Computer Communications 33 (2010) 409–419

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Improving network lifetime with mobile wireless sensor networks

Yinying Yang, Mirela I. Fonoage, Mihaela Cardei *

Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, United States

ARTICLE INFO

Article history:
Available online 17 November 2009

Keywords:
Wireless sensor networks
Mobile sensors
Mobile sinks
Energy efficiency

ABSTRACT

Sensors are used to monitor and control the physical environment. In mobile sensor networks, nodes can self-propel via springs, wheels, or they can be attached to transporters, such as vehicles. Sensors have limited energy supply and the sensor network is expected to be functional for a long time, so optimizing the energy consumption to prolong the network lifetime becomes an important issue. In static sensor networks, if sensors are uniformly deployed, sensors near the sinks die first. This is because besides sending their own sensed data, they also participate in forwarding data on behalf of other sensors located farther away from the sink. This uneven energy consumption results in network partitioning and limitation of the network lifetime. In this paper, we survey mechanisms that utilize nodes' mobility to extend the network lifetime.

We divide these mechanisms into three groups: mechanisms using mobile sinks, mechanisms using mobile sensors redeployment, and mechanisms using mobile relays. Using mobile sinks, energy is saved by using shorter multi-hop data delivery paths and the set of sensors located near a sink changes over time, thus the energy consumption is balanced in the whole network. Using mobile sensors, the initial deployment can be improved through sensor relocation such that to balance energy consumption and to extend network lifetime. Mobile nodes can also be used as relays, which can inherit the responsibilities of the co-locating static sensors or they can carry data to the sink to reduce the cost of long distance communication. We provide overviews and comparisons among different mechanisms.

1. Introduction

Sensors are used to monitor and control the physical environment. A Wireless Sensor Network (WSN) is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it [1,8]. Sensor nodes measure various parameters of the environment and transmit data collected to one or more sinks, using hop-by-hop communication. Once a sink receives sensed data, it processes and forwards it to the users. In mobile sensor networks, sensors can self-propel via springs [6], wheels [9], or they can be attached to transporters, such as robots [9] and vehicles [16].

Sensors are usually battery powered, for example, the Berkeley mote [29] is powered by two AA batteries. In general, sensors are left unattended after the initial deployment and it is difficult to recharge them. It will take a limited time before they deplete their energy and become nonfunctional. A sensor network is usually expected to be functional for several months or one year without recharging [15,26]. Optimizing energy consumption to prolong network lifetime is an important issue in WSNs.

Consider a static sensor network deployed for periodic data reporting. If sensors are uniformly deployed, then the sensors near the sinks consume more energy than those deployed in other parts of the monitored area and will die first. This is because besides sending their own sensed data, they also participate in forwarding data on behalf of other sensors that are farther away from the sink and thus they will deplete their energy more quickly [17,20]. This uneven energy consumption will cause energy holes in the monitored area, resulting in network partitioning. In this case, the sensed data cannot be successfully delivered to the sink. The lifetime of the sensors close to the sink becomes the bottleneck for the network lifetime [17,20]. One way to extend the network lifetime is to exploit the node mobility in mobile WSNs such that to balance the energy consumption.

In this paper, we survey mechanisms which utilize mobility to improve network lifetime. We classify the mechanisms into three categories: mechanisms using mobile sinks, mechanisms using mobile sensors redeployment, and mechanisms using mobile relays.

Using mobile sinks, sensors could communicate with a sink when it gets closer, thus using shorter hop-by-hop data delivery paths. Mobile sinks can also change their location when the nearby sensors' energy becomes low [19]. In this way, the set of sensors located near sinks change over time, the energy consumption is balanced, and the network lifetime is prolonged.

---

* Corresponding author. Tel.: +1 561 297 3459.
E-mail address: mihaela@cse.fau.edu (M. Cardei).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.11.010

A large number of sensors can be distributed in mass by scattering them from airplanes, rockets, or missiles [1]. In this case, the initial deployment is difficult to control. However, a good deployment is essential for longer network lifetime. Mobile sensors could relocate after the initial deployment before the data sensing and transmission begin such that to achieve a desired density requirement and to reduce the energy holes in the network. After the relocation, all sensors remain active and static and the data sensing and transmission begin. With a good deployment, the energy consumption within the monitored area is balanced and the network lifetime is prolonged.

Mobile sensors can also be used as mobile relays. When a mobile relay moves to a new location, it inherits the sensing, transmission and receiving responsibilities of the co-located static sensors which can go to the sleep mode to save energy. With an appropriate sensor scheduling mechanism, the mobile nodes can prolong the lifetime of the bottleneck sensors and as a result the whole network lifetime is prolonged. In addition, mobile relays can be used as ferries. A ferry can carry other sensor data and forward the data to the sink when it gets within communication range. In this case the expensive multi-hop communication over long distances is reduced.

The rest of the paper is organized as follows. Section 2 discusses mechanisms using mobile sinks to prolong network lifetime. In Section 3, we overview mechanisms using mobile sensors relocation to improve the initial deployment. We continue in Section 4 with a presentation of some mechanisms using mobile relays. Section 5 shows a comparison among different mechanisms and Section 6 concludes the paper.

2. Mechanisms using mobile sinks

WSNs usually contain two types of nodes: sensor nodes and sink (or base station) nodes. A sensor node is a small device that has limited power, sensing and computation capabilities, while a sink node has more resources in terms of power, computation, and mobility. Sometimes sensor nodes are grouped in clusters using various mechanisms and one of the sensors is selected as cluster head based on various criteria. A cluster head manages the sensors in its cluster, gathers information from them, and forwards data to/from the sink. Section 2.1 presents algorithms where sinks are moving on predetermined paths and Section 2.2 discusses algorithms where sinks move autonomously.

2.1. Algorithms with pre-determined sink mobility path

Luo and Hubaux [18] developed a joint mobility and routing strategy (JMR) to increase the WSN lifetime of a periodic data collection application. The network consists of a base station and sensor nodes distributed using a Poisson process within a circle of radius R and assumes a relatively dense and strongly connected network. The authors provide a solution for maximizing the network lifetime by addressing a load balancing problem. They propose a min–max solution in terms of the average load of sensor nodes. The base station not only gathers data from sensors that are within communication range, but also uses a multi-hop routing for data collection when the base station stays.

First the authors show the improvement obtained by employing a mobile sink. JMR strategy is obtained by fixing the routing strategy to the shortest path routing and searching for the optimum mobility strategy, and then based on the optimum mobility strategy, it searches for a routing mechanism that performs better than the shortest path routing. Through theoretical analysis, the authors prove that the optimum symmetric strategy is a circular trajectory at the periphery of the network, see Fig. 1a. This trajectory minimizes the distance between the base station and the network center (the center of the circle), thus minimizing the load.

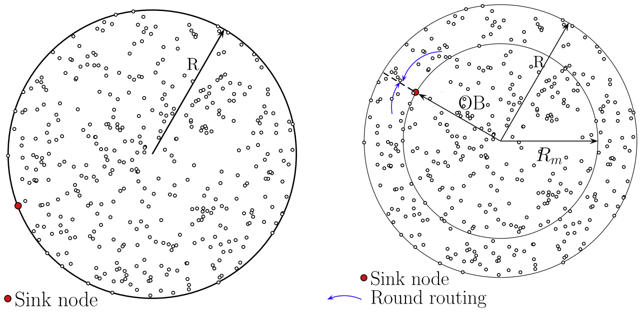
The authors propose a *better* routing strategy by exploiting the energy capacity of the nodes closer to the periphery of the circle to compensate the energy consumption of the hotspots (nodes closer to the center of the network). The sink moves on a circle with radius $R_m < R$ (where R is the original circle radius), see Fig. 1b, and the network is divided into two parts: the area of the disk with radius R_m and the area between the inner circle and the periphery of the network (named annulus). The routing strategy in the inner circle is shortest path routing, while the routing employed in the annulus is a round routing until it reaches OB, where OB is the radius from the center of the circle to the location of the base station, and short path to the base station. Round routing is a shortest path routing involving only nodes in the annulus. It follows the shortest path until it reaches OB using only nodes in the annulus, see Fig. 1b.

