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

Yaling Tao[a], Yongbing Zhang[a,*], Yusheng Ji[b]

[a]*Graduate School of Systems and Information Engineering, University of Tsukuba, Japan, Email: {to.are, ybzhang}@sk.tsukuba.ac.jp*
[b]*Information Systems Architecture Science Research Division, National Institute of Informatics, Japan, Email: kei@nii.ac.jp*

Abstract

Power efficiency and coverage preservation are two important performance metrics for a wireless sensor network. However, there is scarcely any protocol to consider them at the same time. In this paper, we propose a *flow-balanced routing* (FBR) protocol for multi-hop clustered wireless sensor networks that attempts to achieve both power efficiency and coverage preservation. The proposed protocol consists of four algorithms, one each for network clustering, multi-hop backbone construction, flow-balanced transmission, and rerouting. The proposed clustering algorithm groups several sensors into one cluster on the basis of overlapping degrees of sensors. The backbone construction algorithm constructs a novel multi-level backbone, which is not necessarily a tree, using the cluster heads and the sink. Furthermore, the flow-balanced routing algorithm assigns the transferred data over multiple paths from the sensors to the sink in order to equalize the power consumption of sensors. Lastly, the rerouting algorithm reconstructs the network topology only in a place where a head drops out from the backbone due to the head running out of its energy. Two metrics called the *network lifetime* and the *coverage lifetime* are used to evaluate the performance of FBR protocol in comparison with previous ones. The simulation results show that FBR yields both much longer lifetime and better coverage preservation than previous protocols. For example, FBR yields more than twice network lifetime and better coverage preservation than a previous efficient protocol, called the coverage-preserving clustering protocol (CPCP) [18], when the first sensor dies and the network coverage is kept at 100%, respectively.

Keywords: load balancing, clustering, multi-hop, 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.

*Principal corresponding author

Wireless sensor networks (WSNs) are usually composed of a large number of sensors, which are densely and randomly deployed over inaccessible terrains and are utilized in applications such as environment surveillance and security monitoring [1]. In most 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 most critical constraints of WSNs is the power limitation, and therefore it is important to design an energy-aware protocol to prolong the lifetime of a sensor network. Two metrics are usually used to show the lifetime of a sensor network [2, 3]: *network lifetime* and *coverage lifetime*. Those metrics indicate the durations from the beginning instant of the network operation to the instant when a given percentage of sensors die and the ratio of the current coverage by the active sensors to the initial coverage by all the sensors drops below a predefined threshold, respectively. In this paper, we aim to design a new data aggregation protocol that yields longer network lifetime and better coverage preservation.

Techniques such as network clustering, sensor scheduling, and multi-hop transmission are widely used to improve energy efficiency for WSNs [4–18]. Clustering is a technique to group several sensors into a cluster with one as the head and the others as the members. Each member sends data to the head and then the head conveys the aggregated data to the sink. Most previous clustering approaches [4–7] mainly focus on the head selection and the cluster construction, rather than the coverage preservation and the data routing after the cluster formation. Sensor scheduling is a technique used in [18–21] to put some sensors into the sleep mode whose sensing areas are totally covered by other sensors. This technique can be combined into the clustering process, and it usually faces the challenges of network connectivity and coverage preservation. Multi-hop transmission has generally been considered an efficient energy-saving approach for large-scale sensor networks[8–18], and the tree rooted at the sink is the most commonly used multi-hop topology. However, the tree topology has an inherent drawback in that each sensor has only one path to the sink, and therefore the data flow passing through each sensor may be imbalanced, resulting in some sensors running out of their energy quickly. To balance the traffic flows to the sink and equalize the residual energy among the sensors, most previous approaches periodically perform the cluster formation and the network construction. Periodic re-clustering and network reconstruction would shorten the lifetime since the overhead cost for transferring the control messages between the sensors cannot be ignored.

In this paper, we propose a new flow-balanced routing (FBR) protocol to achieve power efficiency and coverage preservation. In contrast with previous protocols, the whole network clustering and construction are performed only once at the beginning of the network operation. The network is reconfigured will be performed locally only in a place where any sensor runs out of its energy. Furthermore, the network topology may not be a tree structure but a multi-level hierarchy, where each sensor may have multiple paths to the sink. In FBR, we first propose a clustering algorithm to determine the cluster formation on the basis of 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. All the parents of a head

should be at the same level which is one higher than the head. A head can transfer its data to the sink through any path via its parents. We propose a flow-balanced routing algorithm that assigns the data flow from a cluster head to the sink to equalize the residual energy of the head's parents. Furthermore, to reconfigure the network topology, we propose a local rerouting algorithm to reconfigure the network topology when any head drops out of the backbone. The main contributions of our work can be summarized as follows.

1. Our proposed protocol takes into account the overlapping degrees of sensors in the clustering decision, resulting in better coverage preservation.
2. By organizing the cluster heads into a hierarchical multi-level backbone, each cluster head may have multiple paths to the sink, and by using our proposed flow-balanced routing algorithm the flow from each head to the sink can be distributed to its parents, resulting in energy balance between sensors.
3. Since the network reconfiguration is performed only when a cluster head runs out of its energy and only in a place where the exhausted head drops out of the backbone, the energy needed for network construction is reduced greatly.

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 our conclusions.

2. Related Work

A number of energy-based data aggregation protocols have been proposed [1, 3]. In this section, we summarize the related works regarding the network clustering, the power scheduling, and the multi-hop transmission, respectively, and say how our protocol differs from them.

Most clustering approaches attempt to achieve the energy efficiency for data aggregation in WSNs [1]. Low-energy adaptive clustering hierarchy (LEACH) [4] is a well-known and simple distributed clustering approach wherein each sensor elects itself as a cluster head with a certain probability, and then the cluster heads act as routers aggregating and transferring sensing data to the sink directly. A centralized version of LEACH, called LEACH-centralized (LEACH-C) [5], was proposed wherein each sensor sends its location information along its residual energy to the sink, and then the sink computes the average energy of all the sensors. Then, the sink decides the incapable nodes whose residual energies are below the average energy. Another extension, called the hybrid energy-efficient distributed clustering (HEED) approach [6], considers the residual energy of each sensor and attempts to obtain a well distribution of the cluster heads in the service area. The computation time of HEED is extremely long since the probability of becoming a head is computed iteratively depending on the residual energy of each sensor. Recently, a coverage-based clustering approach was proposed [18]. The coverage cost of a sensor is defined to be inversely proportion to either the total energy of neighboring sensors or the overlapping redundant degree with neighboring sensors. The basic idea of our proposed clustering approach is similar to that of

CPCP, but our proposed approach only calculate the ratio of the area of each sensor that overlaps with other sensors, resulting in a simpler mechanism.

