

Research Article

A New Energy-Efficient Cluster-Based Routing Protocol Using a Representative Path in Wireless Sensor Networks

Hyunjo Lee, Miyoung Jang, and Jae-Woo Chang

Department of Computer Engineering, Chonbuk National University, Jeonju-si, Jeollabuk-do 561-756, Republic of Korea

Correspondence should be addressed to Jae-Woo Chang; jwchang@jbnu.ac.kr

Received 29 December 2013; Revised 28 May 2014; Accepted 3 June 2014

Academic Editor: Thomas Wook Choi

Copyright © 2014 Hyunjo Lee et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks (WSNs) have been broadly studied with advances in ubiquitous computing environment. Because the resource of a sensor node is limited, it is important to use energy-efficient routing protocol in WSNs. The cluster-based routing is an efficient way to reduce energy consumption by decreasing the number of transmitted messages to the sink node. LEACH is the most popular cluster-based routing protocol, which provides an adaptive cluster generation and cluster header rotation. However, its communication range is limited since it assumes a direct communication between sensor nodes and a sink node. To resolve this problem, we propose a new energy-efficient cluster-based routing protocol, which adopts a centralized clustering approach to select cluster headers by generating a representative path. To support reliable data communication, we propose a multihop routing protocol that allows both intra- and intercluster communications. Based on a message success rate and a representative path, the sensor nodes are uniformly distributed in clusters so that the lifetime of network can be prolonged. Through performance analysis, we show that our energy-efficient routing protocol outperforms the existing protocols up to 2 times, in terms of the distribution of cluster members, the energy consumption, and the reliability of a sensor network.

1. Introduction

Wireless sensor networks (WSNs) have been broadly studied in ubiquitous computing environment because of its widespread utilization. The application area of WSNs includes environmental management, health-care services, and military monitoring [1–3]. WSNs are composed of many sensor nodes equipped with processors, memory, and short-range wireless communication. In real applications, the sensor nodes are distributed in the areas of interest, and they sense data from surrounding environments. The sensor nodes cooperate with each other to transmit the sensed data to the central base station, called sink node. A routing protocol is a way of determining a path between a source node and a destination (i.e., sink node) for sensed data transmission. The efficiency of WSNs is highly dependent on routing protocols that directly affect the network lifetime. The main objective of routing protocols is to enhance both reliability and lifetime of WSNs by considering the capability of a sensor node with resource constraints, such as limited power, slow processor,

and low communication bandwidth. Hence, the challenging issue of routing protocols is to reduce the communication overhead for data transmission by determining an optimal path.

Clustering is one of the most popular techniques for routing protocols. The cluster-based routing is an efficient way to reduce energy consumption within a cluster by decreasing the number of transmitted messages to the sink node. Hence, there have been many researches on cluster-based routing protocols [4–9]. A popular cluster-based protocol, called LEACH [4], proposes a two-phase operation based on a single-tier network using clusters. LEACH randomly selects a portion of nodes as cluster headers, and the cluster headers gather the neighboring nodes to construct clusters. Each node forwards its sensed data to a cluster header, which collects and delivers data to the sink node. There are several extensions of the LEACH protocol to increase energy efficiency, but the existing protocols have some problems. First, they assume that all sensor nodes can transmit data to the sink node with enough power and network capability. However, this

assumption is not applicable to sensor networks deployed in large regions. In real world applications, the obstacles (i.e., wall) should be considered to provide reliable network communication. Secondly, the random selection of a cluster header causes the skewed distribution of clusters. In general, because the randomly selected cluster headers are not uniformly distributed over the network, some nodes will fail to find a cluster header within their communication range for WSNs.

In this paper, we propose a new energy-efficient cluster-based routing protocol, which adopts a centralized clustering approach to select cluster headers by generating a representative path. Therefore, the burden of network configuration and routing from sensor nodes can be greatly reduced. By using a representative path, the sink node selects cluster headers and generates clusters in a distributed manner. First, we use message success rate for a representative path creation. In initializing phase, a sink node sends flooding messages to gather each sensor node's information. Upon receiving flood message, a sensor node searches for neighboring nodes in its vicinity and sends advertisement messages to the neighbors. Each node receiving the advertisement messages sends back an acknowledge signal. The message success rate of a sensor node is calculated using the number of messages received from the other sensor nodes. The nodes with high message success rate are likely to be included in a representative path. To support reliable data communication, we devise a multihop routing protocol that allows both intra- and intercluster communications. Finally, we propose both cluster split and merge algorithms to maintain the efficiency of network.

The rest of the paper is organized as follows. In Section 2, we present related work. In Section 3, we propose energy-efficient routing protocol in WSNs. Section 4 provides the performance analysis of the proposed approach, in terms of power consumption and communication reliability. Section 5 concludes this paper with final remarks and future directions.

2. Related Work

Many researches have proposed different protocols for energy-efficient routing in WSNs. In general, the routing protocols for WSNs can be divided into flat-based routing [10–13], cluster-based (hierarchical-based) routing [4–9], and location-based routing [14–17], depending on the network structure. In flat-based routing, all nodes are typically assigned equal roles or functionality. In cluster-based routing, however, nodes play different roles in the network. In location-based routing, sensor nodes' positions are exploited to route data in the network. Among the existing protocols, the cluster-based routing protocol is particularly more suitable for continuous data transmission in WSNs. In this section, we present an abbreviated overview of the well-known cluster-based routing protocols for WSNs, along with their limitations.

Heinzelman et al. [4] introduced a hierarchical clustering algorithm for sensor networks, called low energy adaptive clustering hierarchy (LEACH). The idea is to randomly select

a few sensor nodes as cluster headers and rotate this role to evenly distribute the energy load among the sensors in the network. The LEACH protocol includes two stages of operation: node clustering and information transmission. First, in the clustering stage (Figures 1(a) and 1(b)), a fraction p of all sensor nodes are chosen to serve as cluster headers. A node that produces the random number being smaller than threshold is selected as cluster header. The other nodes are allocated to the cluster header closest to them. Second, in the information transmission stage (Figures 1(c) and 1(d)), the cluster headers aggregate the data received from their cluster members and send the aggregated data to the base station by single hop communication. LEACH outperforms traditional clustering algorithms by using adaptive clusters and rotating cluster header, which can distribute energy consumption among all the sensor nodes. In addition, LEACH can perform local computation so that the amount of transmitted data can be reduced. However, LEACH assumes direct communication between a node and a base station. This is a high-power operation and shortens the lifetime of the network. Moreover, the random selection of headers does not guarantee optimal cluster construction and may cause rounds of communication when cluster headers are not available.