Simulations show that a mobile base station reduces the load by 75%, thus increasing the network lifetime by 400%. A JMR strategy reduces the maximum load in the network section within the mobility trajectory (e.g. network within the disk of radius R_m), with the tradeoff of an increased load in the network section outside the mobility trajectory (e.g. network within the annulus).

Saad et al. [21] present a solution to the problem of planning an arbitrary moving trajectory for a mobile sink in hierarchical structure sensor networks. The mobile sink starts at a fixed position and follows a well-planned moving path, which ends with the sink returning to the start position. Sensor nodes are randomly deployed in the area. Sensors are organized into clusters and cluster heads are selected. A cluster head has the role of gathering information from the nodes in its cluster, saving data in a buffer, and then communicating data to the mobile sink when it gets in the range.

The main idea of the adapted moving strategy (AMS) is to search within a space of possible configurations for a solution path, having the objective of balancing the tradeoff between energy efficiency, total path length, and buffer overflow deadlines. The moving strategy is divided into two tasks: (1) identification of path-points through which the sink must pass and (2) path optimization. The authors present a simple non-expensive path-point identification model as follows. The clustering mechanism forms clusters and then calculates the centroid of the clusters. It organizes all network cluster head nodes into distinct groups. The algorithm successively merges groups close to each other (maximum two hops apart) until the group is maximized. A group is maximized when no more cluster heads can be added to it. The path-points that form the sink path is simply set to the centroid points of each cluster.

In the second phase, the sink moves along the path through the centroid points of each cluster, which means that it will pass at



(a) JMR network organization (b) JMR round routing

Fig. 1. Joint mobility and routing strategy.

most one hop away from each cluster head, therefore enabling each cluster head to send data directly to the sink, thus saving energy. Path optimization is accomplished by employing the *Bees algorithm* with an appropriate fitness function that incorporates the optimality criterion.

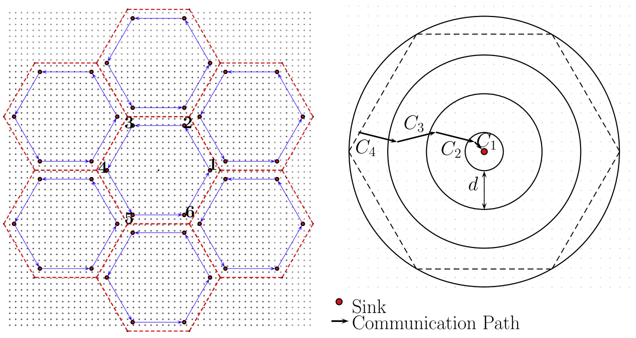
The Bees algorithm is an optimization algorithm inspired by the natural foraging behaviors of honeybees to find the optimal solution. The algorithm requires setting a number of parameters: n – number of scout bees, m visited sites – number of sites out of n, e – number of best sites out of m selected sites, nep – number of bees recruited for best e sites, nsp – number of bees recruited for the other selected $(m\text{-}e)$ sites. The algorithm starts by generating an initial population of n random solution, then it assigns a fitness function to each individual. The individual fitness function is stored in *Best*, the initial population.

The following steps are repeated until the stop criterion is met: select the elite bees and elite sites for neighborhood search, select other sites for neighborhood search, recruit bees around selected sites and evaluate their fitness function, select the fittest bee from each site, assign the remaining bees to search randomly, assign a fitness value to each individual and store the fittest individual B_i in the ith iteration. If $B_i < Best$ then repeat the steps in the loop, otherwise if $Best = B_i$, then the algorithm ends by returning *Best*. The set of path-points is represented as a vector, which can be viewed as the order in which the path-points are visited.

Simulation results prove that the proposed algorithm is 50% better than the one with a static sink and 30% better than the one where the sink moves on the periphery of the network [18].

Marta and Cardei [19] proposed a data-gathering algorithm employing sinks mobility with pre-established path. The heterogeneous WSN consists of a large number of sensor nodes with limited capabilities and multiple mobile sinks with unlimited capabilities. A periodic data-gathering application is considered where data is sensed and b data bits are transmitted by each sensor in each time period T, to the closest sink. Data is forwarded to the closest sink using multi-hop communication based on the collection trees formed by the clustering algorithm. Sensor nodes are uniformly and randomly distributed. The authors address the Sink Mobility for Network Lifetime Increase (SM-NLI) problem, with the objective of designing a sink movement plan such that the network lifetime is maximized and the sinks remain interconnected all the time.

The sensing area is divided into a hexagonal tiling, similar to Fig. 2a, with the sinks being located in the hexagon centers. The energy consumed by each sensor node is computed using a corona approach as illustrated in Fig. 2b, where the hexagon centered at each sink is divided into coronas of width d, where d is the sensor communication range. A message transmitted by a sensor in corona C_4 is forwarded once by a sensor in each corona $C_3,\ C_2,\ C_1$, until it reaches the sink. Thus the energy consumed by a sensor can be computed depending on the corona where it belongs. The sinks form a connected backbone all the time.



(a) Sinks' movement trajectories (b) Corona division for a sink

Fig. 2. Corona division and sink trajectories.

First, the case when the sinks move along the hexagonal perimeter is considered. Sinks movements are synchronized, therefore sinks' relative positions remain the same at all times. Since the sink backbone was initially connected, it remains connected at all times during the sink movement. First, the 6-position sinks movement case is addressed, where each sink moves along the perimeter of the hexagon, stopping in the corners of the hexagon. At each stop, the sink collects data over a period T and then moves to the new location. After stopping in the six corners, the movement cycle is repeated.

The nodes in the first corona (e.g. corona closest to the sink) consume the most energy since they have to forward data on behalf of other sensors that report to the same sink. The sink movement algorithm has the goal to prevent sensor nodes from belonging to first coronas of two different sinks. To achieve this objective, the sinks move along the perimeter of some inner hexagons of the original tilling hexagons. Fig. 2a shows the six intermediate positions on a sink trajectory. Simulations show that the 6-position sinks movement resulted in a 3.48 times improvement in network lifetime compared with the static sink case where the sinks are located in the center of the hexagons.

Beside the 6-position sinks movement, the authors investigate the case when a sink uses as many positions as possible in its movement along the hexagonal perimeter, with the requirement that a sensor does not belong to first coronas of two different sink locations. The number of stops of a sink along the hexagon perimeter depends on the simulation parameters. Simulations conducted on a 12-positions case show 4.86 and 1.39 times improvements in network lifetime compared to the static case and the 6-position case, respectively.

Somasundara et al. [23] propose an adaptive motion control (AMC) for the mobile node. The focus is on controlling the motion of the sink node in time, not in space. This refers to the speed profile of the mobile node along the chosen trajectory. A closed path is considered for the mobile sink node. The objective is to allow the sink to traverse the path within specified latency constraints and attempt to maximize the data delivered. The mobile node identifies congested regions of sensor nodes in order to maximize data delivery. Depending on the number of congested regions the sink will decide how much time it will spend close to each region and the traveling speed, such that to increase network lifetime.

Gandham et al. [11] use an integer linear program (ILP) to determine new locations of the base stations and propose a flow-based routing (FBR) protocol. ILP is formulated to minimize the maximum energy spent by a sensor node in a round. The solution of ILP indicates the base station's next location. Sensor nodes use a multi-hop routing protocol, e.g. Minimum Cost Forwarding (MCF). At the beginning of each round new base station positions are computed and they remain fixed during that round, while the base stations gather data from the network. The sensor network is represented as a graph $G(V,E)$, where $V = V_s \cup V_f, V_s$ – the set of sensor nodes and V_f – the set of feasible sites. $E \subseteq V \times V$ is the set of wireless links. In ILP, y_l is a 0–1 integer variable corresponding to each $l \in V_f$ such that:

$$y_l = \begin{cases} 1 & \text{if a base station is located at a feasible site } l \\ 0 & \text{otherwise.} \end{cases}$$

ILP is formulated to include equations regarding: (1) number of messages a node i transmits to node j, where j is a neighbor of i; (2) energy for transmitting and receiving a message; (3) the number of limited sites is at most K_{max} – the maximum number of base stations; (4) ensuring a message transmission to a feasible site only if a base station is at that site; (5) minimizing the maximum energy spent by any sensor node during a round. The ILP solver returns possible positions for base stations and the flow information is used by the sensor nodes to route messages in an energy efficient manner. One of the base stations will solve the ILP problem. Initially the sinks are deployed randomly and after the topology is constructed at one of the sinks, the ILP problem is solved. ILP solves two minimization problems, one denoted BSL^{mm} which minimizes the maximum energy consumption per sensor node and another denoted BSL^{me} which minimizes the total energy consumption during a round.