Power scheduling is a technique to switch some sensors off to save power while keeping the network connectivity to satisfy a given coverage preservation requirement [10, 18–21]. An approach called SPAN [19] was proposed to save power consumption by putting some nodes into sleep mode for ad hoc wireless networks. SPAN attempts to switch off such nodes that do not affect the network connectivity by maintaining the information of the two hop neighbors in real time. However, it is not given how to turn a sleeping node on again if the network connectivity is damaged since some active nodes run out of their energy. Furthermore, as pointed out by Zhang and Hou in [27], the power saving for a wireless ad hoc network and a wireless sensor network is generally different from each other. It may be difficult to apply SPAN directly for a sensor network. A coverage configuration protocol (CCP) [20] tries to achieve a given coverage goal by turning off as many sensors as possible while keeping the network connectivity. In a coverage-aware protocol [18], called CPCP, two kinds of coverage preservation approaches were proposed. One is based on the coverage redundancy defined by the number of sensors at each point, and therefore if the sensing area of a sensor is covered by more sensors, the sensor will have a higher priority to be a cluster head. The other is based on the the coverage energy defined by the total residual energy that can be used to monitor a location, and therefore if there are more total residual energy that can be used to monitor sensing area of a sensor, the sensor will have a higher priority to be a cluster head. The drawback of this protocol is that it may take much time to calculate the two parameters. On the other hand, in our proposed approach, only the sensing area of a sensor and the area where the sensor and its neighboring sensors overlap are needed in computation.

Multi-hop transmission is generally more efficient to reduce power consumption than the single-hop transmission [28]. Multi-hop transmission can be achieved for intra-cluster or inter-cluster data transmission. In the former, members of a cluster can transfer the sensed data to the cluster head through multiple intermediate members [9–11], while in the latter [8, 12], a backbone network is constructed with the cluster heads. Inter-cluster transmission has been widely used in previous research. In an inter-cluster transmission approach, a cluster head sends the aggregated data from its members to the sink via multiple intermediate cluster heads. An example of a multi-hop transmission mechanism can be found in the IEEE 802.15.4 standard [13], wherein a personal area network (PAN) coordinator triggers the formation of a cluster-tree and works as the root of the cluster-tree. It broadcasts a beacon message to its neighboring coordinators. A coordinator receiving the beacon decides whether to join the tree and, if it does, it also broadcasts the beacon to its neighbors. The standard does not give the details of how to determine the route from each coordinator to the root.

Some recursive approaches are used to construct hierarchical clustering networks [15–17]. The distributed hierarchical agglomerative clustering (DHAC) approach [17] is a bottom-up network construction scheme wherein some nearby sensors are first grouped into a cluster and a sensor with the smallest identification number is elected as the head. Then, the neighboring clusters are grouped into a larger cluster also with the smallest identification number as the head. This process is repeated until the cluster size reaches a given threshold. The energy-efficient multi-level clustering (EEMC) approach [16] is a centralized and top-down clustering scheme wherein the network

topology is constructed from the sink. The sink first collects the location and energy information of all the sensors and then determines the heads on the level next to it and the members of each head. Each head then collects the information of its members and determines the heads on the level next to it again. This process is repeated until the number of levels equals the optimal expected value. Some approaches proposed for single hop transmission such as HEED [6] can be extended to construct a multi-level network by using a recursive approach similar to [15]. We implemented the hierarchical version of HEED, named M-HEED, for comparison. The main problems in those recursive approaches are that they converge very slowly as the head selection and cluster formation has to be done recursively in each level.

Since most multi-hop transmission approaches are based on a tree topology, the traffic flow passing through the sensors may be unbalanced [22]. To alleviate the flow imbalance problem, some approaches [24–26] try to find and use alternative tree structures for data transmission. However, they face the problem of how to find and when to use the alternative trees and most importantly they cannot resolve the problem of the flow imbalance. In our initial work [33], we proposed a flow-balanced protocol that constructs the network in multiple levels and in which the network topology may not be a tree structure. Therefore, each head may have multiple paths to the sink and, by balancing the traffic flow on each path, can equalize the energy consumption of each head. Furthermore, we propose a new cluster formation approach that preserves the network coverage in a simple but efficient way. In cluster formation, a sensor with a larger overlapping degree is selected as a cluster head with a higher priority. An efficient scheme is also proposed to reduce the power consumption of a sensor in sleep mode.

3. Network Model

In this paper, we consider only one sink and a set of homogeneous 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 at 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 via multiple intermediate sensors. 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 called the *cluster range* in this paper. Generally, R is larger than or equal to r and the transmission range d is shorter than the distance from a sensor to the sink.

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 the distance less than $2r$ from sensor i are called the *friends* of sensor i, denoted by F_i, and the sensing areas of sensor i's friends may overlap with that of sensor i. Since sensors are densely deployed in the target field, the sensing area of a sensor may commonly overlap with other sensors. The *overlapping degree* of sensor i, denoted by ρ_i, is defined as the ratio of the overlapping area of sensor i with its friends to its whole

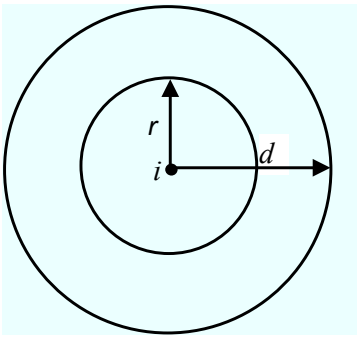


Figure 1: The sensing and the transmission areas of sensor i ($R = 2r, k = 1$).

sensing area as follows.

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

where A_i denotes the sensing area of sensor i and $A_i \cap A_j$ denotes the area sensor i overlaps with its 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 in which sensor i has two friends, j and k, and the area sensor i overlaps with sensors j and k is colored gray. Some methods to calculate the overlapping area of multiple sensors are detailed elsewhere [10, 20, 21, 26, 29**?**].

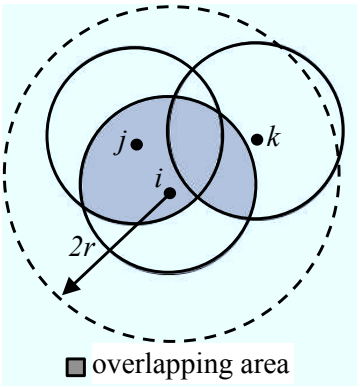


Figure 2: Overlapping area of sensor i with its friends j and k.