Heinzelman et al. [4] presented LEACH-C (low energy adaptive clustering hierarchy-centralized) in order to distribute cluster headers evenly over the network and reduce energy dissipation. During the initial stage, each node sends to the sink node information about its current location and energy level. Therefore, sensor nodes whose remaining energy is below the average energy are excluded from becoming a cluster header. For each round, the sink node runs an optimization algorithm to determine cluster headers and to divide the network into clusters. Because LEACH-C requires the position of each node at the beginning of each round, an expensive global positioning system (GPS) is required for sending the position information. In addition, the number of nodes for each cluster is not guaranteed when forming clusters.

Farooq et al. [5] proposed a multihop routing with low energy adaptive clustering hierarchy (MR-LEACH). MR-LEACH partitions the network into different layers of clusters, based on the distance between a sensor node and a sink node. Cluster headers are chosen by the LEACH protocol and transmit the aggregated data to a sink node by using multihop routing. Therefore, it achieves significant improvement on energy consumption, compared with the LEACH protocol. The problem of MR-LEACH is that the selection of a cluster header in a layer solely depends on the energy residue of a sensor node, without considering distances among cluster headers. Figure 2 depicts an example of MR-LEACH where each cluster is generated in (a) and (b) by the multihop communication from a cluster header to a sink node.

Distance-energy cluster structure algorithm (DECSA) [6] was proposed by considering both the distance and residual energy of nodes. Resulting from multihop communication between the base station and cluster heads, DECSA reduces the energy consumption of the cluster head. Javaid et al. [7] proposed Enhanced Developed Distributed Energy-Efficient

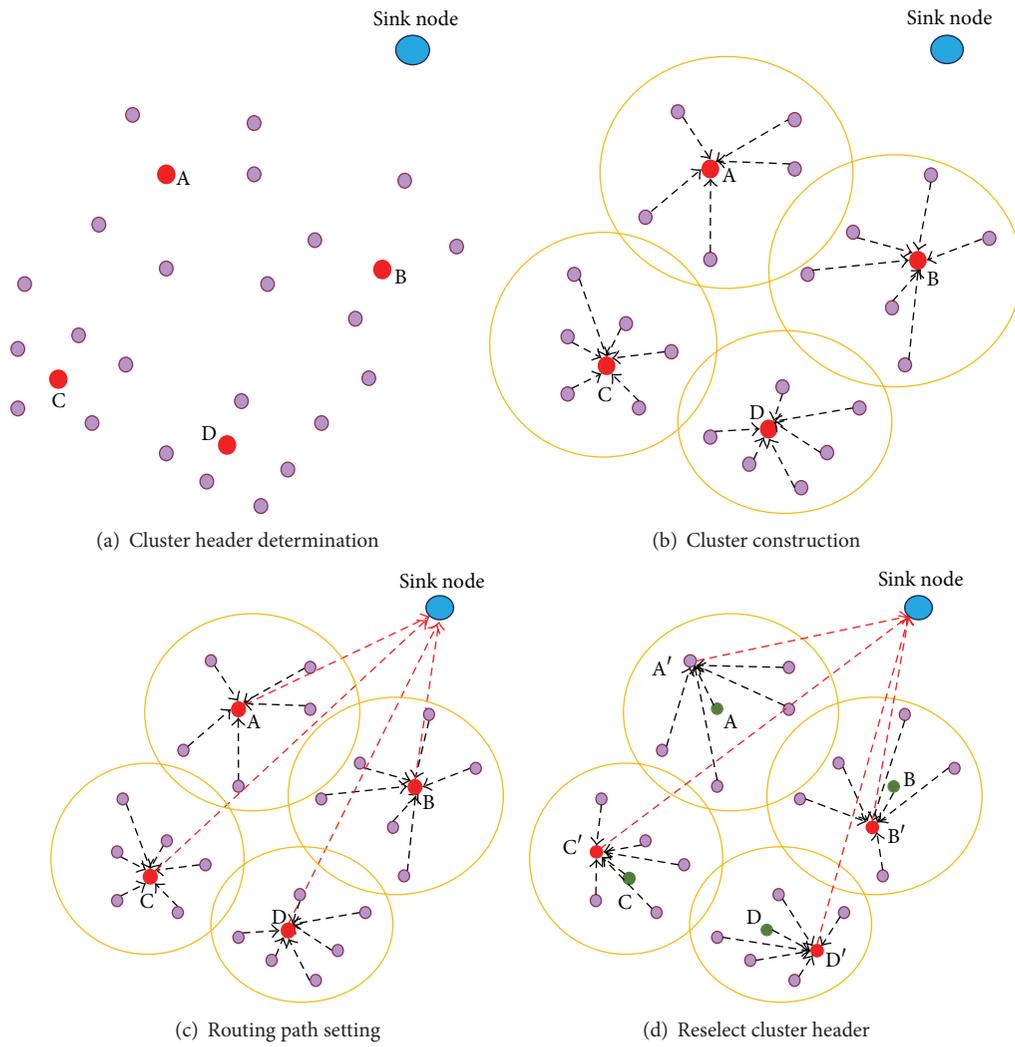


FIGURE 1: LEACH protocol.

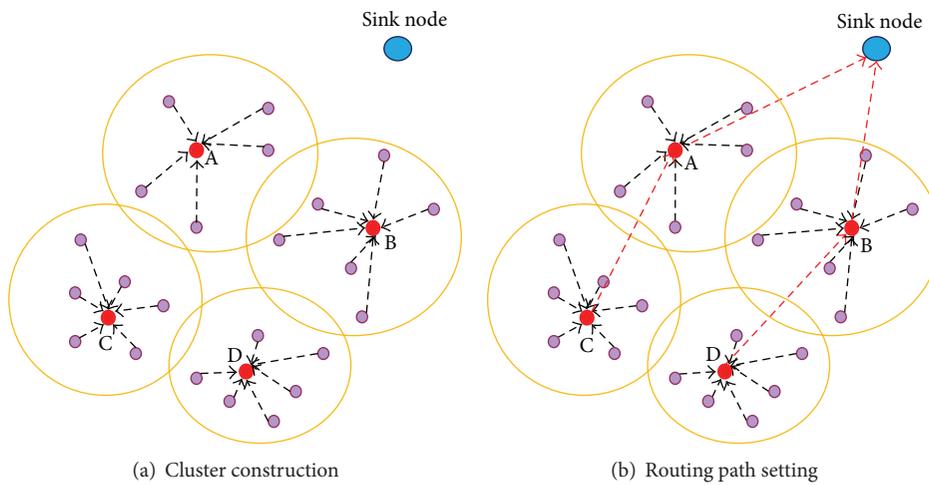


FIGURE 2: MR-LEACH protocol.

