Dynamic Cluster Header Selection and Conditional Re-clustering for Wireless Sensor Networks

Jinsuk Baek, *Member*, IEEE, Sun Kyong An, and Paul Fisher

Abstract — *We propose a new cluster header selection mechanism together with a new cluster formation scheme. With this scheme, each sensor node within a cluster evaluates its relative energy consumption compared to other nodes in a same cluster. Based upon the relative amount of energy consumption in the current round, sensor nodes autonomously select a time frame where they will act as a cluster header in that next round. In addition, they are conditionally allowed to switch their cluster header depending on the signal strength from their current cluster header. Our simulation results show the proposed scheme increases the network lifetime and provides a well-balanced energy consumption pattern among the nodes in a cluster compared to previously proposed schemes*[1].

Index Terms — Wireless sensor network, Network lifetime, CH selection, Re-clustering.

I. INTRODUCTION

A *wireless sensor network* (WSN) consists of a large number of sensor nodes and a *base station* (BS). It provides an efficient extraction of data while reliably monitoring the network over a variety of environments based upon the data transmitted from the sensor nodes.

We note that the sensor nodes are very limited in power, computational capacities, and memory. Of course, it is impossible to recharge the sensor node batteries because they are left unattended once deployed in the field. Hence, the routing algorithm of the network should be designed to be energy efficient allowing for the maximal lifetime of the network. Routing algorithms can be broadly divided into two categories – namely direct routing and indirect routing using a cluster approach. In direct routing algorithms [1, 2], each sensor node directly transmits the acquired data to the BS. Conversely, indirect routing algorithms [3, 4] involve a clustering algorithm that creates multiple clusters of sensor nodes. These clusters elect a *cluster header* (CH) node within a cluster. Under this configuration, each sensor node transmits the acquired data to their CH node rather than the BS. The CH's collect the data and transmit it to the BS.

A comprehensive study [5] on both approaches found that the indirect algorithms with clustering are more energy efficient compared to the direct algorithms. The reason for is, each non-CH node can reduce the amount of data transmitted due to the physical proximity to the CH nodes. Unfortunately, these transmission loads are shifted to the CH's of the clusters. In addition, the CH's have to process the data sent by the non-CH nodes before transmitting it to the BS. As a result, each CH node can easily become a bottleneck and each cluster experiences unbalanced energy consumption for the CH node as long as the CH node has an identical power configuration to the non-CH nodes. This unbalanced energy consumption of the CH node can quickly disable the entire network if the communications are prolonged.

Many proposals have focused on this issue. However, they have their own performance limitation. We proposed a new CH selection scheme aimed at maximizing the operational network lifetime. With our scheme, the current CH node adjusts the next CH sequence in a cluster based upon the self-incentive information claimed by sensor nodes which have satisfied the requirement in terms of the minimum amount of data detection. In addition, our proposed scheme includes a semi-reclustering mechanism which allows each non-CH node to adaptively switch their cluster assignment when it determines that the current cluster is suboptimal compared to other overlapping clusters.

The remainder of the paper is organized as follows. In Section 2, we continue with related work. In Section 3, we propose the new scheme which we call *self-incentive* and *semi-reclustering* (SISR). Section 4 shows the performance of the proposed scheme and we provide conclusions in Section 5.

II. RELATED WORK

Many different approaches have been described to design feasible WSNs. As our concern is CH selection schemes, we discuss some of the associated schemes. In indirect routing algorithms, the data aggregation [6–9] at the CH node eventually allows it to transmit a reduced volume of effective data to the BS. This feature saves bandwidth and improves system capacity. But, if the CH node runs out of energy, the complete network cluster dies, providing no guarantee on system lifetime. This problem was addressed in many proposals by the addition of a CH reselection scheme.

Among them, LEACH [3], allows each cluster to reselect the CH node at proper intervals. While this scheme showed

---

[1] An earlier version of this paper was presented at the 2009 IEEE Sarnoff Symposium. This present paper is an improved version which has been extended to include a new mechanism and comprehensive simulation results.

Jinsuk Baek is with the Department of Computer Science, Winston-Salem State University, Winston-Salem, NC 27110 USA (e-mail:baekj@wssu.edu).

Sun Kyong An is with the Gabriels Technology Solution Inc, New York, NY 10001 USA (e-mail:san@gabriels.net).