The data aggregation is usually run periodically, i.e., once in a regular interval called a round. The cluster formation in most previous approaches [4, 6, 16–18] is performed in each round. However, in our proposed protocol, the cluster formation is performed only once on the basis of the

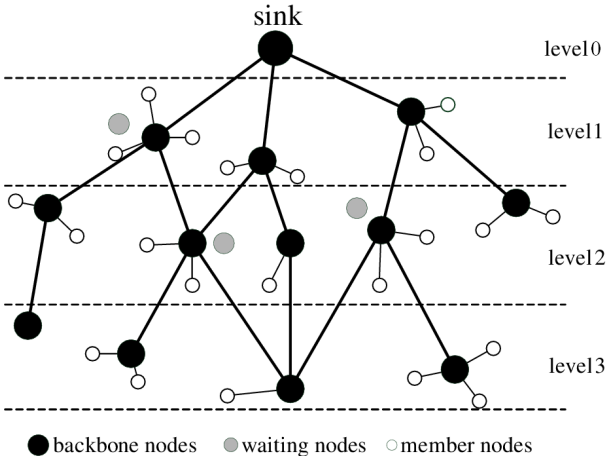


Figure 3: Network model.

overlapping degrees of sensors at the beginning of the network operation. Each sensor tries to become a cluster head in accordance with the value of its overlapping degree i.e., the larger the overlapping degree of a sensor, the higher priority to be a head. A head with $\rho = 1$ is used only for data aggregation and transmission but not for sensing since its coverage area totally overlaps with those of its friends. Furthermore, a sensor with $\rho = 1$ other than a head is put into the sleep mode and called a *waiting node*. A waiting node does nothing but wait for the *HELP* message to replace an exhausted nearby head. If a sensor with $\rho < 1$ receives a HEAD message from one of its friends, it becomes the member of the head.

The cluster heads along with the sink are used to construct a network topology as shown in Figure 3, called the *backbone network*, so that each cluster head can send the aggregated data to the sink. The network topology is not changed unless any cluster-head is dying or loses the connection to the existing network. From Figure 3, we can see that the backbone is not a simple tree but a multi-level hierarchical network in which each node may have multiple parents belonging to the same level. Each node on the backbone can send data to the sink only via its parent(s), and multiple paths may exist from a node to the sink.

4. Proposed Algorithms

Data are aggregated from sensors to the sink in two phases: *route construction* and *data transmission*. In the route construction phase, the sensors are grouped into clusters on the basis of their overlapping degrees, ρ_i, and then a hierarchical backbone is constructed using the cluster heads along with the sink at the top. In the data transmission phase, the sensors send their sensed data to their cluster heads and then the heads forward the data to the sink probably via multiple paths. When a head runs out of its energy or its residual energy becomes lower than a predefined threshold,

it drops out of the backbone and the backbone is reconfigured. For the sake of simplicity, neither the message transmission delay between the sensors nor the computation time at the sensors is taken into account.

4.1. Cluster Formation Algorithm

Unlike previous approaches, our proposed clustering algorithm performs the cluster formation only once at the beginning of network operation so that the overhead for clustering is greatly reduced. Furthermore, a sensor with the largest overlapping degree is selected to be the cluster head to minimize the effect of the death of the sensor. As a result, both the network lifetime and the coverage lifetime can be extended.

At the beginning, the sink broadcasts a $CLS\_FORM(T_0)$ message to inform all the sensors to start the cluster formation, where T_0 is a time limit for all the sensors to finish the cluster formation. After receiving the $CLS\_FORM(T_0)$ message, each sensor sets $D_i = (1-\rho_i)T_0$ and runs Algorithm 1 to determine its own state, i.e., head, waiting node, or member. When the timer t expires, the sensor bids for the head with its neighbors. If there is more than one sensor bidding for the head at the same time, the sensor with the smallest identification number is selected to be the head and it broadcasts a *HEAD* message to its neighbors. If a sensor with $\rho = 1$ receives a *HEAD* message from one of its neighbors, it tries to become a waiting node. Once a sensor becomes a waiting node, it broadcasts a *SLEEP* message to its friends. If a sensor receives a *HEAD* message from a head who is one of its neighbors before its timer expires, it becomes a member of the head. On the other hand, if a sensor receives a *SLEEP* message from a waiting node, it recalculates its overlapping degree without considering the waiting nodes in its friends.

Our cluster formation algorithm is a distributed algorithm. Each sensor, say, sensor i, exchanges its identification number, location information, and state with its neighbors and friends. Therefore, the number of messages transferred between sensor i and its neighbors/friends is $max(O(|N_i|), O(|F_i|))$. In the worst case where all the sensors in a cluster are located in the same position, the overlapping degrees need to be recalculated for $|N_i| - 2$ times, and the computational complexity of our clustering algorithm is $O(|N_i|^2)$. Furthermore, in the worst case where all the sensors in the network are located in the same position, the computational complexity is bound by $O(|S|^2)$. However, in a general case where the sensors are well distributed in the network, we have $|N_i| \ll |S|$ and the number of clusters and the cluster size should be relatively small and the computational complexity is similar to that of previous works like LEACH [4]. Our cluster formation algorithm yields a well cluster distribution similar to the HEED but in a different sense. In HEED, a cluster head is determined on basis of the residual energy levels of the sensors and there should exist one cluster head within the cluster range of a sensor. On the other hand, in our proposed protocol, there exists a cluster head within the cluster range but the cluster head should be the one with the largest overlapping degree among the sensors in the cluster.

4.2. Backbone Construction Algorithm

The backbone is constructed by using the cluster heads along with the sink at the top. The sink initially broadcasts a *BN_CONST* message with a transmission distance of $d = R$ and with

Algorithm 1 Cluster Formation Algorithm.

Initialization

receive *CLS_FORM* from the sink
find friends and neighbors
calculate ρ_i according to eq. (1)
set a timer t for the head determination delay D_i, and then do **State Determination**

State Determination

while $t < D_i$ **do**
  if receive *HEAD* message from neighbor j **then**
    become member of j, and then **exit**
  end if
  if receive *SLEEP* message from friend k **then**
    recalculate ρ_i without considering the waiting friends
  end if
end while
if $t \geq D_i$ **then**
  bid for the head
  if $\rho_i = \rho_k$ $(i < k,\ k \in N_i)$ **or** there is no other bidding sensor **then**
    become the head, broadcast *HEAD* message to neighbors, and then **exit**
  else
    receive *HEAD* from neighbor j
    while $\rho_i = 1$ **do**
      if $i < k$ $(\rho_k = 1, k \in N_i)$ **or** there is no such a neighbor j **then**
        become waiting node, broadcast *SLEEP* message to friends, and then **exit**
      else
        receive *SLEEP* message from friend k and recalculate ρ_i without considering the waiting friends
      end if
    end while
    become member of node j and then **exit**
  end if
end if

a parameter tuple ($k = 1, l = 0, id = sink_{id}$) where k denotes the parameter used for tuning the transmission distance, $d\,(= kR)$, l and id denote the level and the identification number of the message sender, respectively.