Clustering (EDDEEC), which selects a cluster header based on energy level of the nodes. To distribute the equal amount of energy between sensor nodes, EDDEEC dynamically changes the probability of nodes to be chosen as a cluster header in a balanced way. Jorio et al. [8] proposed K-Way spectral clustering protocol, which uses the principle of spectral clustering. The K-Way clustering approach groups sensor nodes into K disjoint classes based on the Laplacian matrix and its eigenvectors. In this way, it calculates a data transmission path to the base station in an energy-saving manner. Nikolidakis et al. [9] proposed Equalized Cluster Head Election Routing Protocol (ECHERP) that aims at energy conservation through balanced clustering. In order to prolong the network lifetime, ECHERP models a network as a linear system by using the Gaussian elimination algorithm and selects the good combinations of cluster heads for minimizing the energy consumption for data transmission.

3. Energy-Efficient Cluster-Based Routing Protocol

The major problem of LEACH [4] is that it selects cluster headers using the probabilistic approach and so it cannot consider a node's residual energy and its location to the sink node. Moreover, because LEACH assumes a direct connection between a cluster header and a sink node, it limits the size of sensor networks. This kind of approach is not appropriate for an energy-efficient routing protocol in WSNs. To support both energy-efficient routing and a wide range of network connectivity, an algorithm is required to consider a node's connectivity with neighboring nodes and its distance to the sink node.

We propose an energy-efficient cluster-based routing protocol using a representative path. To generate a representative path, we adopt a centralized clustering approach by using the message success rate of a sensor node. By using a representative path, a sink node selects cluster headers and generates clusters in a distributed manner. Therefore, the burden of network configuration and routing from sensor nodes can be greatly reduced. Ideally, nodes with high connectivity should become a cluster header to increase the lifetime of network. Our energy-efficient routing protocol consists of four phases: network information table generation, representative paths construction, cluster generation, and cluster management.

3.1. Network Information Table Generation. In this phase, the network information table of every node is generated by using a message flooding technique [18]. First, a sink node broadcasts flooding messages periodically based on the duration of message flooding and the maximum number of flooding data packets. If a node receives the flooding message, it updates its node information into the sink node, as shown in Table I. We use a message success rate (MSR) to guarantee the reliability of data communication. For initial data transmission to the sink node, a sensor node selects a parent node as one having the highest MSR among its neighboring nodes. Hence, our protocol can reduce data loss caused by the limited communication range of sensor nodes. Equation (1)

TABLE I: Node information for a sensor node.

Information	Description
NodeID	ID of a node
HopCount	Minimum number of links from a node to a sink node
NeighborCount	Number of neighbor nodes
Neighbor nodes (ID, MegSuccRate)	Information of neighbor nodes including ID and MSR
ChildnodeCount	Number of child nodes
Child nodes (ID, MegSuccRate)	Information of child nodes including ID and MSR
InitialParentnodeID	ID of parent node selected initially

shows how the message success rate is calculated where i is a node ID, t is a message flooding duration, $PacketsExpected$ means the maximum number of flooding data packets in t , and $PacketsReceived$ means the number of messages from the node i in t . Here, $PacketsExpected$ is calculated by dividing the message duration t with a predefined time interval between queries. For calculating $PacketsReceived$, the algorithm counts the number of received messages during flooding messages as follows:

$$\begin{aligned} & \text{Average Success Rate } (i) \\ &= \frac{PacketsReceived(i \cdot t)}{\max(PacketsExpected(i, t), PacketsReceived(i, t))}. \end{aligned} \quad (1)$$

Secondly, the initial network is formed to forward each node's information to the sink node. A node selects its parent node as one which has the smallest hop count and the highest MSR among its neighboring nodes. Then, the node sends its network information to the sink node through the parent node. Based on the received information, the sink node generates the whole network information table. Figure 3 shows an example of a sensor network with node information where a dotted edge connecting two nodes shows its message success rates (MSR). Each sensor node selects its parent node using the MSR in the initial stage. For example, since the neighboring nodes of node 15 are nodes 11, 14, 18, and 19 and node 11 has smaller hop count than its own, node 15 selects node 11 as a parent node.

Algorithm 1 shows an algorithm for the network information table generation. The algorithm consists of two parts: a sensor node part (lines 1–6) and a sink node part (lines 7–12). In the sensor node part, each sensor node updates its node information by exchanging flooding messages (lines 1–4). When message flooding is over, all sensor nodes send their node information to the sink node through the parent nodes (line 5). On the other hand, the sink node periodically broadcasts a flooding message for maintaining its network information table up to date (lines 9–12).

3.2. Representative Paths Construction. In this phase, the sink node constructs representative paths (RPs). The representative path means a set of nodes, which have been selected from

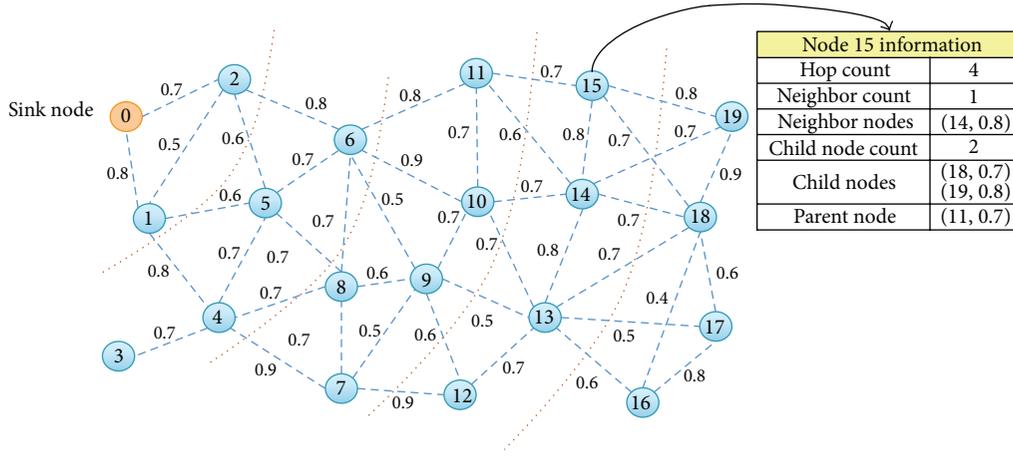


FIGURE 3: Example of sensor network with node information.

