Flow-Balanced Routing for Multi-Hop Clustered Wireless Sensor Networks

Yaling Tao, Yongbing Zhang*

University of Tsukuba,
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573 Japan
{to.are, ybzhang}@sk.tsukuba.ac.jp

Abstract

In this paper, we proposed a *flow-balanced routing* (FBR) protocol for data aggregation in multi-hop wireless sensor networks. A clustering algorithm is firstly proposed to group sensors into clusters based on the overlapping degrees of the sensors. Then, a backbone construction algorithm is proposed to construct a multi-level backbone using the cluster heads and the sink. Furthermore, a flow-balanced routing algorithm is designed to convey the sensed data from the cluster heads to the sink probably via multiple paths. Lastly, a rerouting algorithm is proposed to reconstruct the network topology only at the locations where some cluster heads run out of their energies and drop out of the backbone. The FBR protocol has been evaluated in comparison with previous ones by simulation. The simulation results show that FBR yields much longer lifetime and furthermore better coverage preservation than previous protocols. For example, the lifetime of FBR can be more than 5 times longer than CPCP when the first sensor dies, and meanwhile, the 100% coverage preservation can be 2 times longer in a wide range of parameter settings.

Keywords: multi-hop data aggregation, network clustering, flow-balanced routing, sensor scheduling, overlapping degree, lifetime

1. Introduction

Advances in miniaturization and low-power design have enabled the development of extremely small and low-cost sensors that possess sensing, data processing and transmission capabilities. Wireless sensor networks (WSN) are usually composed of a large number of sensors, which are densely and randomly deployed over inaccessible terrains, and

*Principal corresponding author

are utilized in applications such as environment surveillance and security monitoring[1]. In the most of applications, data sensed by the sensors are sent to a central station, usually called the *sink*, directly (in single hop) or via multiple intermediate sensors (in multi-hop).

One of the sensors' key constraints is the sensor power limitation, and therefore it is important to design an energy-aware protocol in order to prolong the lifetime of a sensor network. Two metrics can be used to show the lifetime of a sensor network [2, 3]: *network lifetime* and *coverage lifetime*. Those metrics indicate the time durations from the beginning instant of the network operation to the instant when some given percentage of sensors die and the ratio of the current coverage by the active sensors to the initial coverage by the whole sensors drops below a predefined threshold, respectively. In this paper, we design a new data aggregation protocol that yields longer network lifetime and coverage lifetime.

In previous researches [19, 20, 10, 11], techniques such as network clustering and sensor scheduling are used for energy conservation for WSNs. By network clustering, several sensors are grouped into a cluster with one head and several members. The members send their sensed data to the head in their cluster and then the head forwards those data to the sink. By sensor scheduling, on the other hand, only some sensors are set to the active mode and used for sensing, while the others whose sensing areas are totally covered by the active sensors are set to the sleep mode. Sensor scheduling is usually realized in the clustering process.

Most of previous approaches focus only on how to group sensors into clusters and select the head in a cluster but not on how to send the collected data from the cluster heads to the sink. There are two common approaches to send the collected data from the cluster heads to the sink: *single hop* and *multi-hop* transmission. In the former, each cluster head aggregates the data from its members and then conveys the data directly to the sink [4, 5, 6]. The heads far away from the sink have to consume more energy to send their data to the sink and consequently die earlier than others. While, in the latter, before transferring the data, the cluster heads are constructed in a tree topology rooted at the sink, and each head sends its data to the sink along a unique path on the tree [7, 8, 9]. Unfortunately, if a head near to the sink has a large number of descendants, it inevitably runs out of its energy quickly and die fast.

In order to alleviate the restriction of a tree structure, a head can be allowed to have more than one path to the sink so that the data are transferred over multiple paths to the sink, equalizing the energy consumption among the sensors. In this paper, we propose a new flow-balanced routing protocol for data aggregation in multi-hop sensor networks. In the new protocol, we first propose a clustering algorithm based on the overlapping degree of each sensor which is defined as the ratio of the overlapping area with other sensors to the whole sensing area of the sensor. Then, we propose a hierarchical network construction algorithm that constructs a multi-level backbone with the sink at the top level and the

cluster heads at lower levels. Furthermore, we design a flow-balanced routing algorithm that balances the transferred data to the sink over multiple paths. In order to reconfigure the network topology with low cost, we propose a local rerouting algorithm that repairs the network topology only at the location any node drops out of the backbone.

The remainder of this paper is organized as follows. Section 2 briefly introduces some important related works. Section 3 describes the network model. Section 4 presents our four algorithms used to construct network and route data over the network. Section 5 shows the performance evaluation by simulation, and Section 6 gives conclusions of our work.

2. Related Work

A number of energy-based data aggregation protocols can be found in the literature [1, 3, 12], and the network clustering is an effective way to save energy for data aggregation. Low-energy adaptive clustering hierarchy (LEACH) [4] is one of the most popular clustering approaches wherein each sensor elects itself to become a cluster heads with a certain probability. Non-head sensors then try to find the nearest head and become its members. The cluster heads act as routers to aggregate and transfer sensed data from their members and send those data to the sink directly. LEACH has some variations such as LEACH-centralized (LEACH-C) [5]. LEACH-C is a centralized clustering algorithm wherein each sensor sends its state information such as its location and current energy to the sink, and then the sink computes the average sensor energy and determines the clusters and the cluster heads depending on the average energy. LEACH and LEACH-C only allow 1-hop transmission, i.e., the cluster heads have to send the data directly to the sink. A multi-hop extension of LEACH, called M-LEACH [7], is proposed wherein sensors can transfer the sensed data to their cluster heads via multiple intermediate sensors (in multi-hop).

Another extension of LEACH called hybrid energy-efficient distributed clustering (HEED) approach [6] takes into account of the combination of energy consumption and communication cost when selecting the cluster heads. Only sensors with high residual energy can be elected to be heads. Even though the heads are well distributed in the service area in HEED, the computation time is extremely long since the probability that a sensor becomes a head is computed iteratively depending on the residual energy of the sensor. Furthermore, each head needs to send the collected data directly to the sink and therefore the heads far away the sink have to consume more energies than others. Although HEED can be extended to a multi-hop network by using a recursive approach similar to [8], the details are not given in [6].

There are some multi-hop data aggregation approaches in the literature [7, 8, 9, 13, 14, 15, 16, 17, 18], wherein the cluster heads are typically constructed in a hierarchical tree

structure. In [18], the cluster heads are constructed in a simple chain structure, while in other approaches, the cluster heads are generally constructed in a hierarchical tree structure. In [14, 15, 16, 17], the hierarchical tree is constructed by using the shortest path from the cluster heads to the sink. The energy aware multi-tree routing (EAMTR) approach [15] attempts to balance the workload of data gathering at cluster heads and avoid using a path through any hotspot. The BioinspirEd backbone selection (BEES) approach [17] divides the service area into a number of hexagons and treat each hexagon as a cluster. The sensor in a hexagon that is closest to the center of the hexagon is selected as the head.