When a cluster head, e.g., node i, receives the *BN_CONST* message from its neighbors the first time, it joins the backbone and takes those neighbors as its parents, whose levels are higher than those of others. After joining the backbone, node i updates its level, l, to be one lower than its parent(s). Then, node i broadcasts the *BN_CONST* message with the transmission distance d, and with a parameter tuple ($k = 1, l, id = i$), and then sends the sink an *ON_BN* message to inform the sink of its existence on the backbone. Therefore, the sink knows whether any cluster head is still not on the backbone. If a cluster head is not on the backbone, the sink increments the value of k and then asks the backbone nodes to broadcast the *BN_CONST* message again. The value of k is increased until the *BN_CONST* message reaches a new head that is still not on the backbone. Note that a backbone node may have multiple parents and once it connects to the backbone, its parent(s) is(are) not 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 $k = 1, l = 0, id = sink$, and asked the nodes on the backbone to search for new heads. Node i received the *BN_CONST* message from nodes 2, 3, and 4 (Figure 4(a)) and then determined its parents to be nodes 2 and 3 (Figure 4(b)). Thereafter, node i set $k = 1, l = 2, id = i$, and broadcasted the *BN_CONST* message with $d(= R)$ but no new head could be found. 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).

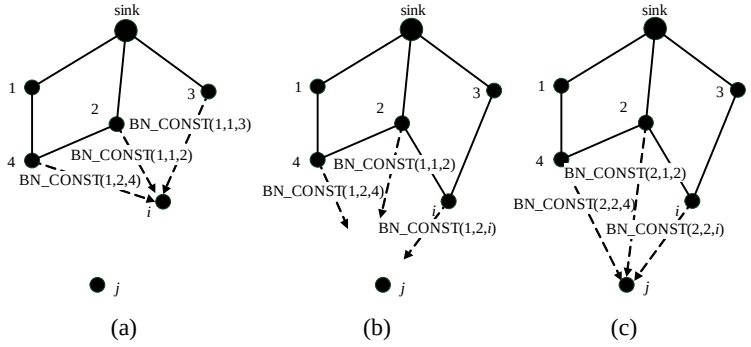


Figure 4: An example of a backbone construction.

In backbone construction, if the *BN_CONST* message cannot reach node i with the current value of k, the sink increments 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 find a neighbor and so k should typically be 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 $k \leq 3$.

Algorithm 2 Backbone Construction Algorithm.

Procedures executed by sink

broadcast $BN\_CONST(k = 1, l = 0, sink_{id})$ with $d\ (= kR)$
if not receive the ON_BN message from all the cluster heads **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**
    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

4.3. Flow-Balanced Routing Algorithm

Each backbone node, i.e., cluster head, collects the sensed data from its members and then conveys the collected data to the sink. A backbone node may have multiple parents and therefore may have multiple paths to the sink as shown in Figure 5. Our goal is to balance the residual energy of each backbone node in order to prolong the network lifetime, that is, to equalize the residual energy levels of the parents of a sensor after sending the collected data. For example, assuming that a backbone node, say, node i, had I-bit data to the sink and three parents whose current energy magnitudes were 0.1J, 0.3J, and 0.4J, respectively. After transferring the data, the residual energy magnitudes of the parents were equalized; e.g., they would become 0.1J, 0.2J, and 0.2J, respectively.

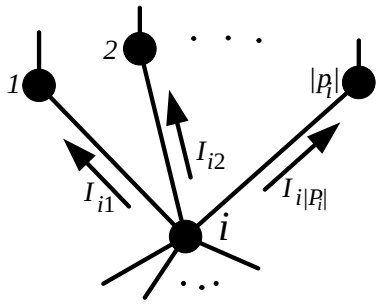


Figure 5: Multiple paths from node i to the sink.

The energy needed to send one bit data from one sensor to another can be calculated by using

the $1/s^n$ path loss model [30] 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, and typically n takes a value between 2 and 5. Similar to Heizelman et al. [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

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

Here, we attempt to let the following relation hold.

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

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 Data Aggregation 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: send flow I_{ij} to parent j ($j \in P_i$) that satisfy eq. (9)

In flow-balanced routing, attempts are made to equalize the residual energy of node i's parents 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. First, node i calculates the total energy needed to convey data I_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 energy equal to or more 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 is triggered.

4.4. Rerouting Algorithm

Instead of reconstructing the whole backbone in each round, we propose a local rerouting algorithm to do the backbone reconfiguration only if the topological change occurs in any place; i.e., if 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, e.g., a given percentage, denoted by r_{th}, of sensor's initial energy, the node's energy is exhausted and the rerouting algorithm is triggered. The

exhausted head tries to find a new head in its neighbors to replace itself. Otherwise, its descendants including the members in its cluster and the children on the backbone lose the connection to the sink, and have to try to repair the connection to the network by themselves. Our proposed rerouting algorithm contains 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 its neighbors to find a capable node to replace itself. A waiting node has a higher priority to become the new head. To detect the HELP message, the waiting node may keep the radio receiver on in each round. Considering that overhearing is not energy efficient [31], we can borrow the beacon scheduling way from [32] by setting the radio receiver on only at the beginning of each round to save energy. In this paper, we also propose a new scheme for the waiting node to detect HELP message. When the data aggregation begins, a waiting node checks the residual energy of each head in its neighbors at the beginning of the first and the second rounds to estimate the remaining lifetime of the head, denoted by t_r, i.e., $t_r \approx \lfloor \frac{E_2 - E_1 r_{th}}{E_1 - E_2} \rfloor$ where E_1 and E_2 denote the residual energy at the beginnings of the first and the second rounds, respectively. To prevent a waiting node from oversleeping, we can use a parameter $\beta\,(0 < \beta < 1)$ and determine t_r as follows.

$$t_r = \lfloor \beta \frac{E_2 - E_1 r_{th}}{E_1 - E_2} \rfloor. \tag{10}$$

Since there may be more than one waiting node in the exhausted node's neighbors, the waiting node with the smallest *id* number is selected as the new head. However, if no waiting node can be found, the node that has the most residual energy and the ratio of the residual energy to the initial energy is higher than r_{th} is selected as the new head. After the new head has been determined, the exhausted head, say, node *i*, informs its members and children of the result. If the new head, say, node *j*, is not a backbone node, node *i* sends a $REQ\_HD(k_i, l_i, i)$ message to *j* so that *j* can determine its level and data transmission distance. In the worst case, if node *i* cannot find any candidate node to replace itself, it involuntarily throws away its descendant(s), and just transfers its own sensed data to its parent node(s) until its death. The abandoned members and children become isolated orphans, and they have to find their new head or parent(s) by themselves.

In the *isolated node reconnection* phase, a member of the exhausted head who realizes its cluster head has gone will try to find a new head. If it can successfully find a backbone node in its neighbors, it becomes its member immediately. However, if it cannot find any backbone node, but has more energy than its neighbors, it becomes the new head and sets $k = 1$. Otherwise, it invites the neighbor with the most energy to become the new head, say, node *j*, and set $k_j = 1$.

An isolated head, say, node *j*, broadcasts a $RECON(k_j, l_j, j)$ message to connect to the backbone. If no backbone node responds, it increments k_j and then broadcasts the *RECON* message again until it finds a backbone node. The rerouting algorithm is illustrated in Algorithm 4.

Algorithm 4 Rerouting Algorithm.

