}$ is a child in G' of sensor $s_{i,r}$, which means that $d(s_{i,(j+1)}, F) < d(s_{i,r}, F)$. This contradicts the proximity criterion (triangular inequality). Hence, the Lemma holds. □

4.4. *Extreme Cases*

So far, the proactive strategy seems not only possible but intuitively more efficient than a passive approach. However, for some specific deployments, the proactive solution may not report any improvements over the passive approach. The deployment shown in Figure 4 shows the trajectory followed by a sensor s_1 during its migration towards the facility F.

If a proactive strategy is selected for this particular deployment, the sensor will start a gradual approach towards the facility. The sensors' limited transmission range implies that only one neighbour will be discovered during the CDG creation. That is, $\forall$ sensor $s_i \in \{s_1, s_2, ...\}$, s_{i+1} is graph neighbour (i.e., child node) of s_i (i.e., $d(s_{i+1}, F) < d(s_i, F)$) and s_{i-1} its corresponding parent. The locally-based swapping selection criteria will force sensor s_1 to exchange positions with its only available graph neighbour s_2. Consequently, s_1 will take the longest possible path to rendezvous with the facility.

In this particular example, the proactive strategy incurs an excessive and unnecessary waste of energy by a continuous sensor swapping. The sensor will eventually reach a BATTERY_CRITICAL state and will default to a passive behaviour. However, this could have been avoided by taking a passive approach and waiting in its original position until the BATTERY_CRITICAL state is reached. In this particular deployment following a passive approach would have maximized sensing time by avoiding temporary coverage holes due to unnecessary swapping operations.

5. Improving the Proactive Strategy

The number of graph neighbours has a direct impact on the performance of the proactive strategy. Having more immediate graph neighbours implies more options when exploring a greedy migration towards the recharge station but it also means more interactions, notifications, etc., as more sensors will be affected by SWAP_COMPLETE and SENSOR_RECHARGE messages. Therefore, it may be beneficial to be more selective when choosing the graph neighbours and perhaps having fewer but better selected neighbours. The problem is to determine the right number of sensors within range that should be selected as graph neighbours. Here, we have a clear trade-off between flexibility when choosing a migration path and the required maintenance overhead.

So far, low battery sensors choose their swapping partners based on the energy levels of their 1-hop graph neighbours. Enhancing sensor knowledge by adding information about the energy levels of the 2-hop graph neighbours may impact the path selection process and facilitate the migration through higher energy areas of the network. However, is more knowledge better to achieve energy equilibrium? or is this added knowledge to costly?

5.1. *Exploring Different Topologies*

In this section we evaluate the same swapping-based migration strategy presented in section 4.3 on a different underlying topology. The requirements for the new topology remain the same, and they are: 1) It should be built using local information only. 2) It should be flexible enough to operate in an asynchronous environment. 3) It should be dynamic, self-correcting. 4) Mobility strategies based on this topology should be loop-free.

For the new topologies to consider, the sensors will select their graph neighbours based on the concept of Gabriel neighbours and Relative neighbours. Two points A and B are said to be Gabriel neighbours if their diametric circle does not contain any other points. A graph where all pairs of Gabriel neighbours are connected with an edge is called the Gabriel graph. In our case, two sensors s_1 and s_2 with coordinates (x_1, y_1) and (x_2, y_2) are Gabriel neighbours if the circle with center $(\frac{x_1+y_1}{2}, \frac{y_1+y_2}{2})$ and radius $\frac{d(s_1,s_2)}{2}$ does not contain any other sensor. A particular case of a Gabriel Graph is the Relative Neighbor Graph where sensors s_1 and s_2 are relative neighbours if there are no other sensors in the *Lune* between sensors s_1 and s_2. That is, if $\forall$ S, $S \neq s_1$ and $S \neq s_2$, $d(s_1, s_2) < max\,\{d(s_1, S), d(s_2, S)\}$ where d denotes the Euclidean distance between two sensors [46, 47].

In this new scenario, where low energy sensors will select their Gabriel or Relative neighbours as the potential swapping partners, the migration strategy towards the recharge station will be based on finding energy efficient routes on a Compass Directed Gabriel Graph (CDGG) or a Compass Directed Relative Neighbor Graph (CDRNG).

Definition 3. Let $G' = (S \cup F, E')$ be the compass directed unit graph of S with respect to recharge facility F. The *Compass Directed Gabriel Graph* (CDGG) of S with respect to F is the subgraph $\hat{G} = (S \cup F, \hat{E})$ of G' where, $\forall (s_i, s_j) \in E'$, $\overrightarrow{(s_i, s_j)} \in \hat{E}$ if and only if $\nexists s_k \in S$ such that $d(s_k, \frac{s_i+s_j}{2}) < d(s_i, \frac{s_i+s_j}{2})$

5.2. *Creating the CDGG and CDRNG*

Figure 5 shows an example of the proposed CDGG for three sensors A,B,C and a facility F. In the first stage of the algorithm, it is assumed that all sensors have the required levels of energy to construct the CDGG. The process is similar to the crearion of the CDG presented in section 4.2. However, to guarantee that only the Gabriel neighbours are selected as graph neighbours, the sensor should implement the following actions:

(1) Upon receiving a NEIGHBOUR_ACCEPT message from a potential Gabriel neighbour S', the receiving sensor S verifies if there is already a graph neighbour in the disc with center $(\frac{S_x+S'_x}{2}, \frac{S_y+S'_y}{2})$ and radius $\frac{d(S,S')}{2}$. If such a neighbour exists, then sensor S sends a NEIGHBOUR_DENY message to S'.