Protocols based on sensor scheduling can be found in [10, 11, 19, 20]. The coverage configuration protocol (CCP) [19] determines the sensors to be in either sleep or active mode. The active nodes have to preserve both the sensing coverage and network connectivity using the coverage eligibility algorithm. The self organize coverage and connectivity protocol (SOCCP) [20] also puts some sensors in sleep mode based on the total coverage-percentage of network and the average energy ratio. The backward of this approach is that the active sensors may be unevenly distributed in the network.

Researches focused on the sensor coverage can be found in [10, 11]. Coverage-aware clustering protocol (CACP) [10] proposed a cost metric that takes into account of the coverage and residual energy in head selection. Each cluster member determines its own state (in either active or sleep mode) based on the network coverage ratio. Each member in active mode can send the sensed data to its cluster head via multiple intermediate sensors, while each cluster head has to send the collected data directly to the sink. The coverage-preserving clustering protocol (CPCP) [11] proposed several coverage costs such as coverage-aware cost, energy-aware cost and coverage redundancy cost, and attempts to balance one of those costs among the cluster heads so that the coverage lifetime is prolonged.

3. Network Model

In this paper, we consider only one sink and a set of sensors, denoted by S, that are deployed randomly over the target field. The target field is indicated as $M \times N$ square units. It is assumed that the sink can reach all the sensors in the target field and has no energy limitation. Each sensor has a given unique identification number and a limited sensing range, denoted by r, which covers a disk area centered this sensor with radius r as shown in Figure 1. The data sensed by a sensor can be transferred to the sink directly in single hop or through other sensors in multi-hop. On the other hand, the transmission range of a sensor, denoted by d, can only be tuned to one of the discrete distances $kR(k = 1, 2, \ldots)$, i.e., $d = kR$ where R denotes a given fixed distance, also called the *cluster range* in this paper. Furthermore, d is limited to be equal or less than the distance between the sensor

and the sink. Therefore, the transmission area of a sensor includes the area with a distance of d away from the sensor as shown in Figure 1.

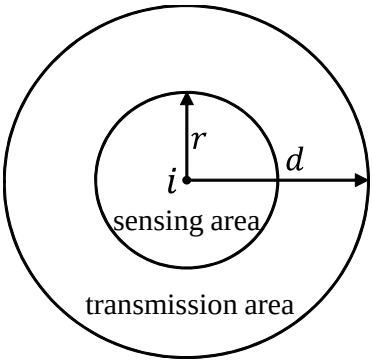


Figure 1: The sensing and transmission area of a sensor.

The sensors within the cluster range of sensor i, R, are called the *neighbors* of sensor i, denoted by N_i. On the other hand, the sensors located in the area with a distance less than $2r$ from sensor i are called the *friends* of sensor i, denoted by F_i, and the sensing areas of those sensors may overlap with that of sensor i. Since sensors are densely deployed in the target field, the coverage area of a sensor may commonly overlap with other sensors. The *overlapping degree* of sensor i, denoted by ρ_i, is defined as the proportion of sensor i's overlapping area with its friends to the whole of sensing area of sensor i as follow.

$$\rho_i = \frac{1}{A_i} \bigcup_{j \in F_i} A_i \cap A_j \tag{1}$$

where A_i denotes the sensing area of sensor i and $A_i \cap A_j$ denotes the overlapping area of sensor i with sensor i's friend j. Obviously, we have $0 \leq \rho_i \leq 1$, and when $\rho_i = 1$ it means that the sensing area of sensor i is totally covered by its friends. Figure 2 illustrates an example of the overlapping area of sensor i with its two friends, j and k, and the overlapping degree ρ_i equals $\frac{(A_i \cap A_j) \cup (A_i \cap A_k)}{A_i}$. To calculate the overlapping area, an approach proposed in [21] can be used and there are also some approximation methods in [10, 19, 20, 17, 22].

In previous approaches such as LEACH [4] and HEED [6], the cluster formation is performed periodically, i.e., in every round. However, in this paper, the clusters are formatted only once based on the overlapping degrees of the sensors at the beginning of the network operation. The network topology will not be changed unless a cluster-head dies or loses the connection to the network. Sensors in the network are classified into three classes,

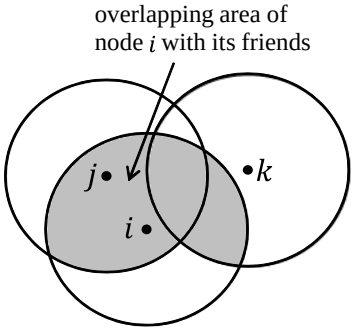


Figure 2: Overlapping areas of sensors.

cluster heads, *waiting nodes*, and *member nodes*, denoted by S_b, S_w, and S_o, respectively. The cluster heads are used to construct the backbone and each cluster head collects and aggregates the data from the sensors in its cluster, called its *members*, and conveys the aggregated data to the sink. A sensor with the largest overlapping degree will be selected as a head, and therefore a sensor whose overlapping degree equals 1 will be selected as a head with higher probability than others. A head with $\rho = 1$ will be used only for data aggregation and transmission but not sensing since its coverage area is totally overlapped by its friends. Furthermore, a node with $\rho = 1$ other than the head will be treated as a waiting node in sleep mode. A waiting node does nothing but just wait for the *HELP* message to replace the exhausted nearby head on the backbone. A member node whose overlapping degree is less than one sends its sensed data directly to the head in its cluster.

The backbone is composed of the cluster heads and the sink and a backbone example is shown in Figure 3. In this figure, the backbone nodes and the waiting nodes are indicated by the black and the gray points, respectively. The member nodes, on the other hand, are shown by the white circles. From Figure 3, we can see that the backbone is not a simple tree but a hierarchical topology and a node may have multiple parents belonging to the same level. Each node on the backbone can only send data to the sink via its parent(s) and there may exist multiple paths from a node to the sink.

4. Proposed Algorithms

Data aggregation from sensors to the sink is performed in two phases, *route construction* and *data transmission*. In the route construction phase, the sensors are grouped into clusters based on their overlapping degrees, $\rho_i (i \in S)$, and then a hierarchical backbone is constructed using the cluster heads with the sink at the top. In the data transmission

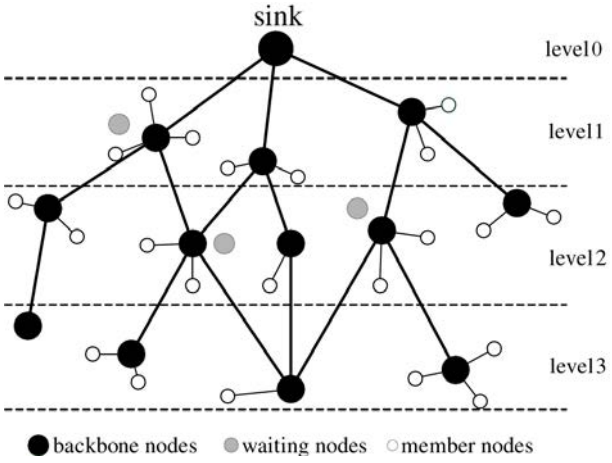


Figure 3: Network model.

phase, the sensors send their sensed data to their cluster heads and then the heads forward the data to the sink via multiple paths in the backbone. When a head runs out of its energy or its residual energy becomes lower than a predefined threshold, it will drop out of the backbone and the backbone will be reconfigured.