Head Replacement executed by exhausted head i

broadcast *HELP* message to neighbors
if receive response(s) from neighbor(s) **then**
  select the best node, denoted by j, as new head
  if node j is not a backbone node **then**
    send $REQ\_HD(k_i, l_i, i)$ message to node j
  end if
  become a member of node j, and inform members and children to connect to node j
else
  ask members and children do **Reconnection**
end if

Reconnection executed by an isolated member i

if find a backbone node j in neighbors **then**
  become member of node j
else
  if find a node, j ($j = \arg\max(E_k), k \in N_i, E_k/E_0 > r_{th}, E_k > E_i$) **then**
    send $REQ\_HD(k_i, l_i, i)$ message to node j, and become member of node j
  else
    become new head and broadcast $RECON(k_i, l_i, i)$ within $d\,(= k_i R)$
    while cannot find a backbone node l within $d\,(= k_i R)$ whose level is higher than l_i **do**
      $k_i = k_i + 1$
    end while
    become child of node l
  end if
end if

Reconnection executed by an isolated head j

broadcast $RECON(k_j, l_i, j)$ within $d\,(= k_j R)$
while cannot find a backbone node l within $d\,(= k_j R)$ whose level is higher than l_j **do**
  $k_j = k_j + 1$
end while
become a child of node l

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 range	R	10m
Transmission tuning parameter	k	≥ 1
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
Control message size	msg	100 bits
Data compression ratio	r_{dc}	30%
Energy threshold ratio	r_{th}	30%
Head percentage (LEACH)	p	5%
Head percentage (HEED)	C_{prob}	5%
Energy threshold (HEED,CPCP)	P_{min}	10^{-4}J
Transmission range(broadcast) (CPCP)	R_{bc}	20m

5. Simulation

We compared the performance of our proposed HFB with those of LEACH, HEED, M-HEED, CPCP, and HEED-FBR using simulation. Two performance metrics, *network lifetime* and *coverage lifetime*, are used for comparison. The network lifetime is defined as the 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 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 a multi-hop hierarchical 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.

Our simulation program was developed using Java. The parameter values used in the simulation experiments are shown in Table 1, and the performance metrics were also examined with various parameter values. A network model with a 150×150 m^2 square area was used. The sensors were randomly deployed in the network but the sink was located at the position (150, 150). All sensors had an initial energy of $0.5J$. We assumed that each sensor was assigned a unique identification number. In practice, a method [11] can be used to assign distinct identification numbers to the sensors. The data from each sensor to the sink were assumed to be 2000 bits, and the sizes of the control messages exchanged between sensors and between a sensor and the sink were assumed to

be all the same and were 100 bits. Time in the experiments was proceeded in rounds similar to previous protocols [4, 6]. At the beginning of each round, the cluster formation is performed in previous protocols but in our proposed protocol it is performed only once at the beginning of the network operation.

5.1. Lifetime Comparison

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 parameter values used in the experiments are shown in Table 1. The simulation experiment was repeated 10 times in order to calculate the confidence intervals of the results. One example of 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, and some nodes have multiple paths to the sink.

Figure 7 compares the network lifetime of our FBR protocol with those of LEACH, HEED, M-HEED, CPCP, and HEED-FBR. The results shown in the figure were obtained as the sample means of 10 experiments with 95% confidence intervals. Since the half widths of the confidence intervals are all less than 2% of the sample means, they are not shown in the figures. We see from this figure that CPCP outperforms other conventional approaches and FBR yields a much longer lifetime than the others. The lifetime of FBR when the first sensor died is near 10 times longer than that of CPCP and is around 5 times longer when 10 percent of the sensors have 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. Similarly, by comparing HEED-FBR and HEED, we see that the flow-balanced routing algorithm plays a key role in data aggregation and that 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 [18]. 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 on the basis of HEED, attempts are made for the cluster heads at each level to be distributed uniformly in the network, resulting in 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.

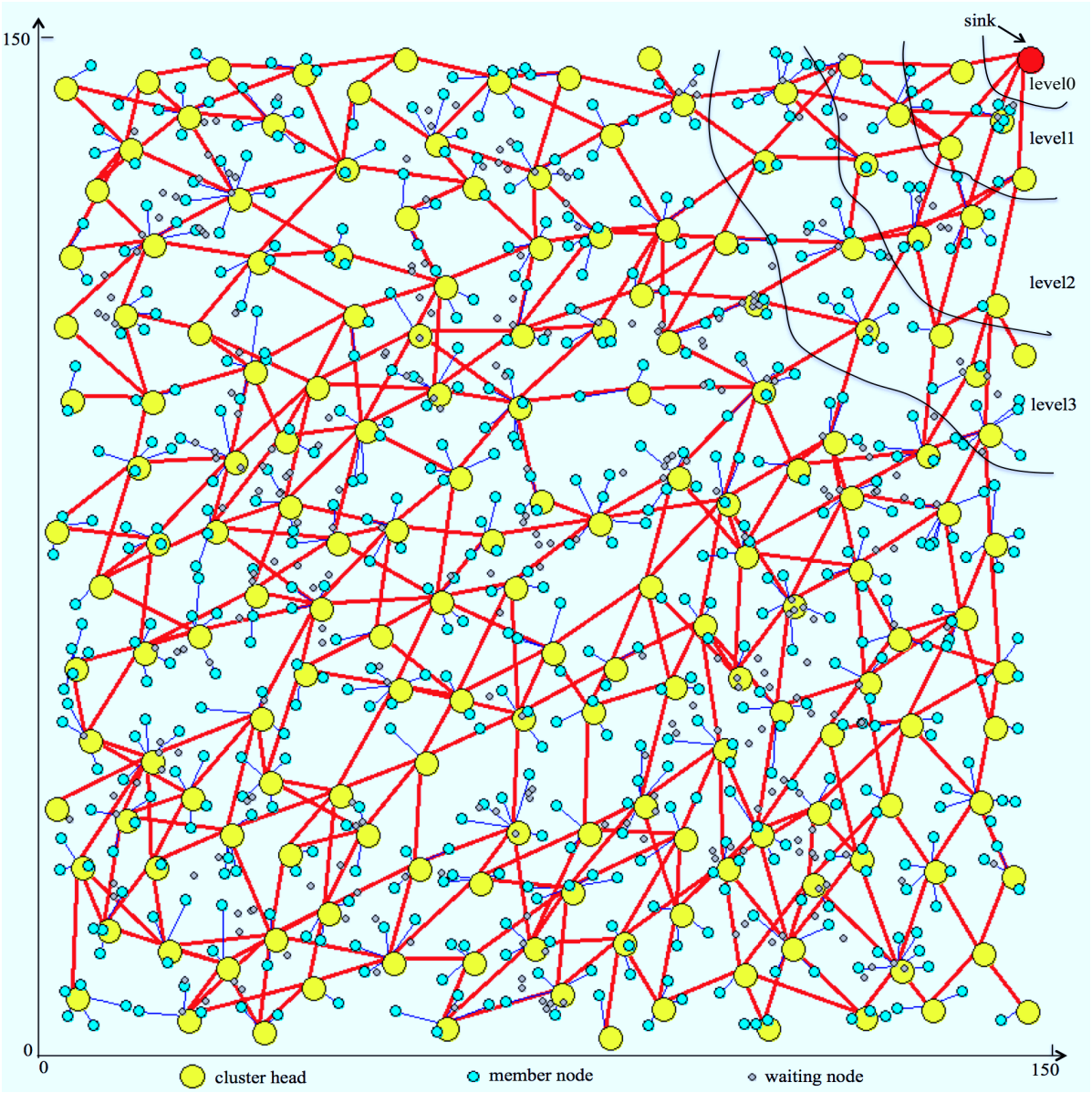


Figure 6: An example of the hierarchical network topology constructed in FBR.