The steps performed by each sensor node in the FBR protocol are: (1) maintain a counter for each outgoing link; (2) use a round robin approach to select the outgoing links when a sensor node has data to transmit; (3) if the counter is greater than the number of packets, then all the packets to be transmitted are sent through a single link. The link counter is decremented by the number of packets transmitted. If the counter is smaller than the number of packets, then only a number of packets equal to the counter is transmitted on that link and another link is selected through the round robin algorithm to send the remaining packets; (4) if all the counters for outgoing links are zero then a link is randomly selected and all the packets are sent on that link. To perform the FBR mechanism as described above, each sensor node needs to know the flow informations sent by the base station to each sensor node. Simulations show that BSL^{mm} performs twice as good as BSL^{me} when three base stations are deployed.

2.2. Algorithms with autonomous sink movement

Bi et al. [4] propose an autonomous moving strategy (AMS) that applies to a WSN consisting of many sensor nodes and a mobile sink in charge of gathering sensed data periodically. Both sensor and sink nodes are aware of their location. First, sensor nodes discover their 1-hop neighbors by exchanging Hello messages. Next, the sink starts gathering sensed data periodically.

Each data-gathering period consists of three phases: (1) the sink sends a notification message to inform sensor nodes of its position, controlling the spreading range of the notification using a time-to-live (TTL) field; (2) sensor nodes report their data to the sink through a multi-hop path by employing a location-based routing algorithm, as both the sink location and the neighbors locations are known; (3) the sink decides its new location-based on the sensors energy levels gathered in the previous phase and arrives at the new location before the next gathering period starts. In the last phase, the sensor nodes can go to sleep since they do not participate in the sink's decision.

The sink autonomous moving strategy is called *half-quadrant-based moving strategy* (HUMS) [4] and is incorporated with the data-gathering protocol. The sink, called *energy mower*, moves proactively towards the node that has the most residual energy to balance energy consumption among all sensors in the network. Through the data-gathering process, the sink collects information about the residual energy and the location of the nodes with the highest and lowest energy level, respectively. In each data-gathering period, the energy mower will reselect the moving destination (*MoveDest*) according to the new energy distribution collected in the current round. The movement is constant and in each round the sink moves only one hop (one communication range distance) closer to *MoveDest*. If a new high is encountered in the network before the energy mower reaches *MoveDest*, it alters its path to reach the new location of the sensor node with the highest energy.

When changing *MoveDest*, the difference in energy must be at least one unit, otherwise the energy mower maintains its path. The energy mower in HUMS makes a moving decision based on a coordinate system that takes its current location as the origin and divides the coordinate system into eight half-quadrants. A *quasi-hotspot* is defined as the set of sensor nodes that are reported to have the lowest energy in the network for one period. Depending on the location of these quasi-hotspots in the eight half-quadrants and the location of the destination, the energy mower will decide the trajectory to follow.

There are two cases to consider depending on whether the energy mower is far from or close to the *MoveDest*, respectively. When the energy mower is far from the *MoveDest*, the goal of the half-quadrant-based algorithm is to make the sink avoid sectors that contain quasi-hotspots, if possible. When there are no clean neighboring sectors (left and right of the destination sector), the algorithm follows a *minimum-influence position selection* algorithm (MIPS). In MIPS, the energy mower takes into account the position distribution of some near quasi-hotspots when determining a sojourn position in the sector selected by the half-quadrant-based algorithm.

In the second case, when the sink is close to the destination, a *square hopping mechanism* is used by the energy mower to determine the preferred sojourn position near the destination. This mechanism requires the selection of four points around the destination on a circle whose radius is smaller than the communication range. The goal here is to move the sink about one hop away from the destination, to force the destination (sensor node with highest residual energy) to forward messages on behalf of other nodes in the network. The authors consider the case when a destination has few neighboring nodes, thus it will forward data on behalf of few nodes, keeping the destination's energy level on high. Because of this, a *blacklisted-based* mechanism is introduced to prevent the energy mower from considering these dangerous nodes. A dangerous node is the one which has a small number of neighbors and which keeps the destination energy level on high while draining the energy of other nodes. These nodes are undesirable because they can partition the network while the change condition (the destination energy level is above the threshold so the sink does not need to move) is still true.

In the blacklist mechanism, the energy mower maintains a blacklist with dangerous nodes in the network. The energy mower uses two thresholds to determine if a node is dangerous or not: TH_p represents the threshold for the number of data-gathering periods in which the energy mower selects the same node as *MoveDest*, and TH_D represents the threshold for the number of total data messages received from the same *MoveDest*. When TH_p is exceeded and TH_D is below expectations with regard to the same node (*MoveDest*), the energy mower adds the current node *MoveDest* to the blacklist and temporarily prevents it from being selected as *MoveDest* again. A record is deleted from the blacklist after a predetermined interval. Another type of nodes added to the blacklist are the nodes with the highest residual energy in the network, located within the communication range of a quasi-hotspot.

The performance of HUMS is compared with three data-gathering protocols: static sink located in the center of the network area, random moving sink, and sink moving on the network periphery [18]. Simulation results show that HUMS algorithm outperforms the others in terms of average network lifetime when the initial energy of sensor nodes is varied. The average network lifetime for different node densities is studied. HUMS outperforms the other algorithms when the number of nodes is among 150 and 250, otherwise Peripheral [18] performs better. In the irregular-shaped networks with randomly distributed sensor nodes, HUMS outperforms the other algorithms.

Wang et al. [24] propose a solution for increasing WSN lifetime by employing adaptive location updates for the mobile sink. Each node and the sink know their own location and the location of their one hop neighbors. The proposed protocol, Adaptive Local Update-based Routing Protocol (ALURP), is an improvement of a previous work by the authors, LURP. After network deployment, the sink broadcasts its location information to the entire network. The routing process is divided into two stages: (1) data packets are forwarded from sensor nodes to a destination area, denoted DN_A, and (2) data packets are forwarded to the sink in the destination area. When a sensor node has a data packet to send to the sink, it will first forward it to a node in the small area centered to a *virtual center* (VC), according to some geocasting protocol. When the data packet reaches DN_A, the second routing stage begins. The packet is forwarded to the sink based on a topology-based routing protocol, instead of being flooded within the destination area. Initially, VC has the location of the sink and the destination area is a disk with the center in VC and fixed radius.

If the sink moves within the destination area, then it has to broadcast its location only within the destination area. Otherwise, the new sink location information has to be flooded in the whole network and a new destination area has to be built. When the sink moves outside of the destination area, there may be nodes far away from the sink which will not get the location update in time. In this case, the sensor node will send the data packet toward the known destination area and once arrived there, the sensor nodes will know how to forward the packet toward the new destination area. The size of the destination area is an important parameter in the protocol with impact on the network lifetime.

The destination area can be a disk with center in VC and radius equal to the distance between VC and the sink. Initially VC has the location of the sink and the destination area is the disk with a fixed radius centered in VC. As the sink moves, the destination area is modified, having as a new radius the distance between VC and the new position of the sink. This is an *adaptive area* since the radius changes as the sink moves. The new sink location information is broadcasted only in this adaptive area. A problem arises when the sink moves closer to the VC, thus reducing the adaptive area. The sensor nodes that were previously in the adaptive area will hold an obsolete sink location information. The solution to this problem is to have the sink inform the nodes in the former adaptive area, but not those in the current adaptive area, to flush the location information of the sink.

In simulations, ALURP is compared with LURP and with Flooding-based Location Update Protocol (FLUP) [31]. The sink chooses a random neighbor and moves toward it. After the sink reaches the location, it chooses another random neighbor and moves toward it, and so forth. The simulation results show that ALURP performs better than the other two protocols in terms of energy consumption when the node density is varied. ALURP performs 10 times better than LURP when the simulation is run for 75000 nodes. LURP performs better when the velocity of the mobile sink is increased. The optimal radius for the adaptive area is 105 m for a network of 12,000 nodes deployed in an area of $2000 \times 2000\ \mathrm{m}^2$. ALURP reduces the delay and energy consumption and is suitable for large-scale and delay-sensitive WSNs.