4.1. Cluster Formation Algorithm

Unlike previous approaches, the cluster formation in FBR is performed only once at the beginning of network operation. The sink broadcasts a *CLS_FORM* message to inform all the sensors to start the cluster formation. Each sensor calculates its own overlapping degree and broadcasts it to its neighbors. Therefore, each sensor knows the overlapping degrees of its neighbors. A sensor with a higher overlapping degree has a higher probability to become the cluster head, and for two sensors with the same overlapping degree, the one with a smaller *id* number has higher priority. For the sake of discussion, it is assumed that no more than two sensors send messages at the same time and the transmission delay of the control message between two sensors is negligibly small. When there is more than one sensor with $\rho = 1$ in a sensor's sensing range, those sensors bid for the cluster head. The winner becomes the head and broadcasts a *HEAD* message to its neighbors. A sensor with $\rho = 1$ receiving the *HEAD* message will try to become a waiting node based on its *id* number and, if succeeds, broadcasts a *SLEEP* message to its friends.

On the other hand, for a sensor, e.g., sensor i with $\rho_i < 1$, if it receives a *HEAD* message from its neighbor, e.g., sensor j, then sensor i joins S_o and becomes a member of sensor j; otherwise, sensor i selects an unclassified sensor j ($j \in N_i, \rho_j > \rho_i$), as its head by sending a *SEL_HEAD* message to sensor j. When a sensor receives the *SEL_HEAD* message, it

will become the cluster head. After determining the class of a sensor, its overlapping degree will not be changed. However, when a sensor whose class has not been determined receives the *SLEEP* message from a friend j, it recalculates its overlapping degree by removing sensor j. The cluster formation algorithm is given in Algorithm 1.

In cluster formation, each sensor i only needs exchange the information of its location, state, and overlapping degree with its neighbors within the distance R and with its friends within the distance $2r$. In the worst case where all of the other nodes are the neighbors of node i, node i needs to exchange the information with $|S|$ neighbors. Therefore, the number of the exchanged messages for node i is bound by $O(|S|)$.

4.2. Backbone Construction Algorithm

The backbone is constructed by the cluster heads with the sink at the top. The sink initially broadcasts a *BN_CONST* message with a tuple (k, l, id) where k denotes the parameter used for deciding the transmission distance $d\,(= kR)$ while l and id denote the level and the id number of the message sender, respectively.

When a cluster head, e.g., node i, receives the *BN_CONST* message at the first time, it will wait a time interval, denoted by τ_0, for any possible message from other nodes that have already been on the backbone. When τ_0 expires, node i determines its parents P_i where P_i denotes the set of nodes whose levels are the highest among those node i have received. After joining the backbone, node i broadcasts the *BN_CONST* message with distance d, and sends the sink an *ON_BN* message to inform the sink its status and therefore the sink knows whether the current backbone includes all the sensors. The sink will wait the *ON_BN* messages from the new heads for a time interval denoted by τ_s. If there is still any head not on the backbone when τ_s expires, the sink increments the value of k and then asks the backbone nodes to broadcast the *BN_CONST* message again. The value of k will be increased until the *BN_CONST* message reaches a new head that has not been added to the backbone yet. Note that a backbone node may have multiple parents and once it connects to the backbone, its parent(s) will not be changed.

The backbone construction algorithm is given in Algorithm 2. The backbone construction process of a sample network using Algorithm 2 is shown in Figure 4. The sink initially set $k = 1$ and asked backbone nodes to search for new heads, and node i received *BN_CONST* messages from nodes 2, 3, and 4 (Figure 4(a)) and then determined its parents, nodes 2 and 3 (Figure 4(b)). Thereafter, node i broadcasted *BN_CONST* message with $d(= R)$ but no new head could be found. After τ_s expired, the sink incremented k and asked the backbone nodes to do the search again, and node j received *BN_CONST* messages from nodes 2, 4 and i as shown in Figure 4(c).

In backbone construction, if the *BN_CONST* message cannot reach node i with a given k, the sink will increment the value of k. By assuming that the distance between node i and the sink to be d_{is}, we have $k \leq \lceil \frac{d_{is}}{R} \rceil$. For a network with high density, a node can easily

Algorithm 1 Cluster Formation Algorithm.

Initialization

find friends and neighbors
calculate ρ_i according to eq. (1) and broadcast ρ_i to neighbors
after received all ρ_j ($j \in N_i$), do **Classification**

Classification

if $\rho_i = 1$ **then**
  bid for the head if the cluster head has not been determined
  if won the bid **then**
    join S_b and broadcast *HEAD* message to neighbors
  else if $i < j$ ($j = \max(\rho_k), k \in N_i, k \notin (S_w \cup S_b)$) **then**
    join S_w and broadcast *SLEEP* message to friends
  else if received *SLEEP* message from a friend j **then**
    do **Recalculation** and **Classification**
  end if
else
  if received *HEAD* message from j ($j \in N_i$) **then**
    join S_o and become member of j
  else if cluster head has not been determined **then**
    if $\rho_i \geq \rho_j (j \in N_i)$ **then**
      join S_b and broadcast *HEAD* message to neighbors
    else
      find j such that $j = max_{k \in N_i}(\rho_k)$
      join S_o and become member of j
      send *SEL_HEAD* message to j
    end if
  else if received *SEL_HEAD* **then**
    join S_b and broadcast *HEAD* message to neighbors
  end if
end if

Recalculation

if sensor i's classification has not been determined **and** received *SLEEP* message from j **then**
  recalculate ρ_i
end if

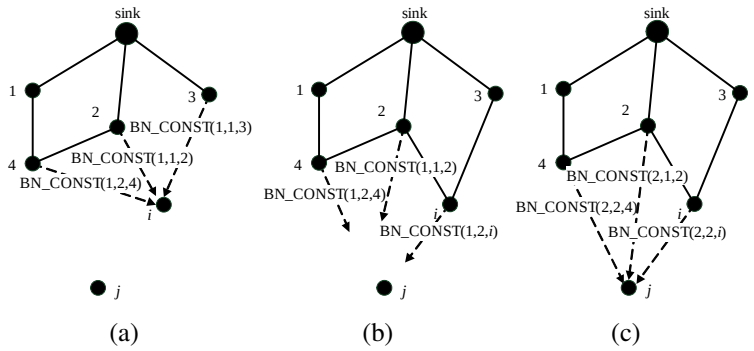


Figure 4: An example of a backbone construction.

Algorithm 2 Backbone Construction Algorithm.