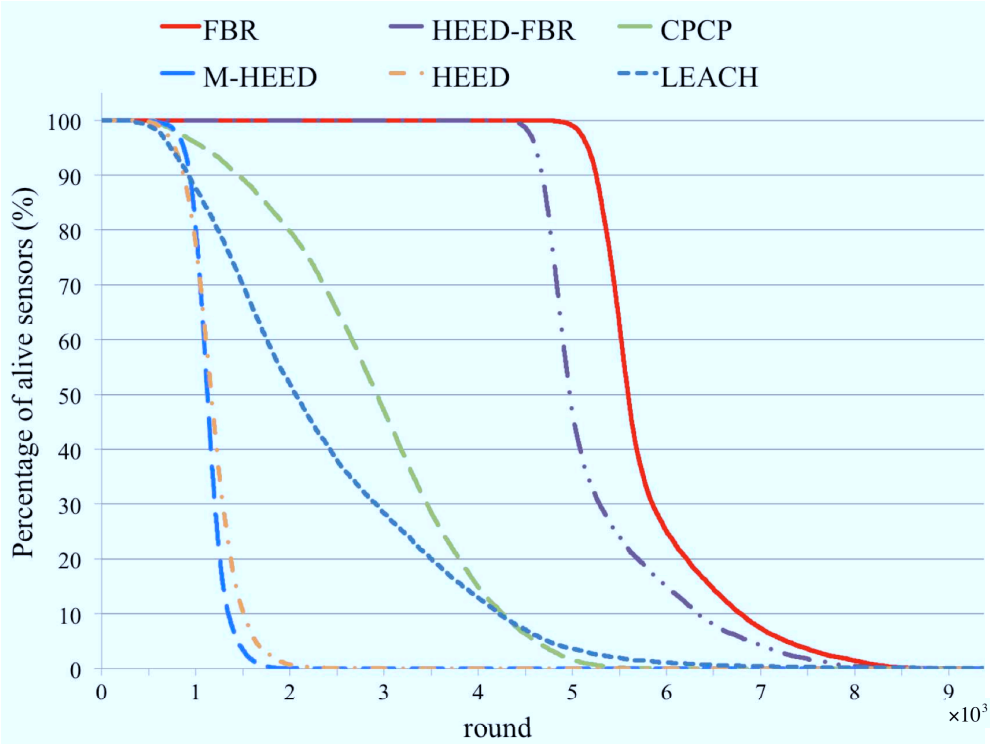


Figure 7: Network lifetimes of various protocols.

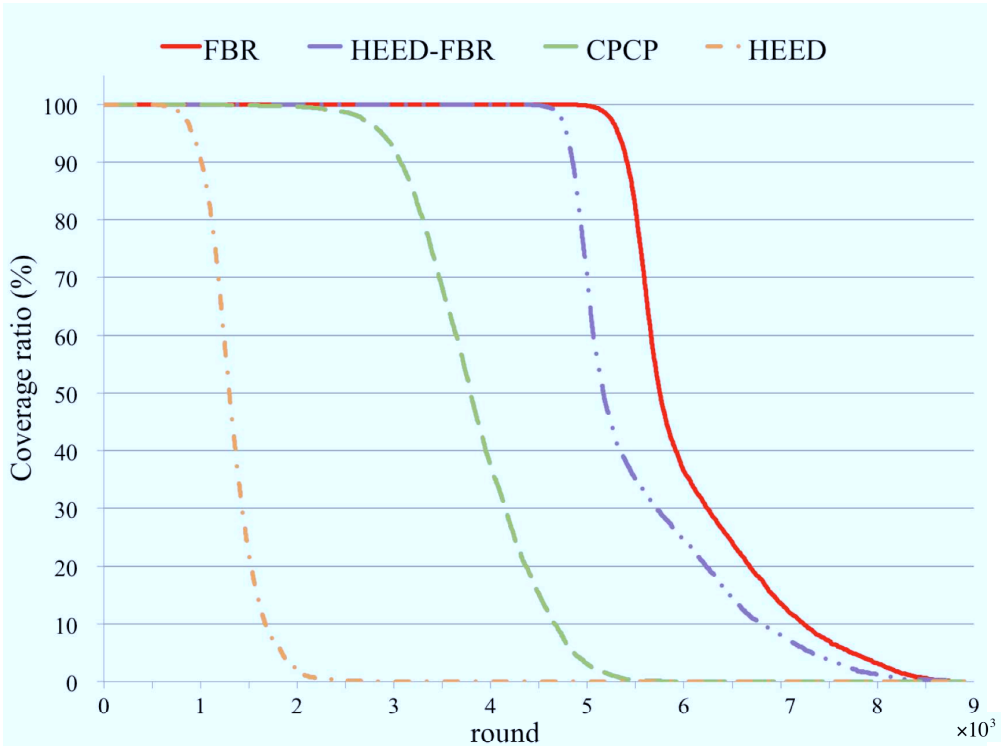


Figure 8: Coverage lifetimes of various protocols.

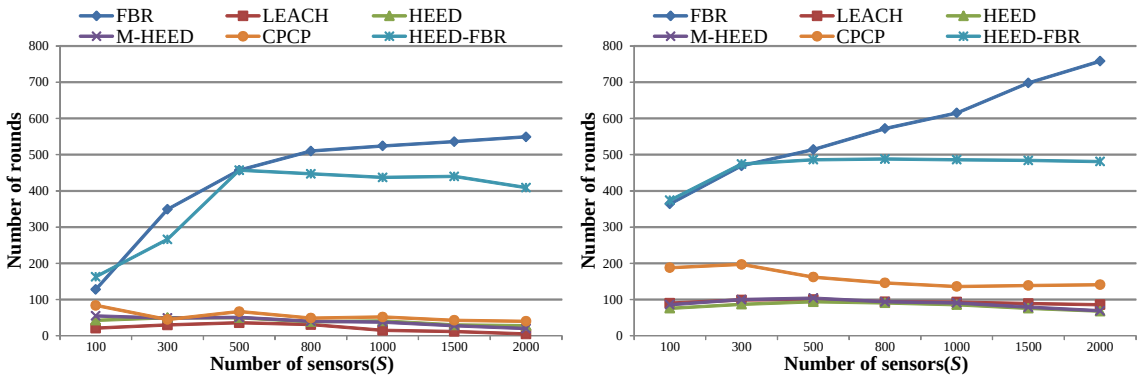
3) In large-scale sensor networks, a large number of sensors are usually deployed randomly in the target field. The coverage areas of some sensors may totally overlap those of others. Taking out the overlapped sensors does not degrade the usability of the network at all. In FBR, the overlapping sensors are taken as *waiting nodes*, that is, those sensors are put into the sleep mode to reduce energy consumption.