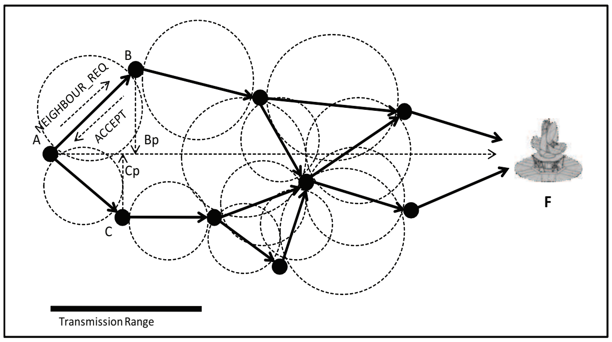


Fig. 5. Compass Directed Gabriel Graph

(2) If no existing graph neighbour is found in the previous step, this means that sensor S' is in fact a Gabriel neighbour. However, some of the existing graph neighbours could be affected by this newly accepted sensor and they are no longer Gabriel neighbours. If the newly accepted sensor S' falls in the diametric disc between sensor S and one of the existing graph neighbours S_i, the neighbour in question should be excluded by sending it a NEIGHBOUR_DENY message.

5.3. *Increasing Sensor Knowledge*

Another possible enhancement to improve the overall performance of the proactive strategy and help low energy sensors reach the recharge station faster is to add additional information about the energy levels of the 2-hop graph neighbours. Regardless of the topology chosen (i.e., CDG, CDGG, or CDRNG), having the 2-hop neighbouring information combined with the 1-hop greedy strategy should lead to a more energy efficient path selection. To implement this new approach, a series of changes to the existing algorithms is necessary. For example, the neighbouring information stored by each sensor s needs to change to include the tuple $(s_i, E_{S_i}, E_{S_{i_{2hop}}})$ where s_i is the *i-th* 1-hop neighbour of s. E_{S_i} represents the energy level and $E_{S_{i_{2hop}}}$ represents the average energy levels of the 1-hop graph neighbours of s_i.

The information about existing 1-hop graph neighbours will be appended to the NEIGHBOUR_ACCEPT messages sent during the graph creation phase. When a sensor sends a NEIGHBOUR_ACCEPT message to its parent, the message will now include the average energy level of its existing 1-hop neighbours. This new piece of information will have to be up-

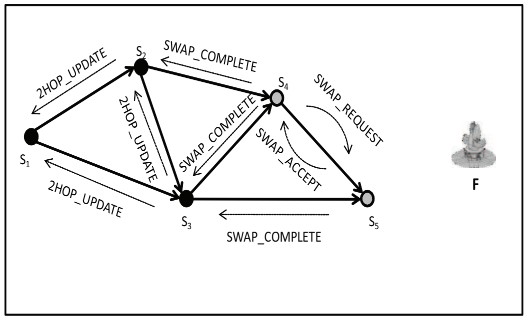


Fig. 6. Sensor swapping with 2-hop neighbours updates.

dated once the migration or swapping phase is initiated. Consequently, two swapping sensors will exchange this new piece of information as part of the swapping process. Furthermore, sensors reacting to a SWAP_COMPLETE message will generate a new message NEIGHBOUR_2HOP_UPDATE to inform their parents about the changes of their 2-hop graph neighbours.

Let us examine the example shown in Figure 6 to illustrate the new interactions required during a swapping operation. In this example, sensors S_4 and S_5 have agreed to swap positions after the corresponding exchange of SWAP_REQUEST and SWAP_ACCEPT messages. Once the sensors arrive at the location occupied by their swapping partners, both sensors (i.e., S_4 and S_5) will send SWAP_COMPLETE messages to their parents S_2 and S_3. The SWAP_COMPLETE message received by sensor S_2 contains the tuple $(S_4, E_{S_4}, E_{S_{4_{2hop}}})$. After updating its neighbouring information with the newly received information, S_2 computes the combined energy level of its 1-hop graph neighbours: $E_{S_{2_{2hop}}} = \frac{E_{S_3}+E_{S_4}}{2}$ and sends a new NEIGHBOUR_2HOP_UPDATE
$(S_2, E_{S_4}, E_{S_{2_{2hop}}})$ message to its parent S_1.

It is clear from the previous example that for each successful swapping operation there will be an overhead produced by the new NEIGHBOUR_2HOP_UPDATE messages. The density of the graph, determined by the neighbour selection criteria and the sensor transmission ranges, will have a great impact on how many of these new notification messages are generated. The next section examines the impact of this added knowledge, its relationship with the underlying topology chosen, its potential benefits and possible drawbacks.