Procedures executed by sink

broadcast $BN\_CONST(k = 1, l = 0, sink\_id)$ with $d\,(= kR)$
reset timer t
if not receive any ON_BN message until $t \geq \tau_s$ **then**
    $k \leftarrow k + 1$ and broadcast $UPDATE(k)$ to all the backbone nodes
    go to step 2
end if

Procedures executed by a head

if receive $BN\_CONST(k, l, id)$ at the first time **then**
    reset timer t
    record all BN_CONST messages until $t \geq \tau_0$
    select backbone node(s) j with the lowest level (l_{min}) as parent(s) and put it(them) in P_i
    $l_i \leftarrow l_{min} + 1, k_i \leftarrow k$
    send ON_BN message to sink
    broadcast BN_CONST with a tuple (k, l_i, i) and distance $d\,(= kR)$
end if
if already on backbone **and** receive $UPDATE(k)$ from sink **then**
    broadcast BN_CONST with a tuple (k, l_i, i) and distance $d\,(= kR)$
end if

find a neighbor and so k is typically small. For example, in our simulation model with a $150 \times 150 \text{m}^2$ target field, 1000 sensors, and the sink at (150, 150), when we set $R = 10$, we have $k \leq 3$.

4.3. Flow-Balanced Routing Algorithm

Each node on the backbone, i.e., cluster head, collects the sensed data from its members and aggregates those data, and then conveys the data to the sink. A backbone node may have multiple parents and therefore multiple paths to the sink as shown in Figure 5. In this section, a flow-balanced routing algorithm is proposed to balance the traffic flow from a head to the sink via multiple paths.

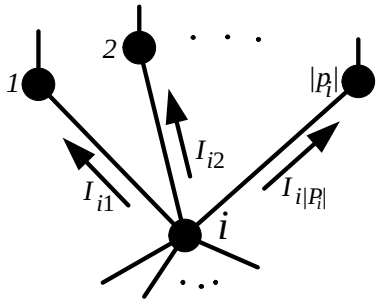


Figure 5: Multiple paths from node i to the sink.

The energy needed to send one bit data from a sensor to another one can be calculated by using the $1/s^n$ path loss model [23] as follows.

$$P_{relay}(s) = (\alpha_1 + \alpha_2 s^n)\gamma, \tag{2}$$

where s is the transmission distance, α_1 is the total energy per bit consumed by the transmitter and the receiver electronics, α_2 accounts for energy dissipated in the transmit op-amp, γ is the number of bits relayed per second, n is the path loss exponent, typically n takes the value between 2 and 5. Similar to [4], we calculate the transmission energy by using the free space (s^2 power loss) and the multi-path fading (s^4 power loss) channel models as follows.

$$E_t = \begin{cases} (E_e + \epsilon f s^2)I, & \text{if } s < s_0, \\ (E_e + \epsilon m s^4)I, & \text{if } s \geq s_0, \end{cases} \tag{3}$$

where $s_0 = \sqrt{\frac{\epsilon f}{\epsilon m}}$, and E_e is equivalent to α_1 in (2) and we set it to 50nJ/bit. Since the power control can be used to invert this loss by appropriately setting the power amplifier, if the distance is less than a threshold s_0, the free space (fs) model is used, that is, $n = 2$

and α_2 is set to $\epsilon f = 10pJ/bit/m^2$. Otherwise, the multipath (mp) model is used, that is, $n = 4$ and α_2 is set to $\epsilon m = 0.0013pJ/bit/m^4$. The energy needed to receive I-bit data can be calculated by

$$E_r = E_e I. \tag{4}$$

Since each backbone node may have multiple parents, the energy needed to send a given size message to each of its parents needs to be estimated. Let $E_j\,(j \in P_i)$ to denote the residual energy of node i's parent j. Assuming that node i has I-bit data, denoted by I_i, to the sink and that the flow from node i to its parent j is denoted by I_{ij}, then we have

$$I_i = \sum_{j \in P_i} I_{ij}. \tag{5}$$

Therefore, we can write the energy/bit consumption for conveying data I_{ij} at node j as

$$\Delta E_{ij} = (\alpha_1 + \alpha_2 (k_j R)^n) I_{ij} = \varepsilon_j I_{ij}, \tag{6}$$

where $\varepsilon_j = (\alpha_1 + \alpha_2 (k_j R)^n)$. Therefore, the residual energy of node i's parent j, denoted by X_j, after conveying data I_{ij} can be written as

$$X_j = E_j - \Delta E_{ij} = E_j - \varepsilon_j I_{ij}. \tag{7}$$

The goal of the proposed routing algorithm is to balance the residual energy of the backbone nodes in order to prolong the network lifetime, that is, to minimize the difference of the residual energy between the parents of sensor i after sending data I_i. That is, we attempt to realize the following goal.

$$X_j = \bar{E} = \frac{1}{|P_i'|} \sum_{l \in P_i'} X_j, \quad j \in P_i', \tag{8}$$

where $P_i'\,(P_i' \subseteq P_i)$ denotes the set of node i's parents to which the flow from node i is greater than 0, i.e., for $E_j > \bar{E}\,(j \in P_i'), I_{ij} > 0$. According to eqs. (5), (6), (7), and (8), we can determine I_{ij}. On the other hand, for $E_j \leq \bar{E}\,(j \in P_i \setminus P_i')$, the flow from node i to node

j should be 0, i.e., $I_{ij} = 0$. Therefore, we have for $j \in P_i$

$$I_{ij} = \begin{cases} 0, & \text{if } E_j \leq \bar{E}, \\ \dfrac{I_i + E_j \sum\limits_{l \in P'_i} \dfrac{1}{\varepsilon_l} - \sum\limits_{l \in P'_i} \dfrac{E_l}{\varepsilon_l}}{\varepsilon_j \sum\limits_{l \in P'_i} \dfrac{1}{\varepsilon_l}}, & \text{if } E_j > \bar{E}. \end{cases} \tag{9}$$

The proposed routing algorithm is executed by each node to determine the flow to each of its parents that satisfies eq. (9).

Algorithm 3 Flow-Balanced Routing Algorithm.

1: get residual energy of parents $E_j (j \in P_i)$
2: $P'_i \leftarrow P_i$
3: calculate I_{ij}
4: **if** there is any parent $j, I_{ij} < 0$ **then**
5: set $I_{ij} = 0$ and remove j from P'_i
6: goto step 3
7: **end if**
8: Obtain $I_{ij} (j \in P_i)$ that satisfy eq. (9)

In flow-balanced routing, the residual energy of node i's parents are attempted to be equalized after data transmission. The calculation of I_{ij} (line 3 in Algorithm 3) plays a key role in determining the capable parents to which node i can send some data. Firstly, node i calculates the total energy needed to convey data i and estimates the average energy of its capable parents by ignoring those parents with energy lower than the average. This process is repeated until all the capable parents have equal or more energy than the average energy. The computational complexity of this process is bound by $O(|P_i|^2)$. Note that only the total residual energy of node i's parents are considered here, and if node i's parents do not have enough energy to convey the collected data, node i is regarded as an isolated node with no capable parent. Then, the isolated node reconnection mechanism described in Section 4.4 will be triggered.