Paul Fisher is with the Department of Computer Science, Winston-Salem State University, Winston-Salem, NC 27110 USA (e-mail:fisherp@wssu.edu).

Contributed Paper
Manuscript received 09/03/10
Current version published 12/23/10
Electronic version published 12/30/10.

partial success, we point out that it requires a new cluster construction process at every round. With cluster construction, each cluster has to reselect a new CH node with random probability, and among the potential CH nodes, the actual node should be adaptively optimized for minimal communication distances to the maximum number of network member nodes. Due to this repetitive cluster set-up phase, the nodes of the cluster have to spend additional delay and energy resulting in at worst only a suboptimal solution. EARACM [10] selects some overhearing nodes as relay nodes. These schemes adopted the multi-hop communication to further reduce the energy consumption. Unfortunately, this benefit comes from sacrificing the resulting transmission delay and communication overhead since each relay node has to maintain the status of the other relay nodes. A more recent approach [4] eliminates the repetitive set-up phase by pre-determining the CH sequence at the initial set-up phase. We also consider the fact that the sensor nodes usually remain in the sleep mode but wake up when they detect events from the environment and transmit the acquired data to their CH node. With this fact, the operational scheme forces the sensor nodes detecting events to consume more energy than other nodes which are still on sleep mode. When this occurs, the fixed CH sequence cannot reflect the energy levels of any proposed CH's before final selection in order to provide the longest network lifetime.

III. PROPOSED SCHEME – SISR

A. Basic Idea

Each sensor node in a cluster has a different amount of available energy at any given time depending upon 1) the frequency of the sleep/wake cycles, and 2) the amount of detected/transmitted data. We recognize that the sensor nodes, actively involved in data detection and transmission processes, tend to move into an abnormal status more quickly than other lesser used nodes.

Under this situation, giving the same duty as a CH node to all sensor nodes with a random probability is not fair and results in a suboptimal network lifetime. In order to resolve this problem, our proposed scheme allows each sensor node to determine its own incentive based upon the amount of energy thus far spent for data detection and transmission. This incentive value indicates the number of rounds the sensor node will be exempted from CH consideration while the determination of the new CH node is taking place. Depending upon the suggested incentive by each sensor node, the current CH node reschedules the CH sequence that will be used in the next round.

Once deployed, the cluster is fixed and the initial CH sequence is determined by the signal strengths from the initial CH node in a cluster. As the sensor network operates multiple rounds, it is possible that a non-CH node recognizes a CH node in a neighbor cluster physically closer than its current CH node. In such a case, our scheme allows a non-CH node to join another cluster in order to reduce energy consumption, a semi-clustering. Note that this switching is not always allowed and not also allowed for all non-CH nodes in a cluster because frequent re-clustering can negatively affect the overall network lifetime. Instead, this only involves the non-CH nodes that satisfy requirements, to be defined later.

B. Set-up Phase

A typical operation for a single cluster was proposed in [3] where the operation is divided into rounds. Each round consists of two separate phases – namely a set-up phase and a steady phase. At the initial set-up phase, we propose the following sequence of actions.

1. The BS broadcasts a *HELLO* message to all the nodes. Based upon the received signal strength, each node computes the approximate distance to the BS.
2. Upon receiving *HELLO* message, nodes become candidate CH nodes with a random probability *P* and broadcast the *ADVERTISE* message with an initial radio range *RR*. This *RR* value is continuously increased until the node receives at least one *ADVERTISE* message from other nodes.
3. Based on the *ADVERTISE* messages, each node checks if there is a candidate CH node having higher *P* than itself. If found, it selects the node as the CH node and gives up the competition. Otherwise, it will be elected as a CH node by the other nodes. The elected CH node will broadcast *INVITE* message with its current radio range *RR*. Once each node decides its CH node, it transmits a *JOIN_REQ* message to the CH node. In this way each node makes autonomous decisions without any centralized control.
4. The CH node receives multiple *JOIN_REQ* messages from its non-CH nodes in the forming cluster. Based upon the signal strength of these messages, it decides the CH sequence for the current round and transmits this *SCHEDULE* information to its non-CH nodes.
5. By referencing the sequence number sent by the CH node, each non-CH node can recognize when it has to become a CH node. For example, if a non-CH node receives a sequence number 7 from the current CH node, the node will become a CH node in the 8th frame.