6. Experimental Results

Previous work on energy consumption of wireless sensor networks and protocols such as 802.11, have found that the energy required to initiate communication is not negligible. In particular, loss of energy due to retransmissions, collisions and acknowledgments is significant [48, 49]. Protocols that rely on periodic probe messages and acknowledgments are considered high cost. It is also noted in the literature that sensors' energy consumption in an idle state can be as large as the energy used when receiving data [49]. On the other hand, the energy used in transmitting data could be between 30-50% more than the energy needed to receive a packet.

A common consideration for any solution involving mobile entities is how to accurately represent the energy spent when moving from one location to another. Locomotion cost depends on many factors such as the weight of the electronic components, irregularities in the terrain, obstacles, etc. For simplicity, in [26, 33], the weighted Euclidean distance between origin and destination is used as the cost of relocating a robot. In particular, in [33] is observed that the energy required to move their robotic sensors was 54x the energy required to send a packet over the same distance and the energy spent in communications (i.e., send/receive) was 25% more than the battery drain in the idle state.

6.1. *Experimental Environment and Performance Criteria*

The different scenarios are implemented in Omnet++ [50] along with the mobility framework extension [51]. For all experiments, the sensors and charging facilities were randomly placed in an area of 1000x1000m^2. The analysis of our simulated results centers on two important aspects of the solutions: 1) Whether or not a state of equilibrium is achieved and the number of sensor losses until such condition is met; 2) Impact of several variables such as: underlying topology, transmission range, number of recharge sockets/ports and sensor knowledge.

In an ideal system, all sensors will reach the BATTERY_CRITICAL state when they are exactly at one-hop distance from the rendezvous location. When the trip to the recharge station is made from a one-hop position (i.e., there are no graph neighbors), it is called a "one-hop run" or "optimal run". Contrarily, if the final trip is made from any other location, it is called a "panic run" [32]. In all the simulated scenarios, the quality of the strategy is measured in terms of optimal runs vs. panic runs. Con-

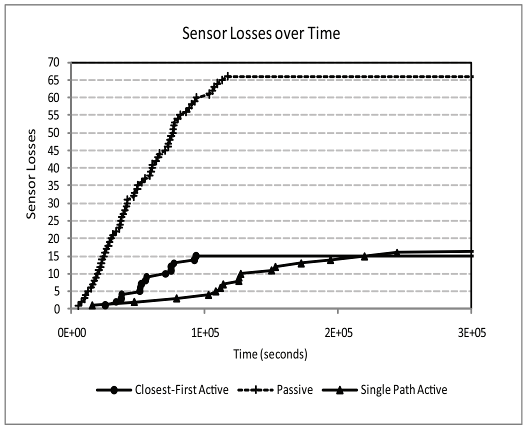


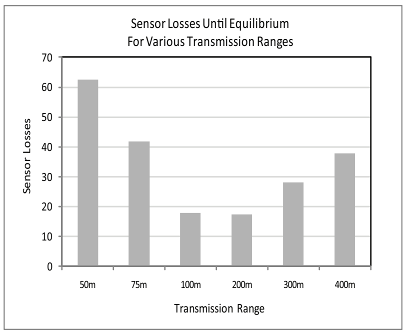
Fig. 7. Passive Strategy vs. Proactive Strategies

stant cost values are assigned to each basic operation (i.e., send, receive, idle and move). Initial values for these operations are based on some of the observations found in [33, 48, 49].

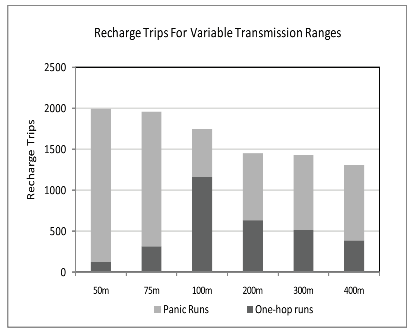
6.2. *Passive vs. Proactive*

The first set of experiments attempt to find out how the proactive strategies perform when compared to a passive approach. The goal is to measure the number of sensor losses due to battery depletion over time until the system reaches a state where no further sensor losses are reported (i.e., state of energy equilibrium). In this context, several proactive strategies are examined: 1) The closest-first strategy, where sensors attempt to make forward progress by swapping positions with the closest neighbour and 2) Single path strategy, where sensors select a single graph neighbour (e.g., first discovered).

The results of an experiment involving 100 sensors and one service facility are shown in Figure 7. The facility is equipped with two sockets, allowing two sensors to be recharged simultaneously. A series of 30 tests with different random deployments are run for 10^6 simulation seconds. The sensor transmission range is fixed at 100m and the energy ratio for sending/receiving a packet is set to a constant ($E : E/2$). Locomotion costs were based on the weighted Euclidean distance with a weight factor of $\frac{1}{5}E$ per meter traveled. The results show that the two variations of the proactive strategy reached the state of equilibrium. This means that all the energy



(a)



(b)

Fig. 8. Experimental results: Variable Range

spent during the graph creation, swapping and graph reconfiguration in a network with a 100:1 sensor-facility ratio with two sockets did not overwhelm the system to the point of preventing it from reaching equilibrium. Surprisingly, even the single path proactive strategy outperfor! med the passive approach by a significant margin.