4.4. Rerouting Algorithm

Instead of reconstructing the whole network in each round, we propose a local rerouting algorithm to reconfigure the backbone only around the location the topological change occurs; that is, at where a node drops out of the backbone due to its energy exhaustion. If the residual energy of a backbone node becomes lower than a predefined threshold denoted

by r_{th}, e.g., a percentage of sensor's initial energy, the node becomes energy exhausted and the rerouting algorithm is triggered. When an exhausted backbone node drops out of the backbone, it will try to find a new head to replace itself; otherwise, its descendants will lose the connection to the backbone. In order to deal with those cases, we propose a rerouting algorithm that has two phases, *head replacement* and *isolated node reconnection*.

In the *head replacement* phase, the exhausted head, e.g., node i, broadcasts a *HELP* message to backbone nodes within R to find a capable neighbor to replace itself. Each neighboring backbone node responds to the request with its own residual energy if the ratio of its residual energy to its initial energy (E_0) is greater than r_{th}, and then node i selects the node with the most energy as the candidate to replace itself. Since there may be more than one waiting node in a cluster, the waiting node with the smallest *id* number will be selected with the highest priority. After determining the new head, node i informs the result to its children, and then those children try to connect to the new head. Node i also becomes a member of the new head. In a worse case, if node i cannot find any capable node to replace itself, it involuntarily throws away its children and becomes a member of another node. The abandoned descendant(s) of node i become isolated orphans and they have to find their new head by themselves.

In the *isolated node reconnection* phase, a node who realized its head has gone tries to reconstruct the connection to the backbone. It broadcasts a *RECON* message with distance $d\,(= kR)$ to find a capable backbone node for connection. If there is no any response when a time duration τ expires, it will increment k and broadcast the *RECON* message again. The search for a new head is done based on the following rules.

- The new head should be located in the cluster range, R, and its residual energy should be greater than a predefined energy threshold. Otherwise, the isolated node becomes a new head.
- A waiting node with the smallest *id* number has the highest priority to be a new head.
- The old head is only responsible for finding a candidate, the members and children of the old head have to determine the new head by themselves.
- The new head can only be selected from a backbone node with the level not lower than the old head.

The rerouting algorithm is illustrated in Algorithm 4.

Algorithm 4 Rerouting Algorithm.

Head Replacement executed by i

broadcast *HELP* message with distance R
if receive responses from neighbors $j\,(j \in N_i)$ **then**
    select the best node j according to rules described in (4.4)
    if node j is not a backbone node **then**
        send *CON_BN* message to node j with $k\,(= k_i)$
    end if
    inform members and children to connect to new head j
    join S_o and become a member of node j
else
    inform members and children do **Isolated Node Reconnection**
end if

Isolated Node Reconnection

for each node m with no head **do**
    if a backbone node $j(j \in N_m)$ can be found **then**
        Take node j as the new head or the new parent
    else
        if find a node $j\,(j = \arg\max(E_k), k \in N_i, E_k/E_0 > r_{th}, E_k > E_m)$ **then**
            send *CON_BN* message to node j with $k\,(= k_m)$
        else
            do **Connect To Backbone**
        end if
    end if
end for
for each head h with no parent **do**
    do **Connect To Backbone**
end for

Connect To Backbone

if receive *CON_BN* message **then**
    Obtain k from *CON_BN* message
end if
broadcast *REQ_CON* message to find a backbone node with distance $d\,(= kR)$ and k
if no response from any backbone node **then**
    increment k and goto step 4
else
    connect to the responded node(s)
end if

Table 1: Parameters used in simulation

Parameter name	Symbol	Value
Field size		150×150 m^2
Sink location		(150,150)
Number of sensors	S	1000
Sensing range	r	5m
Cluster size	R	10m
Initial energy	E_0	0.5J
Transmission energy/bit	E_e	50nJ/bit
Amplifier energy(fs)	ϵf	10pJ/bit/m^2
Amplifier energy(mp)	ϵm	0.0013pJ/bit/m^4
Data size	I	2000bits
Message size	msg	100 bits
Data compression ratio	r_{dc}	30%
Energy threshold	r_{th}	30%
Head percentage (LEACH)	p	5%
Head percentage (HEED)	C_{prob}	5%
Energy threshold (HEED)	P_{min}	10^{-4}J

5. Simulation

In the simulation experiments, a network model with a 150×150 m^2 square area was used. The sensors were randomly and uniformly deployed in the network, while the sink was located at the position (150, 150). Time in the experiments was proceeded in round as previous researches, and the data from each sensor to the sink were assumed to be 2000 bits. The size of the control messages exchanged between two sensors and between a sensor and the sink were assumed to be all the same and were 100 bits.

Our proposed FBR protocol has been compared with previous approaches, LEACH, HEED, M-HEED, CPCP, and HEED-FBR, using two performance metrics, *network lifetime* and *coverage lifetime*. The network lifetime is defined as the time duration from the beginning instant of the network operation to the instant when any or a given percentage of the sensors die. On the other hand, the coverage lifetime is defined as the time duration from the beginning instant of the network operation to the instant when the ratio of the coverage of the current alive sensors to the coverage of the whole sensors drops below a predefined threshold. The M-HEED approach is an extended version of HEED developed in this paper wherein the cluster heads are constructed hierarchically using the recursive approach proposed in HEED. Furthermore, HEED-FBR is a modified version of FBR wherein the cluster formation algorithm is replaced by HEED. The above mentioned protocols are also examined by varying the system parameter values such as sensors's

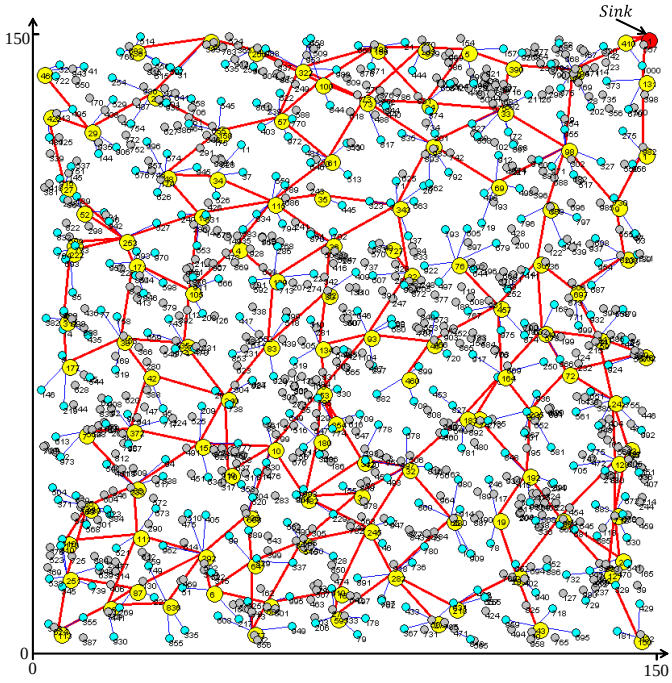


Figure 6: Clustered network topology constructed using FBR.

sensing range, head energy threshold, and so on.

5.1. Lifetime Comparison

In order to avoid any unfair treatment over previous approaches, we simulated those protocols with a wide range of parameter settings and chose the best parameter combinations for the comparison. The parameters used in the experiments are given in Table 1, and the network topology obtained using our FBR protocol is shown in Figure 6. From this figure, we see that the backbone topology is a novel multi-level structure rather than a simple tree where each node may have multiple paths to the sink.