Network information generation algorithm **T_f : Time duration for flooding messages****numMsg_f: Maximum number of messages in T_f**

```

(1) Sensor Node(Message msg){
(2)   if (expiredTime <  $T_f$ ) then
(3)     if (curHopCnt > msg.HopCnt + 1) then curHopCnt = msg.HopCnt + 1
(4)     UpdateNetInfo and flood message
(5)   else SendNetInfo(parentID, NetInfo(nodeID));
(6) }
(7) Sink Node(Message msg){
(8)   while (expiredTime <  $T_f$ ) FloodingMsg();
(9)   If Receive Message(NetInfo) from Sensor Node then {
(10)    For each entry  $i$  of NetInfo
(11)    NetInfoTable[NetInfo.id][parent[i]] = NetInfo.HopCnt[i]
(12) }

```

ALGORITHM 1: Network information table generation algorithm.

each hop using their connectivity. The selected node from a hop, called an anchor node, has better connectivity than its neighboring nodes. For example, there are three neighboring nodes A, B, and C in the first hop and their connectivities are $A > B > C$. Because node A has the highest connectivity among them, it becomes an anchor node. The definitions of anchor node and representative path are as follows.

Definition 1 (anchor node and representative path). There are k number of neighboring nodes in the h th hop ($1 \leq h \leq m$) = $\{N_{h1}, N_{h2}, \dots, N_{hk}\}$:

- (i) anchor node (AN_1) in the 1st hop = $\{N_{1i} \mid \forall j, \text{Connectivity}(N_{1i}) \geq \text{Connectivity}(N_{1j}), \text{ where } i, j \text{ are the integer, } i \neq j, i \leq k, \text{ and } j \leq k\}$;
- (ii) AN_h in the h th hop ($h > 1$) = $\{N_{hi} \mid N_{hi} \text{ and } N_{hj} \text{ are the child nodes of the anchor node } AN_{h-1}, \forall j, \text{Connectivity}(N_{hi}) \geq \text{Connectivity}(N_{hj}), \text{ where } i, j \text{ are the integer, } i \neq j, i \leq k, \text{ and } j \leq k\}$;
- (iii) representative path (RP) = $\{AN_1, AN_2, \dots, AN_m\}$.

To construct a representative path, our protocol measures the connectivity of nodes by using (2). When a node has the higher connectivity, it has the higher reliability of communication since it has the higher message success rate than its neighboring nodes. In (2), $cnode$ means a current node, $node(i)$ means a neighboring node i of the current node, $MsgSuccRate$ means a message success rate (MSR) between the current node and $node(i)$, $NeighborMsgSuccRate$ means the MSR between $node(i)$ and $node(j)$, and $ChildnodeCount$ means the number of child nodes of $node(i)$. In addition, α, β, γ mean weights for three parts of (2), where the sum of α, β , and γ is 1:

$$\begin{aligned}
 & \text{Connectivity}_{cnode, node(i)} \\
 &= \alpha \times \text{MsgSuccRate}(current, node(i)) \\
 &+ \beta \times \sum \left(\text{NeighborMsgSuccRate}_{node(i), node(j)} \right) \\
 &+ \gamma \text{ChildnodeCount}_{node(i)}.
 \end{aligned} \tag{2}$$

First, for the nodes in the first hop, the sink node calculates their connectivity. Secondly, a node with the highest

connectivity value is selected as an anchor node for the first representative path. Thirdly, our protocol computes the MSR of the neighboring nodes of the first anchor node. Based on (2), the higher MSR between two nodes can guarantee the better reliability of network communication. If there is a neighboring node whose connectivity value is less than a threshold, the neighboring node is excluded from an anchor candidate. In the same manner, the sink node selects all anchor nodes for a representative path.

Algorithm 2 shows an algorithm for constructing a representative path. First, the sink node selects the node with the highest connectivity from the neighboring anchors. By expanding hop distance, it generates a representative path with the selected nodes (lines 1–10). When the number of representative paths is more than two, the algorithm checks whether or not the selected node is one of the member nodes of other representative paths. If the node is already selected, the algorithm excludes the node and reselects another node from candidates (lines 12–13). The algorithm is terminated when all the representative paths are constructed.

Figure 4 shows an example of representative path construction. In this example, we set the weights of α , β , and γ to 0.4, 0.3, and 0.3, respectively, and the initial threshold of MSR is given to be 0.6. First, the sink node measures the connectivity of 1-hop nodes, that is, nodes 1 and 2, from the sink node. Based on (2), the connectivity between nodes 0 and 1 is calculated as 1.22 and the connectivity between nodes 0 and 2 is calculated as 1.18. Since the MSR between node 1 and node 2 is 0.5, the sink node computes the total number of representative paths as 2. Between two candidate nodes, node 1 is assigned to the anchor of the first representative path (RP_1) because node 1's connectivity value is greater than that of node 2. By expanding RP_1 using the connectivity of child and neighboring nodes, RP_1 is composed of six nodes, that is, 1, 4, 7, 12, 13, and 18. When constructing the second representative path, there is a problem of selecting the same node as anchor at node 10 and node 14. Node 10 selects node 14 as an anchor node, instead of node 13, because node 13 already belongs to RP_1 . Similarly, node 14 selects node 19 as an anchor node. As a result, RP_2 is composed of five nodes, that is, 2, 6, 10, 14, and 19.

3.3. Cluster Generation. In the cluster generation phase, our protocol selects cluster headers from the anchor nodes of representative paths and broadcasts a cluster-join message to every sensor node. For minimizing the communication cost, it is important to minimize distances from cluster headers to their cluster members. For this, our protocol considers the possible combinations of anchor nodes of representative paths. Based on the existing work [4], the number of cluster headers is recommended to be set to 5% of the total number of sensor nodes in the network. By using both the number of cluster headers and the number of representative paths, the sink node first calculates the possible combinations for selecting header nodes. For example, we assume that the total number of sensor nodes is 60 and there are two representative paths, RP_1 and RP_2 . In this case, since the number of header

nodes should be set to 3, we can choose n and $3-n$ number of anchor nodes from RP_1 and RP_2 , respectively, where $0 \leq n \leq 3$. Thus, the possible combinations can be represented as $\{(3, 0), (2, 1), (1, 2), (0, 3)\}$, where, in (x, y) , x means the number of the selected anchor nodes from RP_1 and y means the number of the selected anchor nodes from RP_2 . If RP_1 and RP_2 have 5 anchor nodes and 4 anchor nodes, respectively, the total number of possible combinations is calculated as follows:

$$\begin{aligned} & \binom{5}{3}C_3 * \binom{4}{0}C_0 + \binom{5}{2}C_2 * \binom{4}{1}C_1 + \binom{5}{1}C_1 * \binom{4}{2}C_2 + \binom{5}{0}C_0 * \binom{4}{3}C_3 \\ & = 10 + 40 + 30 + 4 = 84. \end{aligned} \quad (3)$$

Since the computation of all possible combinations for cluster generation is a time-consuming and energy-inefficient one, we propose an approach to filter out some nodes based on hop distances. A candidate node whose hop distance is less than the average can be excluded from the possible combinations. Our protocol measures the communication cost of each combination. The communication cost is calculated based on the hop distances of the shortest path from a sensor node to its nearest header node and a subpath from the cluster header node to the sink node in a representative path. Then our protocol selects a combination with the least cost.