In comparison, for a similar network size, the solutions presented in [11] and [19] required 2 and 3 stations, or actors, respectively, to maintain a live network (i.e., 50% or more sensors remain after equilibrium was reached). In our case, equilibrium was achieved with 1 facility with two docking ports for a similar network size and over 80% of network survivability.

Even though the passive strategy reached the state of equilibrium faster than the single-path proactive strategy, the cost in terms of sensor losses was very high. This result implies that if a passive approach is chosen for high sensor-facility ratio deployments, the number of recharge sockets in this experiment is too restrictive. This result is similar to the passive approach followed in [19] where it was observed a significant improvement by adding a second recharge station.

6.3. *Transmission Range*

This experiment was designed to verify the impact of the sensor's transmission range on the overall performance. The characteristics of the network were the same as the test performed in section 6.2. The only difference is that the transmission range was varied from 50m, 75m, 100m, 200m, 300m and 400m.

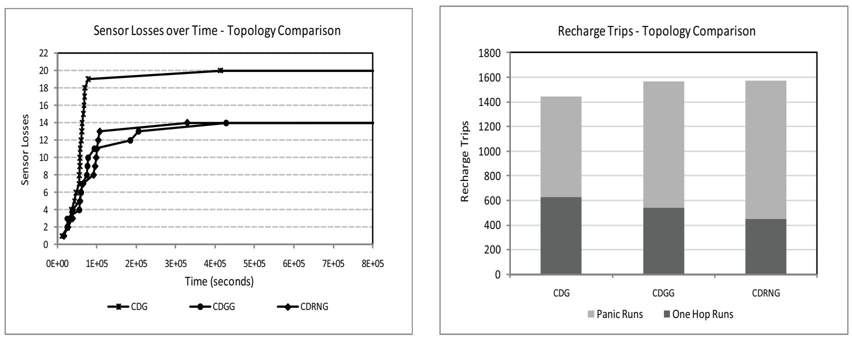
Figure 8(a) shows the cumulative number of sensor losses until equilibrium for each range value. In a deployment of 1000x1000m^2 a transmission range of 50m was too restrictive, which means that most of the sensors were isolated and the number of immediate neighbours in the CDG was too small to guarantee a gradual approach towards the recharge location. Another interesting observation is that by increasing the transmission range, the number of losses decreased dramatically. However, for larger ranges (e.g., 300m and 400m), there was a decline on the overall performance since many neighbours are discovered, resulting in an added overhead to maintain more information per sensor as well as additional interactions due to update messages as a result of successful swapping and recharging operations.

Figure 8(b) shows the quality of the solution in terms of one-hop runs vs. panic runs. In an ideal system, our solution should reach the state of equilibrium using one-hop runs only. As expected, for a transmission range of 50m, most of the trips could be considered panic runs since there is almost no migration due to the lack of 1-hop neighbours. The best breakdown between one-hop and panic runs occurs with 100m range. However, there are more visits to the recharge location, when compared to the 200m, 300m and 400m cases. Although there is no clear explanation for this phenomenon, one can argue that there is a trade-off between the total number of recharge trips and the breakdown between one-hop vs. panic runs. In a panic run situation, a sensor travels from a more distant location and after having been recharged, it needs to return farther to its initial location. This situation creates a coverage hole that lasts longer than holes created by! one-hop runs. However, more one-hop recharge trips also means more coverage holes but for shorter periods of time.

6.4. *Topology Comparison*

This test was designed to determine whether our proactive solution to energy restoration reaches a state of equilibrium when the new proposed CDGG and CDRNG are used as the underlying topologies for the mobility strategies. The experiment measured the cumulative number of sensor losses until energy equilibrium is reached.

Figure 9(a) shows the result of simulations on the same network (see 6.2) involving 100 sensors with fixed transmission range of 100m and one service facility. For all the tests performed on the three different topologies, the mobility strategy selected was the greedy closest-first swapping where a low energy sensor chooses its closest graph neighbour as a swapping partner during its migration towards the recharge station.



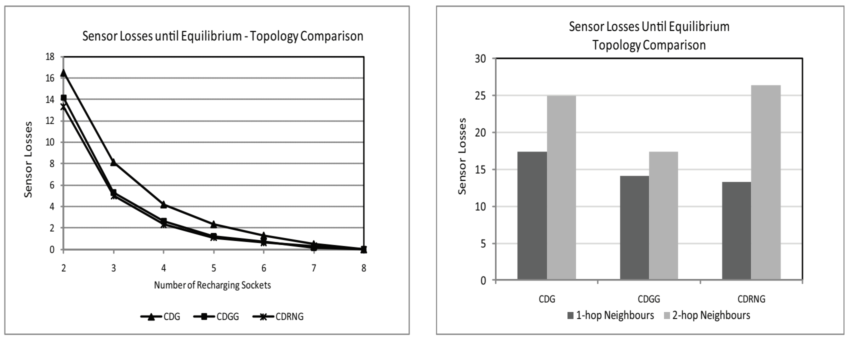
(a) (b)

Fig. 9. Experimental results: Topology Comparison

As expected, the closest-first swapping strategy on the three topologies chosen (i.e., CDG, CDGG and CDRNG) reached the state of equilibrium. The CDGG and CDRNG are sub-graphs of the CDG and according to the experimental results presented in Chapter 6.2, even the single path (i.e., single neighbour) approach reached the state of equilibrium. However, the interesting finding is that although the three topologies reached the state of equilibrium at the same time approximately, the CDGG and CDRNG reported fewer sensor losses due to battery depletion. This is an important observation that implies that fewer but better selected graph neighbours will yield better results if the main goal is to minimize the number of permanent failures due to battery depletion.