Marta and Cardei [19] address the *SM-NLI* problem for the case when the sinks move autonomously such that (1) the sinks remain interconnected all the time forming a *virtual sink backbone*, and (2) network lifetime is maximized. The network model has been introduced in Section 2.1 when the paper is first discussed. The data-gathering mechanism is organized in rounds of time T.

At the beginning of each round, data collection trees are established using a clustering mechanism favoring the closest sink. Each sink serves as a cluster head and it broadcasts a *CLUSTER_INIT*(*ID,hops=0*) message containing the sink id and the number of hops which is initially zero. Each sensor node maintains information about the closest sink and forwards only messages from which it learns about a closer sink. Once the clusters have been constructed, sensor data are collected along the paths formed by the next hop field. At the end of each round, a sink decides whether or not it moves to a new location, depending on the energy levels of its 1-hop sensor neighbors. These are the sensor nodes that will deplete their energy first since they also have to forward messages on behalf of other sensors.

Sinks move in zones with higher energy resources, but they remain interconnected at all times. The sensors 1-hop away from the sink send their current energy levels to the sink at the end of each reporting interval. This information can be piggybacked to data messages. If at least $p\%$ of the sensors have reached a low energy threshold E_{th}, then the sink searches for a new zone where sensors have richer energy resources. The zone where the sink moves must have energy at least E'_{th}, where $E'_{th} = E_{th} + \alpha E_{th}$ and $0 < \alpha < 1$.

Sink S_i uses an incremental ring approach in its search for a new location, by sending *LOCATION-REQ* messages. The first candidate sink locations are the sensor locations in its cluster. When a candidate position is found, the sink must ensure that the sink backbone connectivity is maintained. The sink considers the candidate locations closer to its current position first. If no candidate location is valid due to the connectivity requirement, then S_i increments the number of hops in order to increase the search neighborhood. If no candidate location has been found after the whole cluster has been searched, then the sink does not move to a new location.

When a sink has a candidate moving location, it checks the sink backbone connectivity as follows [19]. Each sink has two transceivers, one for communication with sensor nodes and the other for communication with other sinks. The sinks exchange HELLO messages to determine their l-hop neighborhood. Then using the l-hop neighborhood and the candidate sink location, Breadth-First-Search algorithm [7] is run to check for the connectivity.

In addition, coverage preserving and timely delivery of data aspects are considered in [19]. The authors address the *Coverage-based SM-NLI* problem, where sensors alternate between sleep and active states and the sinks move autonomously such that: (1) the set of active sensors provides area coverage (e.g. each point in the deployment area is covered by at least one sensor), (2) the sinks remain interconnected all the time, and (3) network lifetime is maximized.

Data gathering mechanism is organized in rounds. At the beginning of the first round, a scheduling mechanism is run to decide which nodes stay active or go to sleep, respectively. Then data collection trees rooted at the sinks are formed (using only the active sensor nodes) and data collection phase begins. At the beginning of each round, each sink decides whether it has to move based on the energy level of its 1-hop sensor neighbors. If a sink satisfies the moving condition, then before moving, it triggers an update of the set of active sensors in its cluster such that to maintain coverage and connectivity. The sensors with higher residual energy have higher priority to become active. Then the moving condition is re-evaluated. If the sink still has to move, then it moves to a new location as discussed earlier, otherwise no movement is performed. Two sensor scheduling mechanisms are investigated [19]: using connected dominating set [28] and a grid approach where any sensor in a grid can provide both connectivity and grid coverage.

The second extension [19] has to satisfy an additional requirement of a time-constrained data delivery. The distributed SM-NLI solution is changed as follows. The constraint on the timely delivery is implemented by limiting the maximum number of hops in any data delivery path. When sink S_i decides to move, there is need to ensure that the resulting collection tree paths have length at most t, assuming that at the beginning the sinks were deployed uniformly in a grid and that the maximum number of hops was at most t. The sink S_i can compute the maximum path length in its tree, denoted t_i, piggybacked with data collection messages. If S_i has to move, then its new location has to be at most $t - t_i$ hops away from the current location. This ensures that each sensor in the cluster is at most t hops away from the new sink location. If a sink cannot find a suitable new location, then it does not move.

The simulations [19] consider two sensor deployment cases: a random uniform distribution and a bivariate Gaussian distribution. Initially, the sinks are uniformly distributed in a hexagonal tiling. At the network start-up, each sensor has the same initial energy. Four algorithms were compared: static sinks, 6-position movement sink, 12-position movement sink (described in 2.1), and autonomous movement sink. For uniform sensor distribution, 6-position and 12-position sink movement have longer lifetime than the autonomous sink movement because sinks' movement is synchronized, thus data collection trees are more balanced. For bivariate Gaussian distribution, the autonomous sink movement algorithm has the best performance, followed by the predetermined-path sink movement cases. The coverage based mechanism improves network lifetime by scheduling sensor nodes to sleep. The grid based mechanism outperformed the connected dominating set based approach. When time-delivery is imposed, network lifetime decreases since sinks movements are reduced.

Bi et al. [3] propose two autonomous moving schemes for mobile sinks: one-step moving scheme (OSMS) and multi-step moving scheme (MSMS). The data-gathering mechanism is similar to the one in [4], described earlier in this section. OSMS is a simplified version of the HUMS algorithm [4], with no half-quadrant-based strategy employed. The sink moves in the communication range of the moving destination in one-step, thus avoiding passing by the middle nodes with low residual energy on the moving path. In MSMS, the sink decides to move based on the information of 2-hop neighboring nodes. Compared to [4], data packages will be smaller as the energy and the location information for the sensor nodes with highest and lowest energy are not needed anymore. Only 2-hop neighbors will include energy levels and location information. The sink decides to move closer to the sensor node with the highest energy in its 2-hops neighborhood and closer to the moving destination. Simulation results show that OSMS performs better than MSMS since OSMS affects the middle nodes less than MSMS, which moves one hop at a time in the direction of the destination area and not in one-step as OSMS.

3. Mechanisms using mobile sensors redeployment

Recent research has focused on methods to improve the initial deployment. One possible method is to use mobile sensors, thus allowing sensors to relocate [5,30]. In a WSN, if sensors in the network are uniformly deployed, sensors closer to the sink tend to consume more energy than those located farther away from the sink [17,20]. Papers [5,30] propose to prolong network lifetime by adjusting the sensor density according to the distance to the sink, thus reducing the uneven energy consumption.

Yang et al. [5,30] study a general architecture where sensors send data to a sink located centrally in a circular monitored area. The monitored area is virtually divided in coronas, as illustrated in Fig. 3a. A message originating in corona C_i is forwarded by sensor nodes in coronas $C_{i-1},\ C_{i-2}$, and so on until it reaches corona C_1 from where it is transmitted to the sink. Assume that the energy consumption is proportional to the number of messages transmitted. Intuitively, to balance energy consumption, we will deploy the fewest sensors in the last corona C_n and the largest number of sensors closest to the sink, which is corona C_1, see Fig. 3b. Let ρ_i denote the sensor density in the corona C_i, thus $\rho_1 \geqslant \rho_2 \geqslant \cdots \geqslant \rho_n$. The non-uniform sensor density is computed for each corona [5,30] such that all sensors deplete their energy at the same rate, resulting in a balanced energy consumption.

Yang and Cardei [30] consider a flip-based mobility model, where sensors moving distance and the number of movements are limited. A sensor can move from its current location to a new location when triggered by an appropriate signal. Such movements can be determined by propellers powered by fuels or springs. A centralized mechanism is proposed to reposition sensors after the initial deployment, according to the desired density, for the purpose of maximizing sensor network lifetime while minimizing the total movement of sensors. The monitored area is divided into a grid of regions, where each region is a $R \times R$ square as represented in Fig. 3c. In this case, the division in coronas is not circular, but it follows the regions' contour. When the region granularity is very small, $R \rightarrow 0$, the division in coronas is similar to the one in Fig. 3a, where coronas are circular.