Algorithm 3 shows a cluster generation algorithm. First, the sink node makes possible combinations by using the network information table (lines 1–3). Secondly, the communication cost for each combination is calculated (lines 6–12). Thirdly, the combination with the least cost is selected among them. Finally, the cluster headers send cluster-join messages to all the sensor nodes (line 4).

Figure 5 shows an example of selecting cluster header nodes. Here we assume that the maximum hop distance from a node to its nearest header is only 2. The dotted edge connecting two nodes shows the connectivity of two nodes. Because node 6 and node 13 have the highest connectivity, we consider the combinations that include nodes 6 and 13. In this time, because the threshold of hop distance is 2, we can filter out the combinations that include the neighboring nodes of nodes 6 and 13. Therefore, the combinations (1, 6, 13), (4, 6, 13), (6, 7, 13), and (6, 13, 19) are left as candidates for cluster headers. Finally, among them, nodes 4, 6, and 13 are selected as cluster headers based on the communication cost and connectivity.

3.4. Cluster Management. Cluster head replacement is required to prolong a network lifetime by evenly distributing energy load among sensor nodes. To achieve this, we provide a technique for periodic header replacement and reconfiguration in a cluster. It is obvious that a cluster header consumes a lot more energy than the member nodes of a cluster. Both LEACH and MR-LEACH incur the skewed distribution of clusters because they do not consider the distance between cluster member nodes. Thus, in a dense network, a cluster head is frequently reselected or even the reconstruction of the whole network is required. Our protocol replaces a cluster header periodically by considering the connectivity of nodes in representative paths and their estimated energy residue. Algorithm 4 shows our cluster

```

Make representative path algorithm
maxHop: maximum hop counter
(1) Sink node(NetInfo){
(2)   If nodeID.HopCnt == 1 then Anchors[] = SetAnchor(NetInfo);
(3)   For each node in Anchors[]{}
(4)     Set initial RPPath(i) with selected nodes;
(5)   }
(6)   For all RPPath(i){
(7)     If (RPPath(i).CurAnchor → hop ≥ maxhop) then break;
(8)     cand = SetCand(i, RPPath(i).CurAnchor);
(9)     MeasureWeight(cand);
(10)  }
(11)  nextAnc = SelectNextAnchor(i, cand);
(12)  If nextAnc exists in RPPath(i) then nextAnc = ReselectNextAnchor(cand);
(13)  RPPath(i).CurAnchor = nextAnc;
(14) }

```

ALGORITHM 2: An algorithm for constructing representative paths.

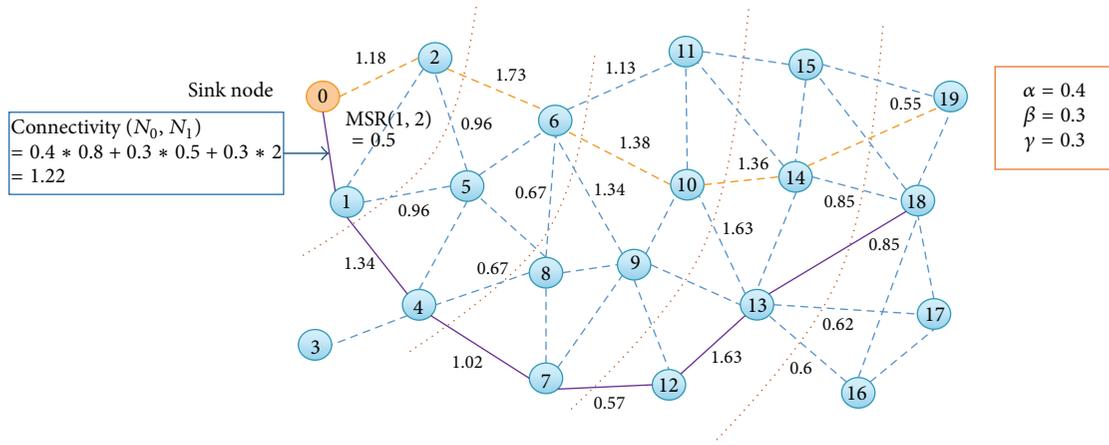


FIGURE 4: An example of constructing representative paths.

management algorithm. The algorithm periodically checks the estimated energy residue of header nodes (lines 2–5). If there is a header node whose energy level is below the given threshold, the header node is replaced by another anchor node in the representative path. Then, a network modification message is sent to all the sensor nodes (lines 6–9).

4. Performance Analysis

4.1. Comparison. We compare our cluster-based routing scheme with the existing routing protocols, such as LEACH, LEACH-C, MR-LEACH, DESCA, EDDEEC, K-Way, and ECHERP, in terms of mobility of node, cluster header selection, cluster header rotation, supporting multilevel hop, and usage of GPS. Table 2 shows the comparison results. Firstly, because DESCA, K-Way, and ECHERP consider the whole network as a static system, they cannot support the mobility of the node. Secondly, only ECHERP does not support the rotation of the cluster headers since it reconstructs all the

clusters when the cluster headers have low energy residue. Thirdly, because MR-LEACH, DESCA, ECHERP, and our protocol can provide a multihop based communication, they can reduce the communication cost by decreasing the distance between the nodes, compared with the other protocols. Finally, even though LEACH-C and K-Way can provide the more efficient cluster optimization algorithm than the other protocols, they cannot be utilized for the typical WSN. This is because they make use of GPS, instead of using the sensors. According to the comparison of routing protocols, it is shown that our protocol is one of the best ones in the typical WSN environment because our protocol not only supports the mobility of the node, but also provides a multihop based communication, without using GPS-equipped sensor nodes.

In our experiment, we compare the performance of our protocol with LEACH and MR-LEACH. We exclude some protocols for the following reasons. First, LEACH-C and K-Way are excluded in our experiment since they utilize GPS-equipped sensor nodes. Secondly, EDDEEC is excluded since it cannot support the multilevel hop based communication. As a competitor, we select LEACH due to its popularity,

```

MakeCluster(NetInfo, *PathList[PathCnt], NumPath){
(1)  For each Path  $p$  from NumPath{
(2)    Comm = Combination(PathCnt, TotalHead, PathCnt, ChooseHead)
(3)    If Comm < MinComm then MinComm = Comm
(4)  } FindNNHead(ChooseHead, NetInfo)
(5) }
(6) Combination(PathCnt, TotalHead,  $q$ ){
(7)  if ( $r == 0$ ) then add_array( $q$ );
(8)  else{
(9)    buf_route[ $r - 1$ ] = route[0][ $n - 1$ ];
(10)   combi( $n - 1, r - 1, q$ );
(11)   combi( $n - 1, r, q$ );
(12) } }

```

ALGORITHM 3: An algorithm of generating cluster.