Unfortunately, the CDGG and CDRNG did not report any improvements in terms of optimal trips to the recharge station. Figure 9(b) shows the number of recharge trips and breakdown between optimal and panic runs for the three topologies in question. For the CDGG and CDRNG there was a small increase in the number of recharge visits compared to the CDG and a small decrease in the number of optimal runs. This decrease is somehow expected since the number of neighbours for both topologies (i.e., CDGG and CDRNG) is more restrictive than the CDG. Once more, choosing different topologies for the migration strategy exposed a trade-off between permanent coverage holes due to battery depletion and more short-lived temporary holes due to more frequent visits to the facility.

The next part of this test was designed to measure the impact of the recharge sockets on the cumulative number of losses until equilibrium and verify whether the perfect equilibrium can be reached by increasing the



(a) (b)

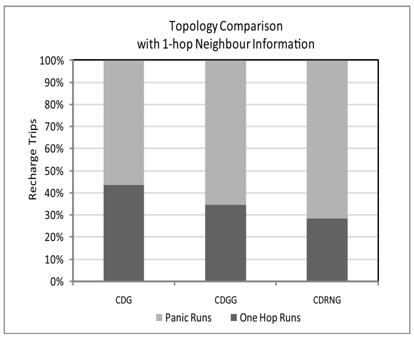
Fig. 10. Experimental results: Topology Comparison II

number of sockets or docking ports in the recharge station. The network setup remained the same and the closest-first greedy mobility strategy was tested on the three topologies (i.e., CDG, CDGG and CDRNG). Figure 10(a) shows the result for this test where the closest-first swapping strategy on the three topologies showed the same progression towards perfect equilibrium. The total number of recharge sockets needed for the perfect equilibrium is the same for the three topologies but the CDGG and CDRNG showed an improvement on the number of sensor losses over the CDG as the number of recharge sockets increased.

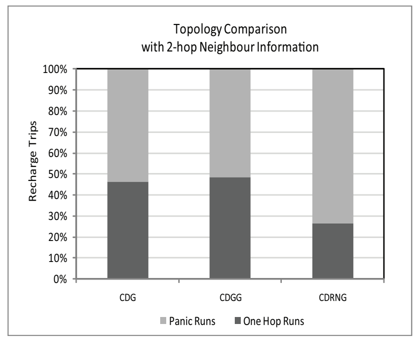
6.5. *Sensor Knowledge*

The goal of this set of tests is to verify the impact of added sensor knowledge, as introduced in Section 5.3, and compare it with the 1-hop information greedy strategies on the three proposed topologies. The network parameters are the same as in the previous tests, with fixed transmission range at 100m. The closest-first swapping strategy is applied on the three topologies (i.e., CDG, CDGG and CDRNG) with information about the energy levels of 1-hop graph neighbours only and 2-hop graph neighbours respectively.

Figure 10(b) shows the number of sensor losses until equilibrium for the three topologies tested with 1-hop neighbour information vs. 2-hop neighbour information. In each case, there was an increase in the number of sensor losses when the migration strategy included the 1-hop neighbour information. When 2-hop information is used, the best performer was the



(a)



(b)

Fig. 11. Experimental results: Sensor Knowledge

CDGG with losses similar to the 1-hop CDG. This is a rather surprising result, which seems to imply that "knowing more individually" about the network is less useful for the collective effort than "knowing less". Knowing more in this case has a direct impact on the number of control messages required to maintain the underlying topology in a consistent state. This phenomenon will be more evident as the graph degree increases. The graph maintenance overhead related to keeping 2-hop neighbour information proved to be crucial to the point that counteracts any possible improvement when compared ! to keeping 1-hop information only.

The idea of adding extra knowledge to the sensors aimed to improve the path selection strategy and increase the number of optimal runs or 1-hop trips to the recharge station. The simulation results shown in Figure 11(b) confirmed our expectations. Added knowledge had, in fact, a positive impact on the selection of the better energy-efficient migration strategy towards the recharge station. There was some marginal improvement on the number of optimal runs for the CDG and CDRNG with a real improvement for the CDGG. The CDGG proved again to be the best performing topology in terms of cumulative sensor losses until equilibrium and breakdown between panic and optimal runs when using 2-hop neighbour information.

7. Closing Remarks and Open Problems

Throughout this chapter we have focused on the problem of energy restoration in a mobile sensor network with static recharging facilities. The *perfect* energy restoration strategy should be able to guarantee a continuous operation of the network without any losses; however in reality some sensor losses will occur. A *successful* energy restoration strategy is one in which the losses are limited in scope and time; that is, the strategy allows the network to reach a state of equilibrium, where no further sensor losses will occur (thus guaranteeing a continuous operation of the network), with the minimum number of sensor losses.