Figure 7 shows the network lifetimes of our FBR protocol compared with LEACH, HEED, M-HEED, CPCP and HEED-FBR. We see from this figure that CPCP behaves better than other traditional approaches and FBR yields a much longer lifetime than others. The lifetime of FBR when the first sensor died is near 10 times longer than CPCP and is around 5 times longer when 10 percent of the sensors died. Furthermore, we can see that HEED-FBR also provides a long lifetime. The difference between the lifetimes of FBR and HEED-FBR shows the usability of the proposed cluster formation approach.

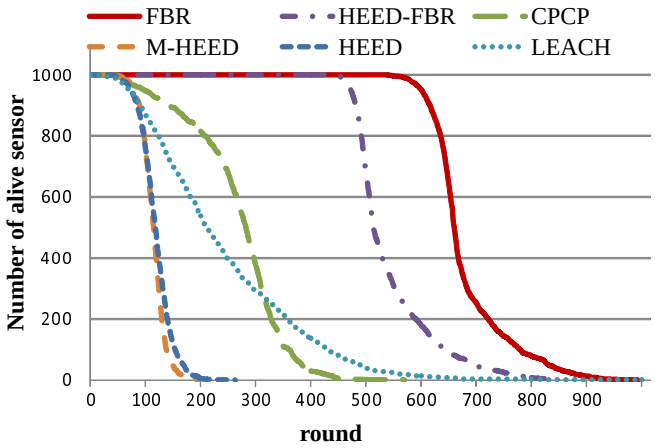


Figure 7: Network lifetimes of various protocols.

Similarly, by comparing HEED-FBR and HEED, we see that the flow-balanced routing algorithm plays a key role in data aggregation and balancing flows between nodes yields a long lifetime.

Figure 8 shows the coverage lifetimes of FBR along with HEED, CPCP, and HEED-FBR. We selected these previous protocols for comparison because both HEED and CPCP are coverage-aware protocols [11]. From this figure, we see that both FBR and HEED-FBR yield much longer coverage lifetimes than others.

The main reasons for the above results can be summarized as follows.

1) The backbone constructed in FBR is not a simple tree but a multi-level structure with the sink at the top. Each head may have multiple paths to the sink and therefore balancing the flow from a sensor to the sink over multiple paths can equalize the energy consumption among the heads, resulting in a longer lifetime. On the other hand, in the multi-level protocols extended based on HEED, the cluster heads at each level are attempted to be distributed uniformly in the network, causing some higher level heads far away from the sink. Those heads have to spend more energy to send data to the sink and die fast. Furthermore, in CPCP the cluster heads are simply constructed as a shortest path tree rooted at the sink. A head near to the sink and with more offspring should die quickly.

2) A local rerouting approach is used in FBR (and HEED-FBR) to repair the backbone topology only at the location where the topological changes occur. In the previous algorithms, on the other hand, the network construction is repeatedly executed at the beginning of each round.

3) In large-scale sensor networks, a large number of sensors are usually deployed ran-

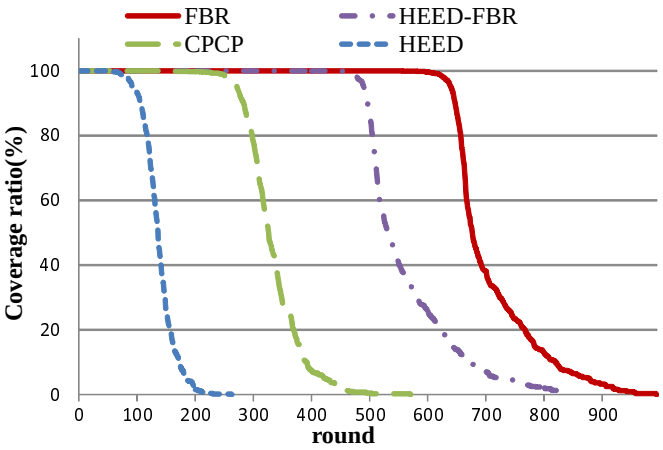


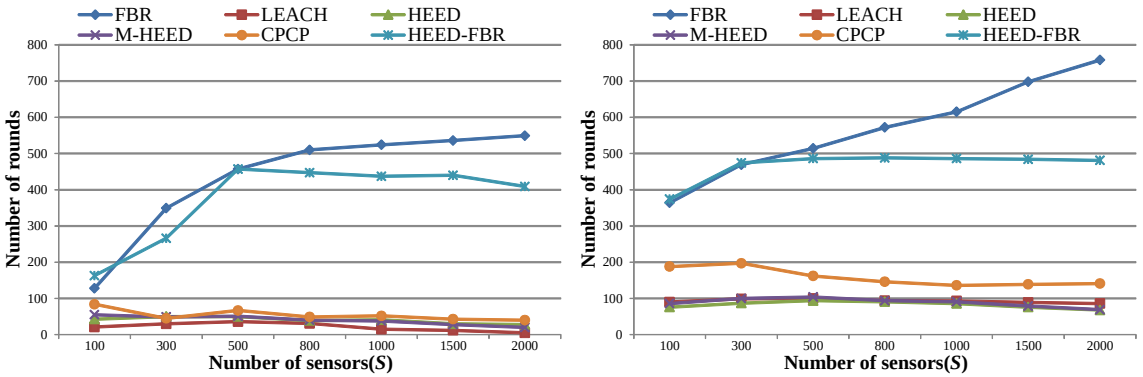
Figure 8: Coverage lifetimes of various protocols.

domly in the target field. The coverage areas of some sensors may be totally overlapped by others. Taking out of the overlapped sensors will not hurt the usability of the network at all. In FBR, the overlapped sensors are taken as *waiting nodes*, that is, let those sensors to be in sleep mode to reduce energy consumption.

5.2. Parameter Examination

In order to further examine the effects of the system parameters on the performance of our proposed protocol, we simulated FBR and also main previous protocols with various parameter settings as follows. For the sake of space limitations, only the network lifetimes of the protocols under consideration are shown in the figures when the first sensor or ten percent of the sensors died. Note that when changing a parameter value, the others are fixed as shown in Table 1.

- S: number of sensors. The number of sensors were set to 100, 300, 500, 800, 1000, 1500, and 2000.
- r: Sensing range. The values of sensing range were set to 2, 5, 8, 10, and 15 in order to evaluate how our protocol behaves.
- R: Cluster range. This is a key parameter both in cluster formation and in backbone construction and its effect on the performance, i.e., the network lifetime was examined. The values of R were set to 5, 10, 15 and 20.



(a) Lifetime when the first sensor died.