5.2. Parameter Examination

To further examine the effects of the system parameters on the performance of our proposed protocol, we simulated FBR and also the main previous protocols with various parameter settings as follows. In these experiments, the parameter being examined is changed while all others are fixed. The initial energy of each sensor was set to 0.05J in order to speed up the experiments, and the others are shown in Table 1. Due to space limitations, the figures show only the network lifetimes of the protocols under consideration when the first sensor or ten percent of the sensors died.

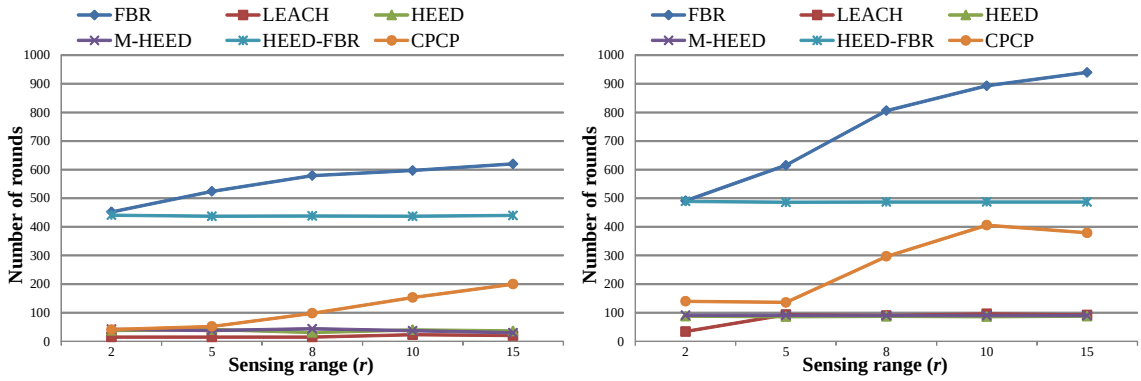
- The numbers of sensors were set to 100, 300, 500, 800, 1000, 1500, and 2000.
- The values of sensing range r were set to 2, 5, 8, 10, and 15.
- The cluster ranges R were set to 5, 10, 15 and 20.
- The energy threshold ratios r_{th} were set to 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.8, and 0.9. This indicates the ratio of the current residual energy of a head to its initial energy and works similarly to parameter p_{min} in HEED and CPCP, but is different in the sense that r_{th} can be adjusted to adapt to various conditions.
- The data compression ratios r_{dc} were set to 0.2, 0.3, 0.4, 0.5, and 0.6. This indicates 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. If a head receives nI-bit data from its children and generates I-bit data to send, then the total data sent to the sink are $r_{dc}(n+1)I$ bits. In the experiments, we also simulated a special case, similar to LEACH, HEED, and CPCP, wherein the data collected at a cluster head are aggregated into one packet no matter how many packets the head receives. The result of this case is shown by η in Figure13.

Figure 9 illustrates the lifetimes of the algorithms under consideration when changing the number of sensors. We see that when the number of sensors increases FBR outperforms others, because more nodes are treated as waiting nodes and the number of paths from each node to the sink may increase, resulting in better flow balancing. Even though CPCP puts some sensors into sleep mode whose sensing areas are totally covered by other sensors, this is decided after the cluster formation phase, but there is no need to consider the sleep nodes in cluster formation. Furthermore, when the number of sensors increases, the network construction overhead in each round also increases, wasting scarce resources, so rerouting locally would obviously work more efficiently than reconstructing the network.



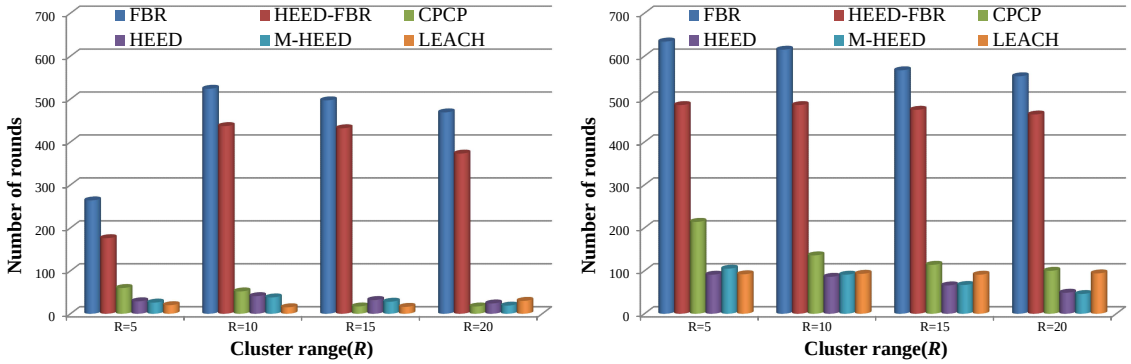
(a) Lifetime when the first sensor died. (b) Lifetime when 10% sensors died.

Figure 9: Lifetimes for various network sizes.



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

Figure 10: Lifetimes for various sensing ranges.



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

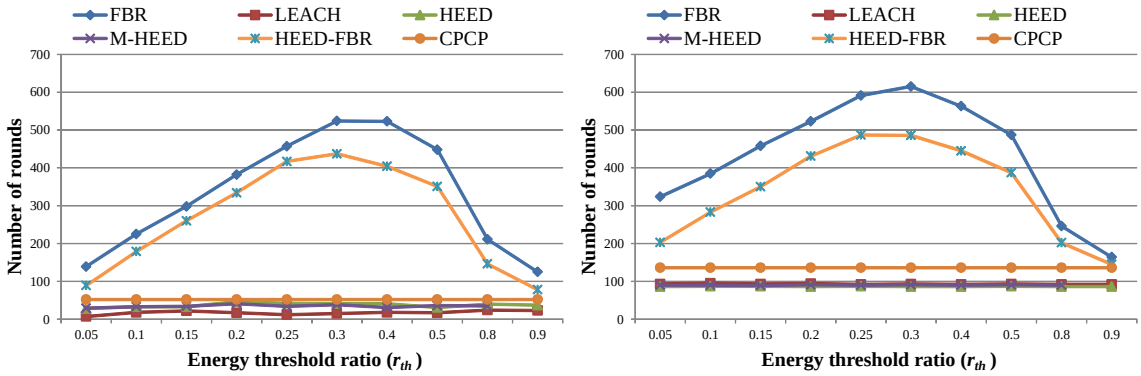
Figure 11: Lifetimes for various cluster ranges.