We have examined the basic approaches, passive and proactive, to energy restoration strategies. To compare the quality of the approaches, passive vs proactive, and the difference between different proactive mobility strategies we focused our analysis on several key indicators, such as the number of sensor losses until equilibrium is reached, the distance traveled to reach the recharge station (i.e., optimal runs vs. panic runs), and the amount of resources needed to achieve a perfect equilibrium (i.e., without any loss ever). The analysis shows the definite advantages of taking a proactive approach to energy restoration.

In terms of proactive strategies, the problem of coordinating the recharging of mobile sensors has been reduced to the problem of finding optimal routes in a logical Compass Directed Graph (CDG) or Compass Directed Gabriel Graph (CDGG) built on top of the original deployment. All the proactive solutions analyzed here have three important properties [32, 33]: 1) The proposed graphs guarantee that sensors reach the recharge facilities in a finite number of swapping operations (the trajectory is loop-free). 2) All decisions made by the sensors regarding the next swapping operation are based on local knowledge (i.e., the algorithms are completely distributed and localized). 3) New sensors can be added or deleted at any time and new neighbours are re-discovered any time a successful swapping or recharge operation takes place, making the graphs dynamic and self-correcting.

Many important problems are open and need to be addressed. Some of them are listed in the following.

The proactive strategies proposed in [32, 33] and examined in this Chapter assume that there are no *obstacles* between the sensors and the chosen recharge facility. However, more challenging environments may contain static obstacles that prevent the sensors from communicating with other sensors or traveling directly to the recharge station. The presence of obstacles in static sensor networks has been the focus of attention in several research papers. For example, in [52] a model for obstacles in static sensor networks is discussed, where obstacles are distinguished into physical and communication ones. A physical obstacle is a network area which prevents the deployment and movement of sensors in that area.

A communication obstacle, on the other hand, causes a disruption to the wireless communication: if the line of sight between two sensors crosses the obstacle, then there is no communication between those sensors. The problem of how to successfully route packets around obstacles has been examined in [53–55]. In mobile sensor networks, this problem has not yet been examined; clearly, the obstacle avoidance strategies have a higher degree of complexity since the algorithms have to guarantee not only communication but also movement around the obstacles.

There are several unexplored variants of the energy restoration problem. For example, let $P = \{p_1, ..., p_K\}$ a set of points of interests and f, a static recharge facility. In this variant, each sensor s must visit an assigned set of points $P_s \subset P$ and repeat this process continuously; in other words, each sensor has to visit a sub-set of points of interest continuously and also visit the facility periodically to recharge its battery. Within this variant, there a number of alternatives: 1) Ordered vs. non-ordered points of interests: the sensor has to visit the assigned set of points in a given order or in an arbitrary order (but always guaranteeing the all points are visited before starting the next round of visits). 2) Disjoint vs overlapping routes or itineraries: $P_{s_i} \cap P_{s_j} = \emptyset, \forall s_i, s_j \in S$ or $P_{s_i} \cap P_{s_j} \neq \emptyset$; in the latter case, two sensors must not visit the same point at the same time. 3) Fixed vs. exchangeable itineraries or point of interest: the assigned itineraries are fixed, or sensor can exchange their itineraries or "pick-up" other sensors' point of interest (e.g., if a sensor dies of energy starvation, another sensor can add the depleted sensor's itinerary to its own).

For all these variants, the main goal is the same as the original problem: to achieve a state of equilibrium where, without any further losses, the sensors fulfill their tasks but also cooperate to share a recharge station with limited resources (i.e., number of recharge sockets).

References

[1] G. Wang, G. Cao, T. L. Porta, and W. Zhang, Sensor relocation in mobile sensor networks, *In Proceedings of IEEE INFOCOM*. pp. 2302–2312 (2005).

[2] X. Li, N. Santoro, and I. Stojmenovic, Mesh-based sensor relocation for coverage maintenance in mobile sensor networks, *In Proceedings of the 4th international conference on Ubiquitous Intelligence and Computing*. pp. 696–708 (2007).

[3] J. Luo and J. P. Hubaux, Joint mobility and routing for lifetime elongation in wireless sensor networks, *In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Infocom 2005).*

pp. 1735–1746 (2005).

[4] W. Wang, V. Srinivasan, and K. Vikram, Extending the lifetime of wireless sensor networks through mobile relays, *IEEE/ACM Transactions on Networking*. **16**, 1108–1120 (2008).

[5] M. Rahimi, H. Shah, G. Sukhatme, J. Heidemann, and D. Estrin, Studying the feasibility of energy harvesting in a mobile sensor network., *In Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 19–24 (2003).

[6] S. Roundy, P. Otis, Y. Chee, J. Rabaey, and P. Wright, A 1.9ghz rf transmit beacon using environmentally scavenged energy., *Digest IEEE International Symposium on Low Power Electricity and Devices* (2003).