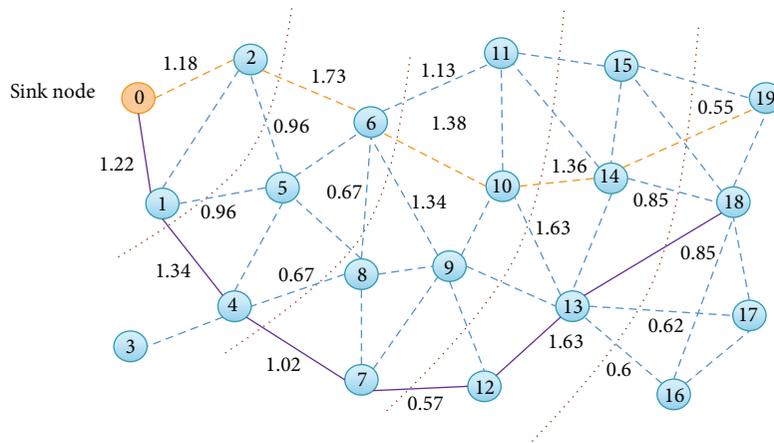


FIGURE 5: An example of selecting cluster headers (three cluster headers: nodes 4, 6, and 13).

even though it cannot support the communication based on multilevel hop. Thirdly, DESCA and ECHERP are excluded in our experiment because it is impossible to add and remove the nodes in DESCA and ECHERP. It is necessary to support the mobility of nodes in the real world environment since communication problems can occur by obstacles, node failures, and so on.

4.2. Experimental Setup. We implemented our scheme with NesC (network embedded system C) and ran our experiment on TOSSIM simulator [19] in TinyOS using an Intel Xeon 3.0 GHz with 2 GB RAM. We set the maximum communication range of sensor nodes to 25 m, which is the maximum communication range of MicaZ. We consider a scenario where every sensor node generates a message and sends it to the sink node. We use three different types of network distribution, that is, uniform, Gaussian, and skewed distribution, where 100 nodes are distributed within 100 m * 100 m network area. Figure 6 shows three types of sensor node distribution.

We experimentally analyze the optimal weights for choosing a cluster header, discussed in Section 3, for the uniform cluster generation. Figure 7 shows the standard deviation of

the number of cluster members for the different weights of α , β , and γ , for instance, 127 meaning $\alpha = 0.1$, $\beta = 0.2$, and $\gamma = 0.7$. Here α is the number of neighboring nodes, β is a message success rate with neighboring nodes, and γ is the number of hops for a sensor node. The total sum of α , β , and γ is 1. As shown in Figure 7, the standard deviation of the number of cluster members is the smallest when α , β , and γ combinations are 3, 2, and 5. So, in this paper, we use the weights of $\alpha = 0.3$, $\beta = 0.2$, and $\gamma = 0.5$ as the optimal weights for choosing a cluster header. In the case when the distribution of nodes is different from that of nodes in our experiment, it is required to set the default values of α , β , and γ . For this, we calculate the averages of the weights being the local optimization values. As a result, we set the default values of α , β , and γ as 0.33, 0.31, and 0.36, respectively.

4.3. Energy Efficiency of Routing Protocols. The primary goal of our scheme is to reduce the communication cost of routing protocol so as to prolong the lifetime of a sensor network. In general, a sensor node has a waiting status, a receiving one, and a transmitting one. Among them, the data transmission status causes the most significant energy loss, since it consumes more than 80% energy of the sensor node.

```

(1) ManageCluster(NetInfo){
(2)   RoundCheck = CurrRound %/Header Update Interval
(3)   If RoundCheck = 0 then {
(4)     Header = ChangeHeader()
(5)   }
(6)   If Energy of Header < Energy of Threshold then {
(7)     Type of Message = ReConstruction of Routing Table
(8)     MakeNetInfo(Message)
(9)   }
(10) }

```

ALGORITHM 4: An algorithm of cluster management.

TABLE 2: Comparison of the protocols.

Protocol	Mobility of nodes	CH selection	CH rotation	Supporting multilevel hop	Usage of GPS
LEACH [4]	Limited	Probability/random	Yes	No	No
LEACH-C [4]	Limited	Probability/random	Yes	No	Yes
MR-LEACH [5]	Limited	Probability/random	Yes	Yes	No
DESCA [6]	No	Distance and energy residue	Yes	Yes	No
EDDEEC [7]	Limited	Random	Yes	No	No
K-Way [8]	No	Random	Yes	No	Yes
ECHERP [9]	No	Random	No	Yes	No
Our protocol	Limited	Connectivity	Yes	Yes	No

Because the amount of transmission energy consumption is influenced by a distance between a sender and a receiver, it is important to have a short communication distance from a sensor node to its parent node in the network hierarchy. In our experiment, we evaluate four measures: the standard deviation of the number of cluster members in a cluster, the average communication range of a sensor node, the lifetime of a sensor node, and the lifetime of the sensor network.

Because imbalanced node distribution incurs a huge amount of energy consumption, we measure the standard deviation of hop distance among the nodes in a cluster. As shown in Figure 8, for all the distribution cases, our scheme has much lower deviation for the number of cluster members than the existing methods. In the uniform network distribution, our scheme achieves 8.07 on deviation whereas LEACH and MR-LEACH have 9.54 and 9.48, respectively. In particular, MR-LEACH generates highly dense clusters in the skewed distribution because the cluster head selection is solely dependent on the energy residue. Our scheme outperforms the existing methods because it performs both cluster splitting and merging algorithms for the optimal cluster construction.

In Figure 9, for random data distribution, the average communication distance of a sensor node is 17.5 m in our scheme, whereas those of LEACH and MR-LEACH are 22.1 m and 21.6 m, respectively. Our scheme has the shortest communication distance because it chooses a cluster header by considering the connectivity of nodes, hop counts, and the number of neighboring nodes, whereas LEACH selects a cluster header using random selection and MR-LEACH

chooses a cluster header using the energy residue of a sensor node.

Figure 10 shows the lifetime of a sensor node. In our experiment, we define a round as the period when all the sensed data from sensor nodes are aggregated into a sink node. Because LEACH performs a direct connection between a node and its cluster header, it shows the worst performance among three protocols. For uniform distribution, the average lifetime of sensor nodes for our protocol is 130.3 rounds, whereas those of LEACH and MR-LEACH are 54.1 and 77.8 rounds, respectively. Even though MR-LEACH provides multihop communication between cluster headers and a sink node, our protocol outperforms MR-LEACH because it generates uniformly distributed clusters with the shortest communication distance between sensor node and its header.