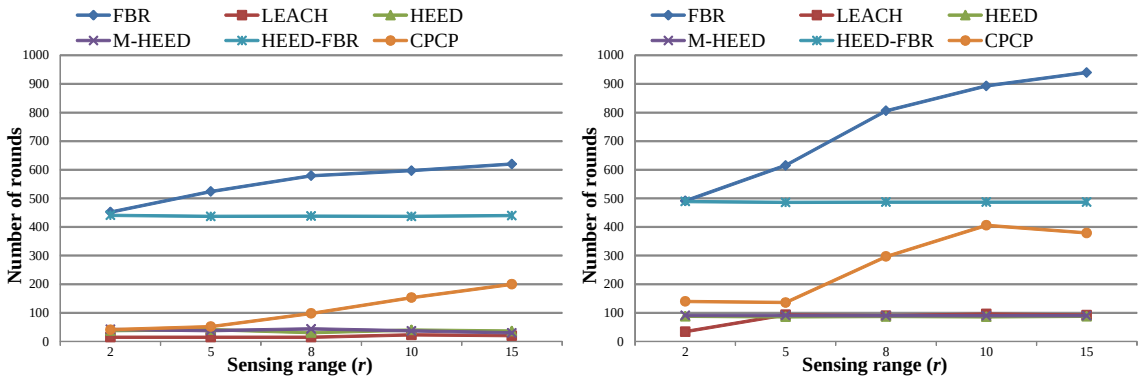
(b) Lifetime when 10% sensors died.

Figure 9: Lifetimes for various network sizes.

- r_{th}: Energy threshold ratio. It is the ratio of the current energy of a head to its initial energy. It works similarly to parameter p_{min} in HEED and CPCP but is different from them in the sense that r_{th} can be adjusted to adapt to different conditions. The values of r_{th} were to 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.8, and 0.9.

- r_{dc}: Data compression ratio. It is the ratio of the size of the aggregated data at a head to the total size of the original data received from its members along with its own sensed data. When a head receives nI-bit data, then it has to send $r_{dc}(n+1)I$-bit data to the sink. The values of r_{dc} were set to 0.2, 0.3, 0.4, 0.5, and 0.6. Furthermore, we also simulated a special case, similar to LEACH, HEED and CPCP, wherein the data collected at a node is aggregated into one packet no matter how large the size of the total collected data is. The result of this case is shown by η in Figure13.

Figure9 illustrates the lifetimes of the algorithms under consideration when changing the number of sensors. We see that when the number of sensors increase FBR performs better than others, because more nodes are treated as waiting nodes, reducing more energy consumption, and there may be multiple paths between each node and the sink, realizing the flow balancing over the multiple paths. Although, CPCP performs the sensor scheduling to put some sensors in sleep mode, the scheduling phase is assigned after the cluster formation phase, while these sleep nodes do not need joining the cluster formation work at all. Furthermore, when the number of sensors increase, the network construction overhead also increases, so rerouting locally would work more efficiently than reconstructing network globally, obviously.



(a) Network lifetime when the first sensor died. (b) Network lifetime when 10% sensors died.

Figure 10: Lifetimes for various sensing ranges.

Figure10 shows the lifetimes of the algorithms for various sensing ranges. We see that FBR performs better as the sensing range increases because when the coverage area of a sensor increases, the overlapping ratio of a sensor also increases. Furthermore, a sensor can find more friends in its widened sensing area and a better head may also be chosen.

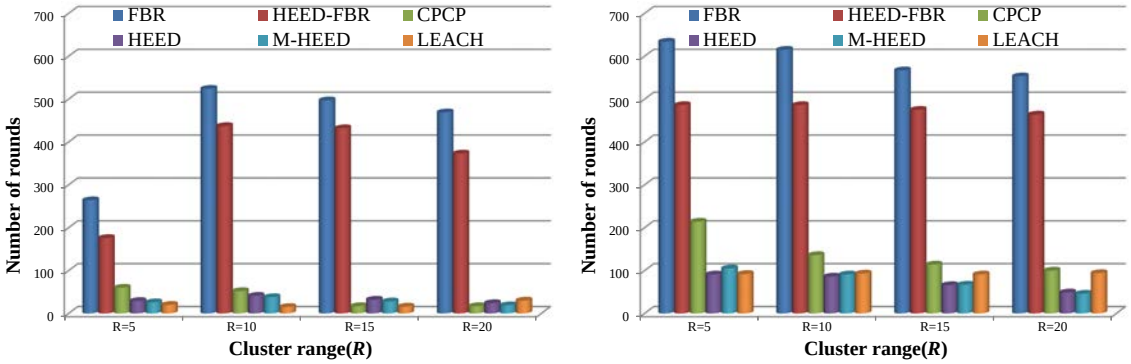
Figure11 shows the lifetimes of the algorithms for different cluster ranges. We see that the cluster range, also used in HEED and M-HEED and CPCP, has greater effect on the performance.

Figure12 shows the lifetimes for different head energy threshold ratios, we find that the r_{th} is a sensitive parameter for FBR while in other algorithms the effect of this parameter can be negligible. From Figure12, we see that the best performance of FBR is realized when r_{th} is around 0.25-0.4. Since r_{th} is tunable parameter, we can adjust its value depending on the network configuration.

Figure13 shows the lifetimes of the algorithms when changing data compression ratios. In this figure, η denotes an extreme case no mater how many data packets a node receives it will aggregate them into one packet. We see from this figure that FBR behaves better than the others and that the network lifetime decreases when the compression ratio becomes low.

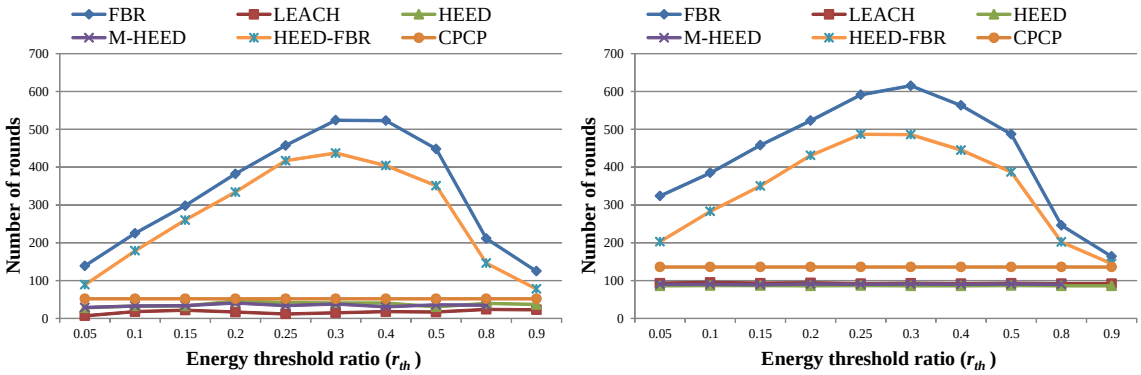
6. Conclusion

In this paper, we have proposed a new flow-balanced routing (FBR) protocol for multi-hop clustered wireless sensor networks. In FBR, the cluster formation is performed only once at the beginning of the network operation and is determined based on the the overlapping degrees of sensors. Some sensors whose sensing areas are covered by others are put