[7] X. Li, A. Nayak, and I. Stojmenovic, Exploiting actuator mobility for energy-efficient data collection in delay-tolerant wireless sensor networks., *In Proceedings of the Fifth International Conference on Networking and Services ICNS*. pp. 216–221 (2009).

[8] Y. Mei, C. Xian, S. Das, Y. Hu, and Y. Lu, Sensor replacement using mobile robots., *Computer Communications*. **30**, 2615–2626 (2007).

[9] D. Simplot-Ryl, I. Stojmenovic, and J. Wu, Energy efficient backbone construction, broadcasting, and area coverage in sensor networks., *Chapter 11. Handbook of Sensor Networks: Algorithms and Architectures*. pp. 343–379 (2005).

[10] T. Tirta, B. Lau, N. Malhotra, S. Bagchi, L. Z., and Y. Lu, Controlled mobility for efficient data gathering in sensor networks with passively mobile robots., *Sensor Network Operations by Wiley-IEEE Press.* (2005).

[11] M. Sharifi, S. Sedighian, and M. Kamali, Recharging sensor nodes using implicit actor coordination in wireless sensor actor networks, *In Wireless Sensor Network*. **2**, 123–128 (2010).

[12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications*. **1**(4), 660–669 (2002).

[13] C.-H. Lung, C. Zhou, and Y. Yang, Applying hierarchical agglomerative clustering to wireless sensor networks, *In Proceedings of the International Workshop on Theoretical and Algorithmic Aspects of Sensor and Ad-hoc Networks (WTASA)*. pp. 97–105 (2007).

[14] M. Perillo, C. Zhao, and W. Heinzelman, On the problem of unbalanced load distribution in wireless sensor networks, *In Proccedings of the Global Telecommunications Conference Workshops (GlobeCom 2004)*. pp. 74–79 (2004).

[15] O. Younis and S. Fahmy, Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, *IEEE Transactions on Mobile Computing*. **3**, 366–379 (2004).

[16] F. Michaud and E. Robichaud, Sharing charging stations for long-term activity of autonomous robots, *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. **3**, 2746–2751 (2002).

[17] J. Warwerla and R. Vaughan, Near-optimal mobile robot recharging with the rate-maximizing forager, *In Proceedings of the 9th European Conference*

on Artificial Life. pp. 776–785 (2007).

[18] F. Arwin, K. Samsudin, and A. R. Ramli, Swarm robots long term autonomy using moveable charger, *In Proceedings of the 2009 International Conference on Future Computer and Communication*. pp. 127–130 (2009).

[19] A. Drenner and N. Papanikolopoulos, Docking station relocation for maximizing longevity of distributed robotic teams, *In Proceedings of the 2006 IEEE International Conference on Robotics and Automation*. pp. 2436–2441 (2006).

[20] A. Howard, M. J. Mataric, and G. S. Sukhatme, Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem, *In Proceedings of DARS*. pp. 299–308 (2002).

[21] G. Wang, G. Cao, and T. Porta, Movement-assisted sensor deployment, *IEEE Trans. Mobile Computing*. **5**, 640–652 (2006).

[22] X. Li, H. Frey, N. Santoro, and I. Stojmenovic, Strictly localized sensor self-deployment for optimal focused coverage, *IEEE Transactions on Mobile Computing*. **10**, 1520 – 1533 (2011).

[23] Y. Toksoz, J. Redding, M. Michini, B. Michini, J. P. How, M. Vavrina, and J. Vian, Automated battery swap and recharge to enable persistent uav missions, *AIAA Infotech, Aerospace Conference* (2011).

[24] L.P.Koh and S. A. Wich, Dawn of drone ecology: low-cost autonomous aerial vehicles for conservation, *Tropical Conservation Science*. **5**, 121–132 (2012).

[25] P. McGillivary, K. Rajan, J. de Sousa, F. Leroy, and R. Martins, Integrating autonomous underwater vessels, surface vessels and aircraft as persistent surveillance components of ocean observing studies, *In Proceedings of the 2012 IEE/OES Autonomous Underwater Vehicles (AUV) Conference*. pp. 1–5 (2012).

[26] Y. Litus, R. Vaughan, and P. Zebrowski, The frugal feeding problem: energy-efficient, multi-robot, multi-place rendezvous, *In Proceedings of the 2007 IEEE International Conference on Robotics and Automation*. pp. 27–32 (2007).

[27] Y. Litus, P. Zebrowski, and R. T. Vaughan, A distributed heuristic for energy-efficient multirobot multiplace rendezvous, *IEEE Transactions on Robotics*. **25**, 130–135 (2009).

[28] A. Couture-Beil and R. Vaughan, Adaptive mobile charging stations for multi-robot systems, *In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 1363–1368 (2009).

[29] M. I. Afzar, W. Mahmood, and A. H. Akbar, A battery recharge model for wsns using free-space optics (fso), *In Proceedings of the 12th IEEE International Multitopic Conference*. pp. 272–277 (2008).

[30] M. I. Afzar, W. Mahmood, S. M. Sajid, and S. Seoyong, Optical wireless communication and recharging mechanism of wireless sensor network by using ccrs, *International Journal of Advance Science and Technology*. **13**, 59–68 (2009).