The mechanism [30] has two steps. First, a maximum flow minimum cost graph for the initial deployment is constructed, then the Edmonds–Karp's algorithm [7,10] is used to find out the movement plan. This movement plan shows the way in which sensors will move (flip) to other regions for energy balancing. Assume that after the initial deployment there are $s_1,\ s_2,\ s_3, \ldots,\ s_m$ sensors in regions $R_1, R_2, R_3, \ldots, R_m$, and that the target number of sensors for each region is $t_1,\ t_2,\ t_3, \ldots,\ t_m$.

To form the multi-flow graph, three vertices are added for each region R_i:

- B_i: Base vertex of the region, which keeps track of the number of sensors in the region R_i.
- In_i: Vertex that keeps track of the number of sensors that moves from other regions.
- Out_i: Vertex that keeps track of the number of sensors that can move to other regions.

Edges are added to the graph. Each edge has a capacity representing the maximum number of sensors that can be transmitted along this edge. Edges inside each region R_i are added as follows:

- Add an edge (In_i, Out_i) with capacity s_i.
- If $s_i \geqslant t_i$, then add an edge (B_i, In_i) with capacity $s_i - t_i$. If $s_i < t_i$, then add an edge (In_i, B_i) with capacity $t_i - s_i$.

In addition, edges are added between two regions reachable from each other. A region can reach k regions on each direction: right, left, top, and bottom, where k is a given parameter. If two regions R_i and R_j are reachable from each other, then two additional edges are added to the graph as follows:

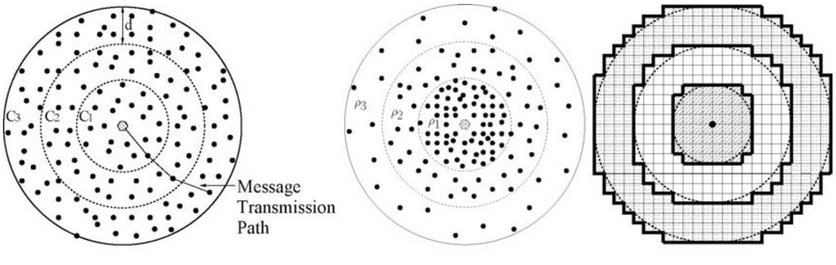
- Add an edge (Out_i, In_j) with capacity ∞.
- Add an edge (Out_j, In_i) with capacity ∞.

The cost values are defined as follows. For all edges inside regions, the cost is zero, since no sensor movement is involved. Edges between regions have a cost of one, since a flow on this edge represents a sensor's flipping from one region to another.

Simulation results show that the network lifetime is effectively prolonged after redeployment. The larger the maximum distance that sensors can flip is, the closer the network lifetime gets to the ideal case, which improves the network lifetime at least n times compared to the initial uniform deployment, where n is the number of coronas.

Cardei et al. [5] consider a more flexible mobility model for the sensors, the number of sensor movements is not limited, and the goal is to relocate sensors to achieve the required densities such that the network lifetime is maximized, while minimizing the total sensor moving distance. Three solutions are proposed: an Integer Programming (IP) approach, a localized matching method, and a scan-based mechanism.

In the IP approach, the objective function asks to minimize the total sensor moving distance. A region in corona C_i can be a source,



(a) (b) (c)

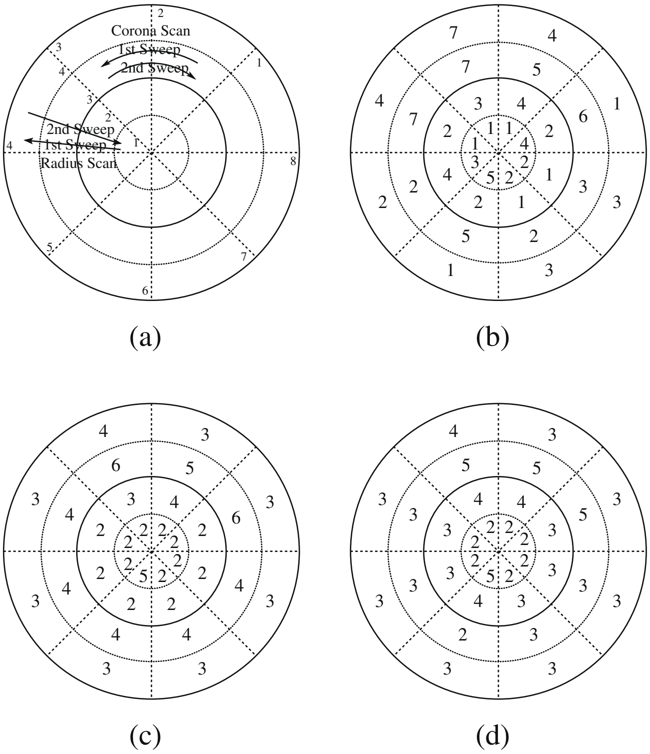
Fig. 3. WSN model using coronas concentric to the sink: (a) uniform distribution, (b) non-uniform distribution and (c) division in coronas for small grids.

hole, or neutral region if the current number of sensors is greater than, less than, or equal to the desire number of sensors. The constraints of the IP require that each hole region receive sufficient sensors from source regions such that to become neutral regions.

The second mechanism [5] is a localized matching method. This is a three-way mechanism. First, a hole region broadcasts the number of sensors it needs to become neutral, within a localized neighborhood. Second, source regions respond with the number of sensors they can contribute. Lastly, the hole region plans which sensors to take, taking into account the distance the sensors have to travel. The decision is broadcasted to the source regions and then the actual sensor movement takes place.

The third mechanism is a scan-based approach. The network is virtually partitioned into thinner coronas (or rings) and sectors, as shown in Fig. 4a. Two scans are used in sequence: corona scans followed by radius scans. A corona scan will balance the number of sensors per corona and at the end of this scan, regions in the same corona will have approximately the same number of sensors, see Fig. 4c. In the radius scan, sensors are redistributed in a sector according to the desired sensor densities, see Fig. 4d. Each scan has two sweeps. The first sweep scans the regions in a corona (sector) from the first region 1 to the last region, and the second sweep does the scanning in the reverse direction. During the first sweep, the total number of sensors per corona (sector) is computed. Then knowing the desired sensor density distribution, the final number of sensors in each region and sensor movement is computed during the second sweep.

Simulation results show that all three mechanisms effectively prolong the network lifetime compared with the initial uniform deployment. IP and localized matching methods have similar network lifetime improvements and they are better than the distribution achieved by the scan-based approach. For the total moving distance, the localized matching method gets close results to that of the IP approach, which is the optimal solution. The scan-based approach has the largest total moving distance. Simulation results also show that the localized approach has larger overhead compared to the scan-based approach in terms of number of control messages exchanged.



(a) (b)

(c) (d)

Fig. 4. Example for the scan-based approach: (a) area partitioning, (b) initial deployment, (c) deployment after the corona scan and (d) deployment after the radius scan.

4. Mechanisms using mobile relays

There can be different ways to use mobile relays to improve network lifetime. Wang et al. [25] consider that mobile nodes can inherit the sensing and relaying (transmission and receiving) responsibilities of the bottleneck nodes. Consider for example the case when a mobile node moves to be co-located with a bottleneck node and performs the transmission and receiving tasks on behalf of the bottleneck node, then the bottleneck node can go to sleep to save energy. As a result the lifetime of the bottleneck node is prolonged and the whole network lifetime is improved.

Using mobile nodes as ferries is another way to prolong network lifetime [13,14,22,27]. Static sensors could send data to the mobile nodes which buffer and then drop off the data at the sink. This store-carry-forward mechanism is widely used as an energy efficient way in delay tolerant networks. Banerjee et al. [2] consider the case when sensors form clusters, the sensors and the sink are static, and the cluster heads are mobile and work as relay nodes.