For the network lifetime, we measure the number of disconnected sensor nodes in network due to their energy consumption for each round. Figure 11 shows the lifetime of the sensor network with respect to LEACH, MR-LEACH, and our protocol. By following [20], a sensor network is no longer effective and available when the sensor network has less than 50% active sensor nodes in the network. When the number of rounds is 100, there is no active sensor node in LEACH, whereas MR-LEACH and our protocol have 49% and 87% of active sensor nodes, respectively. Because our protocol has the shortest communication distance, it improves the network lifetime up to 150% compared to LEACH and MR-LEACH.

To verify the reliability of routing protocols, we evaluate the response rate of a sensor node under varying communication ranges from 0 m to 150 m. By sending hello messages

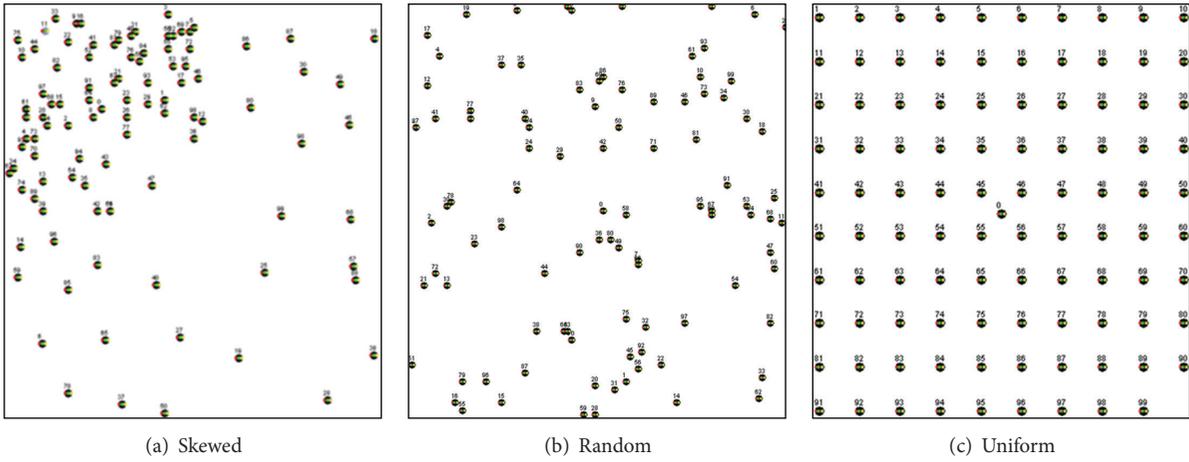


FIGURE 6: Sensor node distribution.

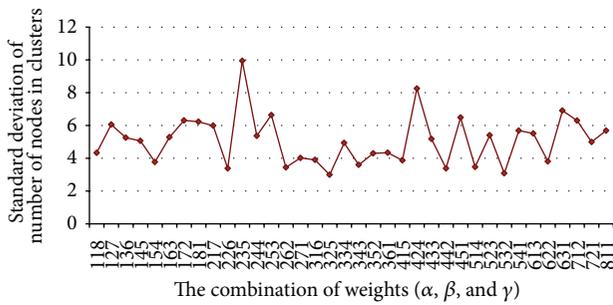


FIGURE 7: Standard deviation of the number of cluster members for the different weights.

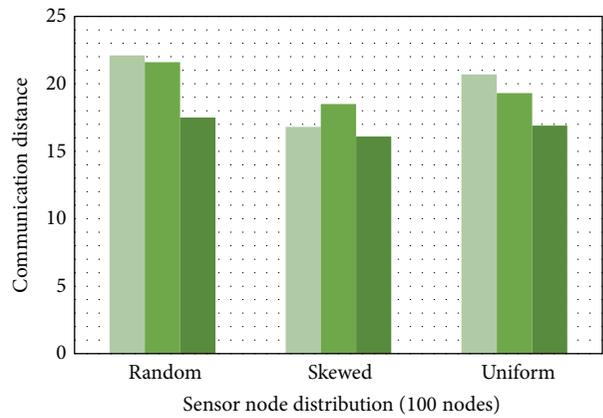


FIGURE 9: Communication distances with varying network distribution.

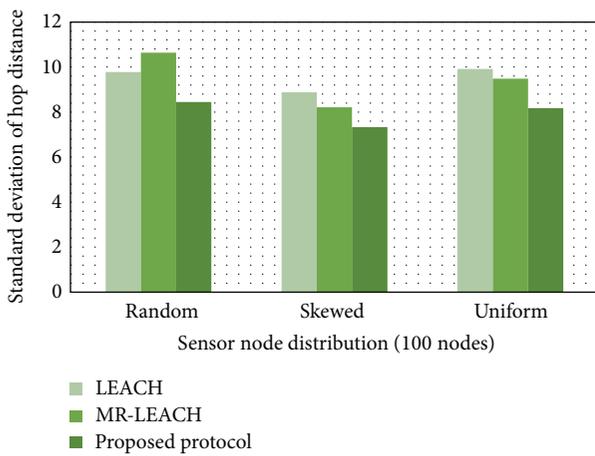


FIGURE 8: Standard deviation of hop distance among the nodes in the same cluster with varying network distribution.

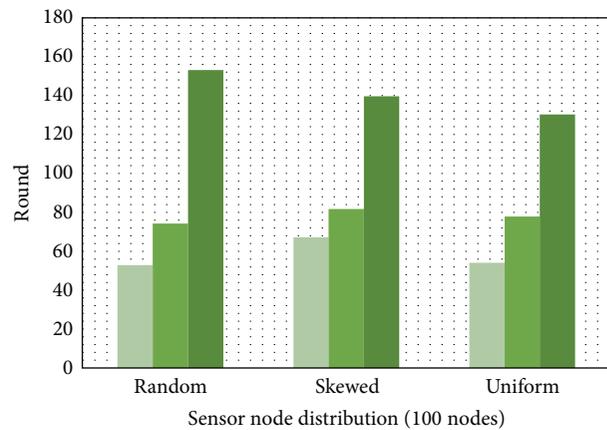


FIGURE 10: Sensor node lifetime.

to all the sensor nodes including cluster headers, the sink node calculates the ratio of the number of the received acknowledge messages to the number of sensor nodes. As shown in Figure 12, when the communication range is 15, the

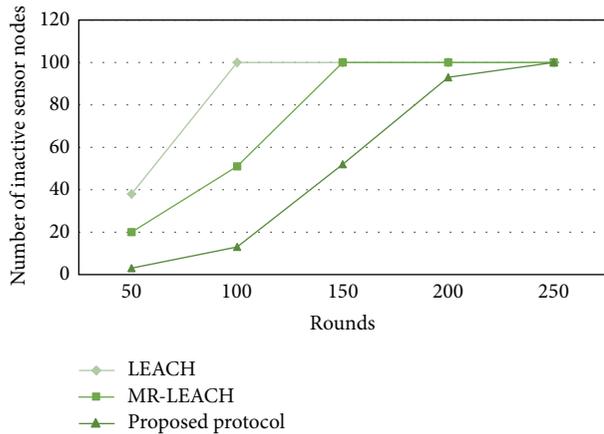


FIGURE 11: Network lifetime.