[31] Y. Peng, Z. Li, W. Zhang, and D. Qiao, Prolonging sensor network lifetime through wireless charging, *In Proceedings of the 31st IEEE Symposium on Real-Time Systems (RTSS)*. pp. 129–139 (2010).

[32] E. Velazquez and N. Santoro, Mobility-based strategies for energy restoration in wireless sensor networks, *In Proceedings of the 6th International Conference on Mobile Ah-hoc and Sensor Networks (MSN 2010)*. pp. 161–168 (2010).

[33] E. Velazquez, N. Santoro, and M. Lanthier, Pro-active strategies for the frugal feeding problem in wireless sensor networks, *In Proceedings of the 2nd International ICST Conference on Sensor Systems and Software (S-CUBE 2010)*. pp. 189–204 (2010).

[34] P. Hebden and A. Pearce, Distributed asynchronous clustering for self-organisation of wireless sensor networks, *In Proceesings of the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP 2006)*. pp. 37–42 (2006).

[35] L. Tan, Y. Gong, and G. Chen, A balanced parallel clustering protocol for wireless sensor networks using k-means techniques, *In Proceedings of the 2nd International Conference on Sensor Technologies and Applications (SENSORCOMM 2008)*. pp. 300–305 (2008).

[36] K. Xu, Y. Jia, and Y. Liu, A novel hierarchical clustering routing algorithm for wireless sensor networks, *In Proceedings of the 2008 International Conference on Internet Computing in Science and Engineering*. pp. 282–285 (2008).

[37] L. X. Zhang, An efficient energy adaptive clustering leach in wireless sensor network, *Key Engineering Materials*. **439**, 510–515 (2010).

[38] C.-H. Lung and C. Zhou, Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach, *Ad Hoc Networks*. **8**(3), 328–344 (2010).

[39] G. Wang, G. Cao, and T. L. Porta, Movement-assisted sensor deployment, *IEEE Transactions on Mobile Computing*. **5**(6), 640–562 (2006).

[40] H. Frey, S. Ruhrup, and I. Stojmenovic, Routing in wireless sensor networks, *Chapter4. Guide to Wireless Sensor Networks*. pp. 81–111 (2009).

[41] I. Stojmenovic and X. Lin, Power-aware localized routing in wireless networks, *IEEE Transactions on Parallel and Distributed Systems*. **12**, 1122–1133 (2001).

[42] T. Z.-J. ation, W. Yi, and G. Zheng-Hu, Eegfgr: An energy-efficient greedy-face geographic routing for wireless sensor, *In Proceedings of the 2007 IFIP international conference on Network and parallel computing*. **4672**, 171–182 (2007).

[43] S. Ruhrup, H. Kalosha, A. Nayak, and I. Stojmenovic, Message-efficient beaconless georouting with guaranteed delivery in wireless sensor, ad hoc, and actuator networks, *IEEE/ACM Transactions on Networking*. **18**, 95–108 (2010).

[44] M. Z. Zamalloa, K. Seada, B. Krishnamachari, and A. Helmi, Efficient geographic routing over lossy links in wireless sensor networks, *ACM Transactions on Sensor Networks*. **4**, 1–33 (2008).

[45] E. Kranakis, H. Singh, and J. Urrutia, Compass routing on geometric networks, *In Proceedings of the 11th Canadian Conference on Computational Geometry*. pp. 51–54 (1999).

[46] K. Supowit, The relative neighborhood graph, with an application to minimum spanning trees, *Journal of the ACM*. **30**, 428–448 (1983).

[47] J. W. Jaromczyk and G. T. Toussaint, Relative neighborhood graphs and their relatives, *In Proceedings of the IEEE*. **80**, 1502–1517 (1992).

[48] L. Feeney, An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks, *Mobile Network Applications*. **6**, 239–249 (2001).

[49] L. Feeney and M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, *In Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Infocom 2001)*. **3**, 1548–1557 (2001).

[50] A. Vargas, The omnet++ discrete event simulation system., *In Proceedings of the European Simulation Multi-conference (ESM'2001)*. pp. 319–324 (2001).

[51] W. Drytkiewicz, S. Sroka, and V. Handziski, A mobility framework for omnet++., *In Proceesings of the 3rd International OMNeT++ Workshop* (2003).

[52] I. Chatzigiannakis, G. Mylonas, and S. Nikoletseas, Modeling and evaluation of the effect of obstacles on the performance of wireless sensor networks, *In Proceedings of the 37th Annual Symposium on Simulation (ANSS'06)*. pp. 50–60 (2006).

[53] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, *Wireless Networks*. **7**, 609–616 (2001).

[54] C. Y. Chang, C. T. Chang, Y. C. Chen, and S. C. Lee, Active route-guiding protocols for resisting obstacles in wireless sensor networks, *IEEE Transactions on Vehicular Technology*. **59**, 4425–4442 (2010).

[55] E. Hamouda, N. Mitton, B. Pavkovic, and D. Simplot-Ryl, Energy-aware georouting with guaranteed delivery in wireless sensor networks with obstacles, *Journal of Wireless Information Networks*. **16**, 142–153 (2009).