Wang et al. [25] propose mechanisms that use mobile relays to prolong network lifetime. Fig. 5 shows an example. Assume that the whole network is composed of two components, component 1 and component 2. The two components are connected via sensors A and B, which are the black nodes in the figure. These two sensors are bottleneck nodes because they need to forward all the traffic between the two components and their lifetime is *T*. Other sensors in the network have lifetime much larger than *T*. If we use a mobile relay node (the circle node in the figure), which has the same transmission range as sensors A and B, the network lifetime can be prolonged. One way is that the mobile node shuttles between sensors A and B and inherits the responsibility of

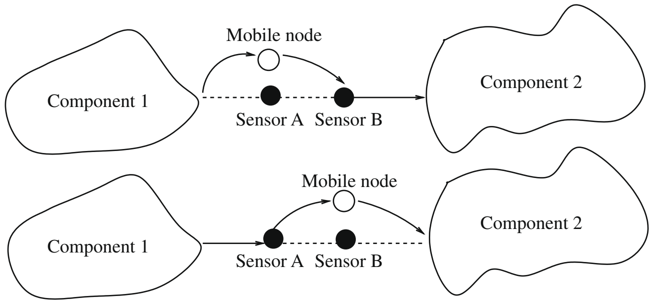


Fig. 5. An example of using mobile relay nodes to prolong the network lifetime.

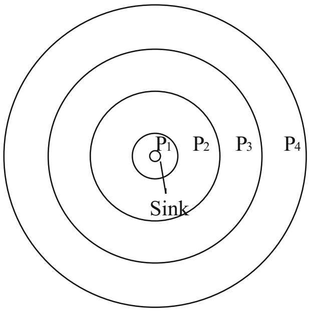


Fig. 6. Division the network into annuluses.

the sensor with which it is co-located. The solid lines between two components can be the alternative transmission paths besides the original path between the component 1 and component 2. With an appropriate shuttling scheduling, the network lifetime can be increased to 2*T*.

Static sensors are assumed to be densely deployed in the monitored area with one static sink located at the center. Besides low-cost energy-restricted sensor nodes, there are also energy rich mobile nodes deployed in the network. Sensors are divided into different sets according to their distance to the sink and the network is virtually divided into annuluses as shown in Fig. 6. The annulus P_i contains all the sensors which can reach the sink in i hops. For example, P_1 contains all sensors that can directly communicate with the sink, P_2 contains sensors which are 2-hops away from the sink, and so on. Q_j denotes the set of sensors that can reach the sink within j hops, $Q_j = \cup_{k \leqslant j} P_k$. The set $\overline{Q_j}$ contains all sensors which are outside Q_j.

Through mathematical analysis [25], it is shown that when there is only one mobile relay, it has to stay within a two hop radius to maximize the network lifetime. According to this observation, authors design the following mobility strategy: starting from the sink, the mobile relay traverses a path which forms a set of concentric circles, centered around the sink with increasing radii, until it reaches the periphery of Q_2.

Next, a routing mechanism is proposed using the above mobility strategy. When the mobile relay is located at position *M* and the sink is located at position *O*, all the traffic in $\overline{Q_2}$ is first aggregated to points on the line *OM* and then forwarded hop-by-hop along the line *OM* to the sink. In the case with m $(m > 1)$ mobile relays in the network, they will stay within Q_{2m}, resulting in nearly $4m$ times increase in network lifetime when the radius of the network is large enough.

Simulations show that the network lifetime improvement ratio increases as the network size and density increase. With moderate size and density, it improves the network lifetime by 130% with only one mobile relay. As the network becomes larger and denser, the lifetime improvement will approach the bound of 300%.

Shah et al. [22] propose a three tier architecture for sparse sensor networks. The key of this architecture are the mobile MULE nodes, which can be served by vehicles (cars, buses), animals, or people equipped with transceivers. The mobility of the MULE nodes is modeled as simple symmetric random walk and their movements cannot be predicted in advance. They are assumed to have short range wireless communication capabilities and they can exchange messages with the nearby sensors or access points encountered when they move. Mobile MULE nodes pick up data from sensors when the sensors are within transmission range, buffer and drop off the data to the access points in close proximity.

The top tier of the three tier architecture consists of network connected devices, such as access points, which can be set up at convenient locations where network connectivity and power are present. The second tier (or intermediate tier) is composed of mobile MULE nodes, which store, carry, and forward data between the top tier and the bottom tier. The MULE nodes are assumed to have large storage capacities (compared with sensors), renewable power, and the ability to communicate with the sensors and network access points. Using their motion capabilities, they collect and store data from the sensors, as well as deliver ACKs back to the sensors if necessary. Besides, MULE nodes can communicate with each other to improve system performance. For example, a multi-hop MULE network can be formed to reduce the latency between MULE nodes and access points.

The bottom tier of the network consists of randomly distributed wireless sensors. They perform sensing tasks and directly send data to mobile MULE nodes when they are close enough. The work performed by sensor nodes should be minimized since they have the most constrained resources.

This store-carry-forward mechanism is energy efficient because sensor nodes communicate over a short range. The energy consumed in transiting a message is related to the transmission range [12]. A longer transmission range will consume more energy compared with shorter transmission range. Sensor nodes use mobile MULE nodes to carry data to the sink. Sensors could transmit data to the MULEs when they come close, using therefore a short transmission range. Besides, short hop-by-hop data delivery paths could be used to transmit data from the sensors to the MULEs, involving therefore a small number of sensors in data forwarding. Using this mechanism, sensors save energy in transmission and the network lifetime is prolonged. However, the disadvantage of the mechanism is that it increases the delivery latency since sensors have to wait for a MULE node to come close enough before the transmission occurs. Such a mechanism is more suitable for delay tolerant applications where energy is a critical issue and the delay requirement is relatively loose.

Simulations [22] focus on the buffer capacity and data success rate. Simulation results show that the sensor buffer requirements are inversely proportional to the appropriate number of MULEs. The MULE buffer requirements are inversely proportional to both the number of mules and access points. When the sensor buffer is large, the buffer capacity on each MULE can be traded-off with the number of MULEs to maintain the same data success rate.

Wu and Yang [27] propose another store-carry-forward mechanism, which uses controlled mobility to improve the delay, the number of relays, and the moving distance of mobile nodes. Mobile nodes are assumed to be resource rich, such as vehicles. The network is divided into grids. Static sensors and mobile nodes deployed in a grid form a cluster and a cluster head is selected to deal with inter-grid communication. Network is assumed to be densely deployed so that each cluster has at least one static sensor and one mobile sensor.

Each cluster has contacts through short links, which are regular wireless communication links or through long links, which are performed through the motion of mobile nodes. It has four short links, which connect its direct neighbors on top, right, left, and bottom clusters. In addition, it has q long links. A sensor u that is within $m \times m$ space may have contacts through long links within $5\ \text{m} \times 5\ \text{m}$ space, i.e. a cluster v. Mobile nodes move along the long links between u and v, pick up data, and deliver data. When a sensor in a grid wants to send a data message, it chooses a contact through a short link or long link, that is as close to the destination as possible. The contacts outside $5\ \text{m} \times 5\ \text{m}$ space are not taken into account, since the moving distance would be too large for using relays.

Simulation results [27] show that the proposed algorithm generates a smaller number of message relays than the case without using mobile nodes while still maintaining a moderate moving distance and delay, which makes the algorithm suitable for wireless sensor networks. A larger network makes the performance of the algorithm more significant.

Banerjee et al. [2] consider a cluster-based sensor network and study the mobility of cluster heads to increase the network lifetime. Sensors and the sink are static. Sensors are uniformly and randomly deployed in the monitored area. In addition, there are mobile nodes deployed as cluster heads. Static sensors form clusters and select the mobile nodes as their cluster heads. Each static sensor joins one cluster and reports sensed data to the closest cluster head. After the clusters are formed, cluster heads start moving within their own clusters and cluster members remain the same all the time. Cluster heads collaborate with each other so that each one of them covers approximately the same fraction of the monitored area and cluster areas do not largely overlap. Each cluster head can directly communicate with the static sink or maintains a connected path to the sink all the time while it is moving. The sensed data is transmitted from the source sensor to the closest cluster head using single hop or multi-hop paths. The cluster head then forwards the data to the sink. The designed mechanism can be applied to both proactive report and reactive event-driven report. In both cases, the data can be transited to the sink without a significant delay.

Three mobility strategies [2] for mobile cluster heads are introduced: energy efficient mobility, event-oriented mobility, and hybrid mobility. In energy efficient mobility strategy, cluster heads always move towards the points where the residual energy is concentrated. The cluster head can use the prediction-based energy map approach to obtain the residual energy of sensors in its cluster. The residual energy could be computed in a predictable manner according to the past history of residual energy consumption, such as the transitional probability of a sensor's state change, and the energy dissipation rate.