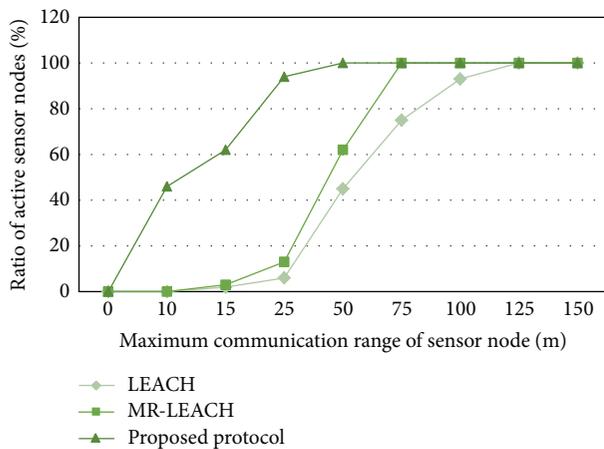


FIGURE 12: Sensor node response rate with varying communication range.

response ratio of the sensor nodes is 60%, whereas LEACH and MR-LEACH show only 2% and 3%, respectively, in terms of the response rate. Because our scheme provides both inter- and intracluster multihop routing, the sensor nodes with the limited communication range in our scheme can send messages to the sink node, via cluster members and a cluster header. Therefore, our routing protocol can support a whole network communication when the communication range is greater than a quarter of the whole network communication range.

5. Conclusion

In this paper, we proposed an energy-efficient cluster-based routing protocol for WSNs. Our protocol is based on a centralized clustering approach by using a representative path. A representative path is generated to select cluster headers and to form clusters in a distributed manner. To provide reliable network connectivity, we measure the message success rate of a sensor node when generating a representative path. To increase the lifetime of network, we select cluster headers as nodes having high connectivity. Therefore, the burden of

network configuration and routing from sensor nodes can be greatly reduced. From our performance analysis, we show that our routing protocol outperforms both LEACH and MR-LEACH, in terms of energy efficiency and network reliability. As a future work, we will study a private data aggregation scheme with the energy-efficient routing protocol.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

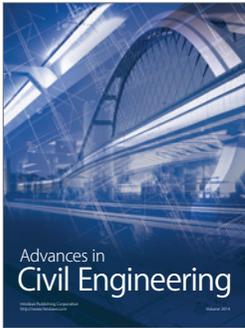
Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Grant no. 2013010099). Also, this work was supported by the Brain Korea 21 PLUS Project, National Research Foundation of Korea.

References

- [1] I. F. Akyildiz and M. C. Vuran, "WSN applications," *Wireless Sensor Networks*, pp. 17–35, 2010.
- [2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [3] Y. Ha, H. Kim, and Y. Byun, "Energy-efficient fire monitoring over cluster-based wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 460754, 11 pages, 2012.
- [4] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [5] M. O. Farooq, A. B. Dogar, and G. A. Shah, "MR-LEACH: multihop routing with low energy adaptive clustering hierarchy," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM '10)*, pp. 262–268, Venice, Italy, July 2010.
- [6] Z. Yong and Q. Pei, "A energy efficient clustering routing algorithm based on distance and residual energy for wireless sensor networks," *Procedia Engineering*, vol. 29, pp. 1882–1888, 2012.
- [7] N. Javaid, T. N. Qureshi, A. H. Khan, A. Iqbal, E. Akhtar, and M. Ishfaq, "EDDEEC: enhanced developed distributed energy-efficient clustering for heterogeneous wireless sensor networks," *Procedia Computer Science*, vol. 19, pp. 914–919, 2013.
- [8] A. Jorio, B. Elbhiri, and D. Aboutajdine, "A new clustering algorithm in WSN based on spectral clustering and residual energy," in *Proceedings of the 7th International Conference on Sensor Technologies and Applications*, pp. 119–125, 2013.
- [9] S. A. Nikolidakis, D. Kandris, D. D. Vergados, and C. Douligeris, "Energy efficient routing in wireless sensor networks through balanced clustering," *Algorithms*, vol. 6, no. 1, pp. 29–42, 2013.
- [10] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

- [11] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 22–31, Atlanta, Ga, USA, September 2002.
- [12] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 221–235, ACM, July 2001.
- [13] T. Saravanan, G. Saritha, and V. Srinivsan, "A analysis of flat routing protocols in sensor N/W," *Middle-East Journal of Scientific Research*, vol. 20, no. 12, pp. 2566–2570, 2014.
- [14] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 70–84, July 2001.
- [15] L. Li and J. Y. Halpern, "Minimum-energy mobile wireless networks revisited," in *Proceedings of the IEEE International Conference on Communications (ICC '01)*, vol. 1, pp. 278–283, June 2000.
- [16] L. Lakshmanan and D. C. Tomar, "Location dependent RB multicast routing in Wireless sensor networks using GPS based system," *Indian Streams Research Journal*, vol. 4, no. 1, pp. 1–6, 2014.
- [17] S. M. Tornell, E. Hernandez-Orallo, C. T. Calafate, J. C. Cano, and P. Manzoni, "An analytical evaluation of a Map-based Sensor-data Delivery Protocol for VANETs," in *Proceedings of the IEEE 14th International Symposium and Workshops on World of Wireless Mobile and Multimedia Networks (WoWMoM '13)*, pp. 1–6, IEEE, 2013.
- [18] Y. Panthachai and P. Keeratiwintakorn, "An energy model for transmission in Telos-based wireless sensor networks," in *Proceedings of the International Joint Conference on Computer Science & Software Engineering (JCSSE '07)*, 2007.
- [19] P. Levis and N. Lee, *Tossim: A Simulator for Tinyos Networks*, vol. 24, University of California, Berkeley, 2003.
- [20] K. J. Choi, M. J. Yoon, I. B. Sim, and J. Y. Lee, "ECS: energy efficient cluster-head selection algorithm in wireless sensor network," *Journal of Communications and Networks*, vol. 32, no. 6, pp. 1342–349, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