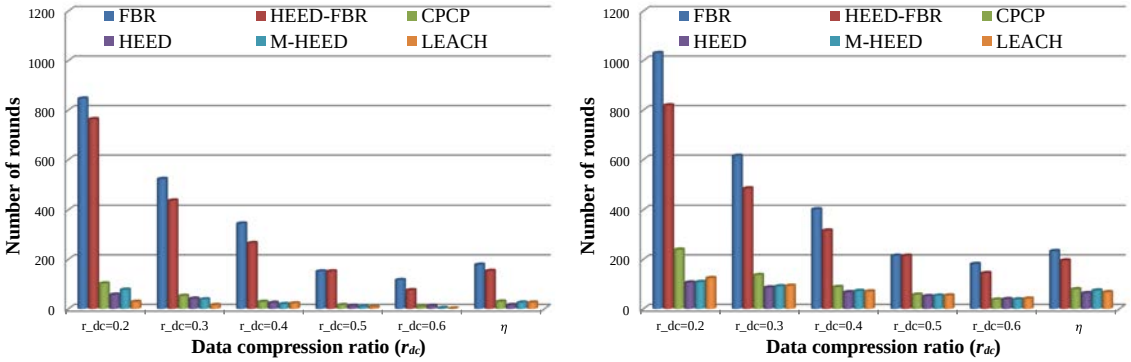
(a) Network lifetime when the first sensor died. (b) Network lifetime when 10% sensors died.

Figure 11: Lifetimes for various cluster ranges.



(a) Network lifetime when the first sensor died. (b) Network lifetime when 10% sensors died.

Figure 12: Lifetimes for various head energy thresholds.



(a) Network lifetime when the first sensor died. (b) Network lifetime when10% sensors died.

Figure 13: Lifetimes for various data compression ratios.

in sleep mode in order to save energy. The cluster heads are constructed in a multi-level architecture with the sink at the top and there may be multiple paths from each head to the sink. Based on this novel network architecture, a flow-balanced routing algorithm is proposed to equally balance the flow from a head to the sink over multiple paths. Furthermore, a local rerouting algorithm is proposed to reconfigure the network topology only at location where any topological change occurs due to the dropouts of exhausted sensors.

The proposed protocol, FBR, has been evaluated in comparison with previous protocols, LEACH, HEED, CPCP, and also two modified versions of HEED, i.e., M-HEED and HEED-FBR, using simulation. The results show that FBR yields longer network lifetime and also longer coverage lifetime than other protocols. The network lifetime of FBR can be more than 5 times longer than CPCP, the best among the previous protocols under consideration, and at the same time the coverage lifetime can be 2 times longer. Furthermore, the effects of the parameters have been examined with a wide range of parameter settings, and FBR always behaves better than others.

Acknowledgement

This work is supported in part by Grand-in-Aid for Science Research (C) 21500069, Japan Society for the Promotion of Science.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks a Survey," *Computer Networks*, Vo1. 38, No. 4, pp. 393-422, March 2002.

[2] I. Dietrich and F. Dressker, "On the Lifetime of Wireless Sensor Networks," *ACM Trans. Sensor Networks (TOSN)*, Vol. 5, No. 1, February 2009.

[3] A.A. Abbasi and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks," *Computer Communications*, Vol. 30, No. 14, pp. 2826-2841, October 2007.

[4] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. 33rd Hawaii Int. Conf. System Sciences (HICSS)*, Vol. 2, pp. 1-10, January 2000.

[5] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wireless Communication*, Vol. 1, No. 4, pp. 660-670, October 2002.

[6] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid Energy-Efficient Approach," *Proc. IEEE INFOCOM*, Vol. 1, March 2004.

[7] V. Mhatre and C. Rosenberg, "Homogeneous vs Heterogeneous Clustered Networks: A Comparative Study," *Proc. IEEE ICC*, Vol. 6, pp. 3646-3651, June 2004.

[8] S. Bandyopadhyay and E. Coyle, "An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *Proc. IEEE INFOCOM*, Vol. 3, pp. 1713-1723, April 2003.

[9] H.O. Tan and I. Korpeoglu, "Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks," *Newsletter ACM SIGMOD Record*, Vol. 32, No. 4, December 2003.

[10] B. Wang, H.B. Lim, D. Ma, and D. Yang, "A Coverage-Aware Clustering Protocol for Wireless Sensor Networks," *Proc. Mobile Ad-hoc and Sensor Networks (MSN), 2010 Sixth Int. Conf.*, Vol. 1, pp. 85-90, December 2010.

[11] S. Soro and W.B. Heinzelman, "Cluster Head Election Techniques for Coverage Preservation in Wireless Sensor Networks," *Ad Hoc Networks*, Vol. 7, No. 5, pp. 955-972, July 2009.

[12] T. Meng and R. Volkan, "Distributed Network Protocols for Wireless Communication," *Proc. IEEE ISCAS*, Vol. 4, pp. 600-603, May 1998.

[13] M. Perillo, Z. Cheng, and W. Heinzelman, "On the Problem of Unbalanced Load Distribution in Wireless Sensor Networks," *Proc. IEEE GlobeCom Workshops*, pp. 74-79, December 2004.

[14] P.H. Hsiao, A. Hwang, H.T. Kung, and D. Vlah, "Load-Balancing Routing for Wireless Access Networks," *Proc. IEEE INFOCOM*, Vol. 2, pp. 986-995, April 2001.

[15] H. Fariborzi, and M. Moghavvemi, "EAMTR: Energy Aware Multi-Tree Routing for Wireless Sensor Networks," *IET Commun.*, Vol. 3, pp. 733-739, May 2009.

[16] F. Ren, J. Zhang, T. He, C. Lin, and S.K. Das, "EBRP: Energy-Balanced Routing Protocol for Data Gathering in Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, Vol. 22, No. 12, pp. 2018-2125, December 2011.

[17] H.S. AbdelSalam and S. Olariu, "BEES: BioinspirEd BackbonE Selection in Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, Vol. 23, No. 1, pp. 44-51, January 2012.

[18] S. Lindsey and C.S. Ranghavendra, "PEAGSIS: Power-Efficient Gathering in Sensor Information Systems", *Proc. IEEE Aerospace Conf.*, Vol. 3, pp. 1125-1130, March 2002.

[19] X. Wang, G. Xing, Y. Zhang, and C. Lu, "Integrated Coverage and Connectivity Configuration in Wireless Networks," *ACM Transactions on Sensor Networks*, Vol. 1, No. 1, pp. 36-72, August 2005.

[20] H. Khosravi and L. Aslanyan, "SOCCP: Self Organize Coverage and Connectivity Protocol," *Proc. 2011 Third Int. Conf.*, Vol. 1, pp. 317-322, September 2011.

[21] F. Librino, M. Levorato, and M. Zorzi, "An Algorithmic Solution for Computing Circle Intersection Areas and Its Applications to Wireless Communications," *Proc. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009. 7th Int. Symp.*, Vol. 1, pp. 1-10, June 2009.

[22] H. Zhang and J.C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Ad Hoc and Sensor Wireless Networks*, Vol. 1, No. 1, pp. 89-124, March 2005.

[23] M. Bhardwaj and A. Chandrakasan, "Upper Bounds on the Lifetime of Wireless Sensor Networks," *Proc. IEEE ICC*, Vol. 3, pp. 785-790, June 2001.