If we assume 1) single-hop symmetric communications among the sensor nodes; 2) each node has l bits of data to be transmitted over distance d, or to the nodes in an $M\times M$ region with k clusters, the amount of energy consumption for the two types of sensor nodes can be expressed as shown:

$$E_{Tx}(l,d)=\begin{cases} l\varepsilon_{elec}+l\varepsilon_{fs}\dfrac{M^2}{2\pi k} & \text{, node to node} \\ l\varepsilon_{elec}+l\varepsilon_{mp}d^4 & \text{, CH-to-BS} \end{cases} \qquad (1)$$

In both forms of equation (1), ε_{elec} represents the energy consumption of the radio dissipation referring to the electronic energy, while ε_{fs} represents the same relationship for amplifying the radio signal referring to the amplifier energy in free space channel mode. ε_{mp} refers to the same in a multi-path fading channel mode [11].

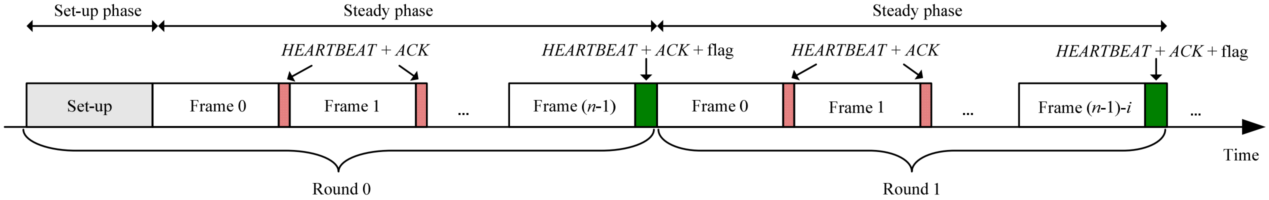

Fig. 1. **An example of the round structure for the proposed scheme**

TABLE 1
AN EXAMPLE OF THE INCENTIVE BASED CH SEQUENCE

Round *rd*			**Round *rd*+1**			**Round *rd*+2**		
node_id	CH sequence	incentive	node_id	CH sequence	incentive	node_id	CH sequence	incentive
node 0	0	0	node 0	0	0	node 0	0	0
node 1	1	0	node 1	1	0	node 1	1	0
node 2	2	***2***	node 2	**-1**	***2***→***1***→***3***	node 2	**-1**	***3***→***2***
node 3	3	0	node 3	2	0→***1***	node 3	**-1**	***1***→0
node 4	4	***1***	node 4	**-1**	***1***→0	node 4	2	0
node 5	5	0	node 5	3	0	node 5	3	0
node 6	6	***3***	node 6	**-1**	***3***→***2***	node 6	**-1**	***2***→***1***
node 7	7	0	node 7	4	0→***1***	node 7	**-1**	***1***→0
node 8	8	***1***	node 8	**-1**	***1***→0	node 8	4	0
node 9	9	0	node 9	5	0	node 9	5	0

Under the same assumptions, a CH node consumes additional energy for receiving data and scheduling the CH sequence as given in Equations (2) and (3) when there are N sensor nodes and k clusters.

$$E_{Rx}(l) = l\,\varepsilon_{elec}\left(\frac{N}{k} - 1\right) \tag{2}$$

$$E_{Sx}(l) = l\,\varepsilon_{sch}\frac{N}{k}, \tag{3}$$

where ε_{sch} represents the energy consumption for scheduling the sequence. We define l_A, l_I, and l_J as a size of the *ADVERTISE* message, *INVITE* message and *JOIN_REQ* message, respectively, such that $l_A << l$, $l_I << l$, and $l_J << l$.