In event-oriented mobility strategy, cluster heads always move towards the event, for example, the location where the traffic data flow of the event is concentrated. A simple implementation is to have mobile cluster heads move towards the direction of the sensors that are sending data to them. This strategy improves the sensor energy consumption around the event since the transmission distance between the source and the cluster heads is decreased.

In the hybrid mobility strategy, both residual energy and event location are considered. Using pure energy efficient mobility, cluster heads may be too far away from the event source. Using pure event-oriented mobility, cluster heads always stay around the event and sensors around the event run out of energy quickly and cause uneven energy consumption. Hybrid mobility considers a weight for these two strategies and combines them. When an event occurs, mobile cluster heads first move towards the event, and then move according to the energy efficient mobility strategy. This strategy ensures that data relays are always the energy rich sensors that are in the vicinity of the event.

Simulation results show the residual energy improvement using the three proposed strategies. Energy efficient mobility with uniform event generation increases the residual energy by 5% more than the event-oriented strategy. However, over a long period of time, if the events follow a non-uniform distribution, then the event-oriented strategy increases the residual energy by 8% compared to the energy efficient strategy. Irrespective of the distribution of the event, the hybrid strategy shows the best performance by increasing the residual energy by 23% more than the former two strategies. The improvement in residual energy with the hybrid strategy is only 14% less than the ideal case, which shows the effectiveness of the proposed strategy.

5. Algorithms comparison

All the mechanisms surveyed in this paper have a common objective which is to improve the network lifetime. All articles assume that sensor nodes have limited energy resources, while sinks or relay nodes have better capabilities. Using mobile nodes in the network is an important property that can be explored to prolong network lifetime.

The algorithms discussed in Section 2 are compared in Table 1. The main objective of these algorithms is to design mechanisms that prolong network lifetime by employing mobile sinks to gather

Table 1
Mobile sinks, algorithm comparison.

Scheme	No. mobile sinks	Reactive vs. proactive	Sink movement pattern	Algorithm char.	Data aggregation	Application type	Known sensor location	Network structure	Sink speed
AMS [21]	One	Proactive	Planned	Centralized	No	Store and send	Yes	Clusters	Constant
JMR [18]	One	Proactive	Planned	Distributed	No	Periodic data reporting	Yes	Flat	Constant
HUMS [4]	One	Reactive	Autonomous	Localized	No	Periodic data reporting	Yes	Flat	Constant
ALURP [24]	One	Proactive	Random	Distributed	No	Event driven	Yes	Hierarchical	Constant
Planned trajectory [19]	Multiple	Proactive	Planned	Distributed	Yes	Periodic data reporting	No	Hierarchical	Constant
Autonomous trajectory [19]	Multiple	Reactive	Autonomous	Localized	Yes	Periodic data reporting	Yes	Hierarchical	Constant
AMC [23]	One	Proactive	Planned	Distributed	No	Store and send	No	Clusters	Adaptive
FBR [11]	Multiple	Proactive	Planned	Centralized	No	Periodic data reporting	No	Flat	Constant
MSMS [3]	One	Proactive	Autonomous	Localized	Yes	Periodic data reporting	Yes	Flat	Constant
OSMS [3]	One	Proactive	Autonomous	Centralized	Yes	Periodic data reporting	Yes	Flat	Adaptive

information from the sensors. These algorithms are compared based on the following criteria:

- *No. mobile sinks*: Represents the number of sinks deployed in the network to gather information from the sensor nodes. Most of the algorithms surveyed use a single mobile sink in analysis and simulations. Some others [19,11] present simulation results for different number of sinks. Increasing the number of sinks up to a point has the effect of improving network lifetime, beyond that point the network lifetime is constant because each sensor node becomes at most 1-hop away from a sink.
- *Proactive vs. reactive*: Represents a characteristic of the routing protocol. Most algorithms surveyed are proactive. A proactive protocol finds a solution (e.g. for saving energy) before the network is partitioned. On the other hand, a reactive protocol will try to find a solution only after some nodes deplete their energy. Using a reactive routing protocol could have a down-fall on the network lifetime as it may cause network partition before having time to respond.
- *Sink movement pattern*: Can be planned, random, or autonomous. In the planned movement schemes, the sink trajectory is pre-established. For 6-position and 12-position algorithms [19] the sink movement follows the perimeter of a hexagon and each sink collects data from its cluster. Autonomous movement implies that a sink determines its new location autonomously, based on the current network conditions. For example [19] a sink could move to a new location-based on the current energy level of the sensors in its cluster. Another possibility is random sink movement, where a sink does not care about energy or other networking information in deciding its next location. For example, in ALURP [24] the sink moves to a random neighbor in each round.
- *Algorithm char*: Indicates the type of algorithm used. A localized protocol, e.g. [4,19], uses k-hop neighborhood information to make its decision, where k takes usually small values, such as 2 or 3. The local neighborhood information is usually obtained by exchanging Hello messages. On the other hand, in centralized algorithms a node needs global information to execute the algorithm. For example [21], the sink could get information about the whole topology and make moving decisions. Localized algorithms are scalable and perform well in large networks such as WSNs.
- *Data aggregation*: Can further improve network lifetime as it reduces the number of messages/packets transmitted in the network. Networks that implement a clustering organization usually employ data aggregation in cluster heads. For example [19], sensor nodes could form data collection trees and each parent node in the tree waits for the children's packets and then sends aggregated information (a fraction of the total number of packets) to its parent.
- *Application type*: The two main types of applications in WSNs are periodic data reporting and event-driven applications. Periodic data reporting applications involve sensing the environment and periodically reporting the data sensed to the sink. Event driven applications monitor the environment and when certain events occur, sensor nodes report this event to the sink. Another type of data reporting is store and send. In one such case [23] sensor nodes are organized in clusters and cluster heads have the role of storing the cluster information until the sink gets close enough.
- *Known sensor location*: Specifies whether the algorithm requires sensors to know their location. A sensor node can determine its location using GPS or by running a location computation mechanism [2]. Location information might be relevant for a sink in deciding its next location or can be used in cluster formation.
- *Network structure*: Indicates the organization of the sensor nodes in WSN. In a flat network, all sensor nodes have the same role of sending the sensed information to the sink. Clustering is used to divide nodes in clusters, usually based on their proximity. Each cluster has a cluster head which deals with the communication with the sink and which gathers and saves the sensing data from the sensor nodes in its cluster. Another approach [19] uses a hierarchical architecture with sensor nodes in one layer and the interconnected sinks in the second layer. Data collection trees rooted at the sinks are then formed. Using different roles for the sensor nodes provides another strategy. For example [24], the nodes in the circular destination area near the sink have the role of routing messages to the sink and they are the only nodes that know the location of the sink. The rest of the sensor nodes route their packets toward the destination area without knowing precisely the location of the sink.
- *Sink speed*: The sink speed is an important parameter in mobile WSNs. Most of the protocols surveyed use a constant speed for moving the sink from one place to another. Some mechanisms [19] consider that the sink moves to a new location and stops there for data gathering, then after a period of time moves to a new location, and so on. The sink could also use an adaptive speed [23] depending on the number of congested areas that it has to pass in order to gather the sensed information.

In Table 2 we compare the mechanisms from Sections 3 and 4 according to the following criteria:

- *Network structure*: WSN organization can be flat or hierarchical. In a flat architecture, every sensor has the same role and responsibility. In a hierarchical structure, sensors have different tasks. For example, in a cluster-based structure some sensors are selected to be the cluster heads. Besides the sensing tasks, they may manage the intra-cluster and/or inter-cluster communication.

Table 2
Mobile sensors/relays, algorithm comparison.