Figure 10 shows the lifetimes of the algorithms for various sensing ranges. We see that FBR outperforms other protocols as the sensing range increases, because when the coverage area of each sensor increases, the number of nodes that can become waiting nodes also increases. Furthermore, a sensor can find more friends in its widened sensing area and may also choose a better head. Figure 11 shows the lifetimes of the algorithms for different cluster ranges. We see that the cluster range, also used in HEED, M-HEED, and CPCP, affects the performance more because the cluster range determines the size of clusters and the number of network levels.

Figure 12 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 FBR performs best when r_{th} is around 0.25-0.4. Since r_{th} is a tunable parameter, we can adjust its value depending on the network configuration. Figure 13 shows the lifetimes of the algorithms when changing data compression ratios. In this figure, η denotes an extreme case: no matter how many data packets a node receives it will aggregate them into one packet. We see from this figure that FBR outperforms the others and that the network lifetime decreases when the compression ratio becomes low.

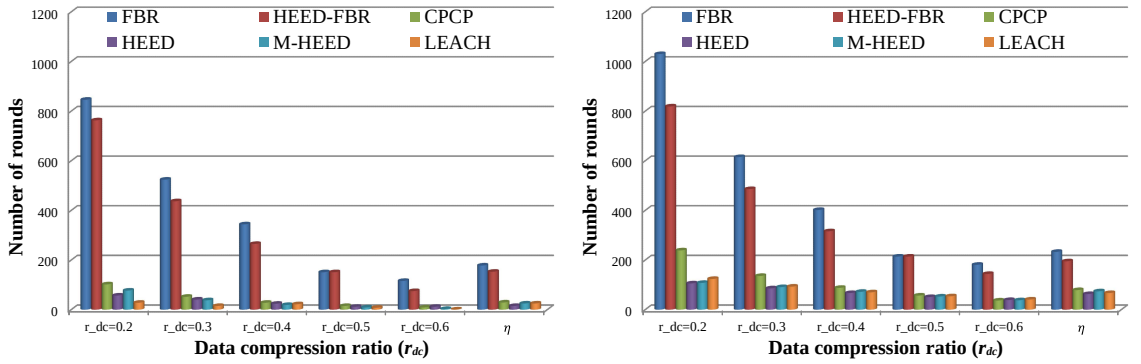
6. Conclusion

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 on the basis of the the overlapping degrees of sensors. Some sensors whose sensing areas are covered by others are put into 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. On the basis of this novel network architecture, a flow-balanced routing algorithm is proposed to assign the flow from a head



(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.

to the sink over multiple paths to equalize the power consumption of sensors. Furthermore, a local rerouting algorithm is proposed to reconfigure the network topology only at the 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 five times longer than that of CPCP, the best among the previous protocols under consideration, and at the same time the coverage lifetime can be two times longer. Furthermore, the effects of the parameters have been examined with a wide range of parameter settings, and FBR always outperforms the others.

Acknowledgements

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] D. Kumar, T.C. Aseri and R.B. Patel, "EECDA: Energy Efficient Clustering and Data Aggregation Protocol for Heterogeneous Wireless Sensor Networks," *Int. J. of Computers, Communications and Control*, Vol. VI, pp. 113-124, March 2011.

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

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

[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] Z. Liu, Q. Zheng, L. Xue and X. Guan, ”A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks,” *Future Generation Computer Systems*, Vol. 28, pp. 780-790, May 2012.

[12] 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.

[13] ”Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs),” *IEEE*, IEEE Standard, 802.15.4-2006, 2006.

[14] K.C. Huang, Y.S. Yen, and H.C. Chao, ”Tree-Clustered Data Gathering Protocol (TCDGP) for Wireless Sensor Networks,” *Proc. Int. Congo FGCN’07*, Vol. 02, pp. 31-36, 2007.

[15] 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.

[16] Y. Jin, L. Wang, Y. Kim, and X. Yang, ”EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks,” *Computer Networks*, Vol. 52, pp. 542-562, February 2008.

[17] C.H. Lung and C. Zhou, ”Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach,” *Ad Hoc networks*, Vol. 8, pp. 328-344, May 2010.

[18] 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.

[19] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, ”Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks,” *ACM/Kluver Wireless Networks (WINET)*, Vol. 8, pp. 481-494, September 2002.

[20] 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.

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

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Ad Hoc Sensor Wireless Networks*, Vol. 1, pp. 89-124, March 2005.

[28] R. Panta, S. Bagchi, I. Khalil, and L. Montestruque, "Single versus multi-hop wireless reprogramming in sensor networks," *Proc. 4th Int. Conf. TridentCom*, pp. 1-7, February 2008.

[29] 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.

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

[31] P. Basu and J. Redi, "Effect of Overhearing Transmissions on Energy Efficiency in Dense Sensor Networks," *Proc. 3rd Int. Sym. IPSN '04*, pp. 196-204, April 2004.

[32] F. Chen, X. Tong, E. Ngai, and F. Dressler, "Mode Switch - Adaptive Use of Delay-Sensitive Or Energy-Aware Communication in IEEE 802.15.4-Based Networks," *Proc. 7th IEEE Int. Conf. MASS*, pp. 302-311, November 2010.

[33] Y. Tao and Y. Zhang, "Hierarchical flow balancing protocol for data aggregation in wireless sensor networks," *Proc. ComComAp*, pp. 7-12, January 2012.

Yaling Tao received the B.S. in computer science and technology from Harbin Engineering University in 2006 and is currently working toward the master's degree at the Graduate School of Systems and Information Engineering, University of Tsukuba, Japan. Her research interests include wireless sensor networks and mobile cloud computing.

Yongbing Zhang received B.S. in Electrical Engineering from Polytechnic University in 1984, and M.S. and Ph. D. degrees in Computer Science both from the University of Electro-Communications in 1989 and 1992, respectively. During 1992-1996, he was a Research Associate at the Department of Computer Science, University of Electro-Communications. He joined the Institute of Policy and Planning Sciences, University of Tsukuba as an Assistant Professor in 1996 and is now a Professor with Graduate School of Systems and Information Engineering, University of Tsukuba. His research interests include distributed/parallel computer systems, communication networks, and performance evaluation.

***Author Biography**

Yusheng Ji received B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo. She joined the National Center for Science Information Systems, Japan (NACSIS) in 1990. Currently, she is a Professor at the National Institute of Informatics, Japan (NII), and the Graduate University for Advanced Studies (SOKENDAI). She is also appointed as a Visiting Professor at the University of Science and Technology of China (USTC). Her research interests include network architecture, resource management, and performance analysis for quality of service provisioning in wired and wireless communication networks. She is a member of IEEE, IEICE, and IPSJ.

yalingtao.eps

yongbingzhang.eps

YushengJi.eps