Scheme	Network structure	Network density	Delay sensitive	Sensors or relays mobility pattern	Known sensor location	No. mobile nodes	Algorithm char.	Time sync.	Minimizing moving distance	Routing proposed	Connectivity required
[2]	Hierarchical	NA	Yes	Controlled	Yes	Multiple	Distributed localized	NA	No	No	Yes
[5]	Hierarchical	High	NA	Controlled	Yes	Multiple	Centralized distributed localized	Yes	Yes	No	Yes
[22]	Flat	Low	No	Random	No	Multiple	NA	NA	No	No	No
[25]	Flat	High	No	Controlled	Yes	One	NA	No	No	Yes	Yes
[27]	Hierarchical	High	Yes	Controlled	Yes	Multiple	NA	No	Yes	Yes	Yes
[30]	Flat	High	NA	Controlled	Yes	Multiple	Centralized	No	Yes	Yes	Yes

- *Network density*: Some mechanisms are designed for dense sensor networks, while others may aim at sparse networks. For example, the mechanisms proposed by Wu et al. [27] assume a dense network with at least one static and one mobile node in each cluster. In another approach [22], MULEs carry data from sensors to access points, therefore this mechanism can be applied to sparse networks.
- *Delay sensitive*: There is a trade-off between energy efficiency and short delivery latency. For example, using a store-carry-forward mechanism, sensors need to wait for the mobile nodes to come close before data transmission occur, which potentially increases the delivery delay. Such mechanisms are more suitable for delay tolerant applications, while other mechanisms try to increase the network lifetime without a significant delivery latency so that they can be applied to real time monitoring applications.
- *Sensors or relays mobility pattern*: Mobile nodes can follow a random mobility model or alternatively, they can follow a controlled mobility pattern.
- *Known sensor location*: Some mechanisms [2] assume that sensors know their location through GPS or other localization protocols. The location information is used by mobile nodes to determine their new location or it is useful for routing purpose.
- *No. mobile nodes*: Specifies the number of mobile nodes. One or more mobile nodes are exploited in different mechanisms.
- *Algorithm char.*: Some mechanisms are centralized [30]. In this case one node has global information and does all the computations. In sensor networks, this node can be the sink. Centralized mechanisms are expected to return better results, but suffer from bad scalability. Some other mechanisms are distributed or localized, when nodes make decisions without global information. These mechanisms may not achieve the best performance, but they have usually good scalability.
- *Time sync.*: Some mechanisms require time synchronization. For example, in a round based mechanism, sensors use a time synchronization mechanism to compute the starting and ending time of a round.
- *Minimizing moving distance*: Some mechanisms assume that mobile nodes are energy rich, therefore the moving distance and the energy spent on moving are not limited. In other cases, nodes' mobility is limited, such as the flip-based mobility model [30]. In that case, minimizing the energy spent on moving is considered to be a design goal.
- *Routing proposed*: Some mechanisms only focus on the node mobility, while others provide joint mobility and routing mechanisms [25].
- *Connectivity Requirement*: specifies whether sensor nodes must ensure a connected topology. Some schemes require sensor connectivity, while others [22] use mobile nodes to carry data between disconnected network regions.

6. Conclusion

In static wireless sensor networks, sensors close to the sink run out of energy much more faster than sensors in other parts of the monitored area. This causes a limited network lifetime. In this paper, we provide a survey on mechanisms that utilize node mobility to prolong the network lifetime. The mechanisms are classified into three groups: using mobile sinks, using mobile sensors relocation, and using mobile relays. An overview and comparison of these mechanisms are presented.

Acknowledgement

This work was supported in part by the NSF grants CCF 0545488.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.

[2] T. Banerjee, B. Xie, J.H. Jun, D.P. Agrawal, Increasing lifetime of wireless sensor networks using controllable mobile cluster heads, Wireless Communications and Mobile Computing (2009).

[3] Y. Bi, J. Niu, L. Sun, W. Huangfu, Y. Sun, Moving schemes for mobile sinks in wireless sensor networks, in: IEEE International Performance, Computing, and Communications Conference (IPCCC), 2007.

[4] Y. Bi, L. Sun, J. Ma, N. Li, I.A. Khan, C. Chen, HUMS: an autonomous moving strategy for mobile sinks in data-gathering sensor networks, Eurasip (2007).

[5] M. Cardei, Y. Yang, J. Wu, Non-uniform sensor deployment in mobile wireless sensor networks, in: The IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, (WoWMoM'08), June, 2008.

[6] S. Chellappan, X. Bai, B. Ma, D. Xuan, C. Xu, Mobility limited flip-based sensor networks deployment, IEEE Transactions of Parallel and Distributed Systems 18 (2) (2007) 199–211.

[7] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2001.

[8] D. Culler, D. Estrin, M. Srivastava, Overview of sensor networks, IEEE Computer 37 (8) (2004) 41–49.

[9] K. Dantu, M.H. Rahimi, H. Shah, S. Babel, A. Dhariwal, G.S. Sukhatme, Robomote: enabling mobility in sensor networks, IPSN 2005, 2005, pp. 404–409.

[10] J. Edmonds, R.M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, Journal of the ACM 19 (2) (1972) 248–264.

[11] S.R. Gandham, M. Dawande, R. Prakash, S. Venkatesan, Energy efficient schemes for wireless sensor networks with multiple mobile base stations, in: IEEE Global Telecommunications Conference, 2003.

[12] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocols for wireless microsensor networks, HICSS (2000).

[13] B. Hull, V. Bychkovskiy, K. Chen, M. Goraczko, E. Shih, Y. Zhang, H. Balakrishnan, S. Madden, CarTel: a distributed mobile sensor computing system, Proceedings of the SenSys (2006).

[14] S. Jain, R.C. Shah, W. Brunette, G. Borriello, Exploiting mobility for energy efficient data collection in wireless sensor networks, Mobile Networks and Applications (2006) 327–339.

[15] S. Kumar, T.H. Lai, J. Balogh, On k-coverage in a mostly sleeping sensor network, in: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (Mobicom), 2004.

[16] U. Lee, E.O. Magistretti, B.O. Zhou, M. Gerla, P. Bellavista, A. Corradi, Efficient data harvesting in mobile sensor platforms, PerCom Workshops, pp. 352–356, 2006.

[17] J. Li, P. Mohapatra, Analytical modeling and mitigation techniques for the energy hole problem in sensor networks, Pervasive and Mobile Computing 3 (3) (2007) 233–254.

[18] J. Luo, J.-P. Hubaux, Joint mobility and routing for lifetime elongation in wireless sensor networks, in: IEEE INFOCOM, 2005.

[19] M. Marta, M. Cardei, Improved sensor network lifetime with multiple mobile sinks, Elsevier Journal of Pervasive and Mobile Computing 5 (5) (2009) 542–555. October.

[20] S. Olariu, I. Stojmenovic, Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting, in: IEEE INFOCOM, 2006.

[21] E.M. Saad, M.H. Awadalla, R.R. Darwish, A data gathering algorithm for a mobile sink in large-scale sensor networks, in: The Fourth International Conference on Wireless and Mobile Communications, 2008.

[22] R. Shah, S. Roy, S. Jain, W. Brunette, Data mules: modeling a three-tier architecture for sparse sensor networks, in: Proceedings of the IEEE Workshop on Sensor Network Protocols and Applications (SNPA), 2003.

[23] A.A. Somasundara, A. Kansal, D.D. Jea, D. Estrin, M.B. Srivastava, Controllably mobile infrastructure for low energy embedded networks, IEEE Transactions on Mobile Computing 5 (8) (2006). August.

[24] G. Wang, T. Wang, W. Jia, M. Guo, J. Li, Adaptive location updates for mobile sinks in wireless sensor networks, The Journal of Supercomputing 47 (2) (2009) 127–145.

[25] W. Wang, v. Srinvasan, K. Chua, Using mobile relays to prolong the lifetime of wireless sensor networks, in: MobiCom'05, 2005.

[26] K. Wu, Y. Gao, F. Li, Y. Xiao, Lightweight deployment-aware scheduling for wireless sensor networks, ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks 10 (6) (2005) 837–852. December.

[27] J. Wu, S. Yang, Polylogarithmic store-carry-forward routing using mobile nodes, in: Proceedings of IEEE MASS, 2007.

[28] J. Wu, H. Li, An extended localized algorithm for connected dominating set formation in ad hoc wireless networks, IEEE Transactions on Parallel and Distributed Systems (2004) 908–920.

[29] http://www.xbow.com.

[30] Y. Yang, M. Cardei, Movement-assisted sensor redeployment scheme for network lifetime increase, in: The 10th ACM/IEEE Intl. Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM'07), October 2007.

[31] F. Ye, G. Zhong, S. Lu, L. Zhang, Gradient broadcast: a robust data delivery protocol for large scale sensor networks, ACM Wireless Networks (WINET) (2005).