Accordingly, the amount of energy consumption by the two different types of sensor nodes during the set-up phase will obey Equations (4) and (5), respectively.

$$\begin{aligned} E_{\text{Setup\_CH}} &= E_{Tx}(l_A,d) + E_{Rx}(l_A) + E_{Tx}(l_I,d) \\ &\quad + E_{Rx}(l_J) + E_{Sx}(l) + E_{Tx}(l,d)\,. \end{aligned} \tag{4}$$

$$\begin{aligned} E_{\text{Setup\_non-CH}} &= E_{Tx}(l_A,d) + E_{Rx}(l_A) \\ &\quad + E_{Tx}(l_J,d) + (l_I + l)\varepsilon_{elec}\,. \end{aligned} \tag{5}$$

C. Steady Phase

Steady phase is initially operated with multiple rounds, each having multiple frames equal to the number of sensor nodes in a cluster. Once TDMA-based access protocol is assumed, a CH node receives data from its non-CH nodes according to a fixed schedule while a non-CH node transmits its data only when 1) there is an available, dedicated time slot for the node; and 2) they have detected activity of interest. At the end of each frame, the CH node broadcasts a *HEARTBEAT* message to remove the abnormal nodes, which do not respond with a *HEARTBEAT-ACK* message. Based upon the messages, the CH constructs a *DEAD_NODE* message including a list of dead nodes and broadcasts it to its non-CH nodes. Lastly, each non-CH node transmits their *HEARTBEAT-ACK* message, including their incentive values with a flag bit at the end of the each round. The content of the flag bit indicates the existence of the incentive. Based on these messages, the CH node creates a {node_id} list, which is a list of non-CH nodes that have requested exemption from CH node consideration. The CH node broadcasts the list to the non-CH nodes. That is, the CH node informs its non-CH nodes which nodes will not act as a CH node for the next round. Each node now reschedules their CH sequence by decreasing its current CH sequence number. Of course, this is dependent on the number of nodes in the {node_id} list having a smaller node_id than its own node_id. As a result, the cluster has n–i frames in the next round, when there are n nodes including i nodes, which have requested an exemption from CH node consideration. Fig. 1 shows an example of the round structure for the proposed scheme. For example, there are initially 10 nodes (node 0 – node 9) in a cluster and the node 9 is the CH node for the last frame of the round *rd*. Node 2 has an incentive value 2, node 4 has an incentive value 1, node 6 has an incentive value 3, and the node 8 has an incentive value 1. These four nodes will: 1) send [*HEARTBEAT-ACK* + 1]; 2) decrease their incentive values by 1 for the round *rd*+1, resulting in 1, 0, 2, and 0, respectively; and 3) set their CH sequence values to –1.

On the other hand, other nodes will send [*HEARTBEAT-ACK* + 0] to the CH node. The CH node broadcasts the information to other nodes, saying that nodes 2, 4, 6, and 8 will be waived from CH node consideration in the next round. The nodes then adjust their sequence numbers. As a result: 1) the sensor nodes having node_id 0 or 1 do not change their sequence; and 2) the node 3, 5, 7, and 9 adjust their sequence to 2, 3, 4, and 5, respectively. There will now be 6 frames in the round *rd*+1. A similar procedure is performed at round *rd*+2, and shown in Table 1 and Fig. 2.

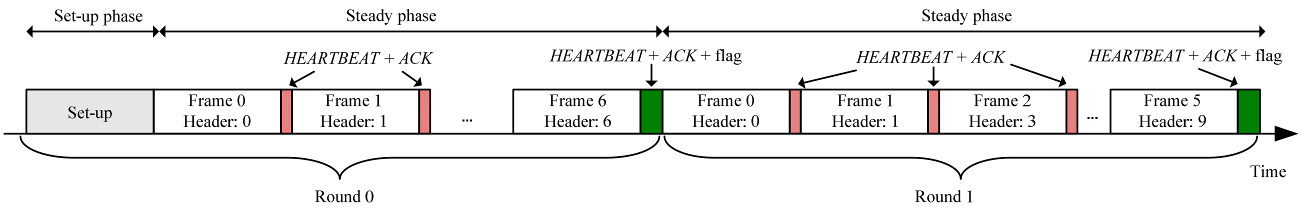

Fig. 2. An example of the incentive-based CH sequence for each frame

When we set the energy consumption $E_{Ax}(l)$ for aggregating data sent by its non-CH nodes to the same amount of $E_{Sx}(l)$, the quantity of energy consumed in the two different types of sensor nodes in a single frame during the steady phase will obey Equations (6) and (7), respectively. For a CH node, note that the first transmission is CH-to-BS - long distance transmission - while the second one is a short distance transmission for the *HEARTBEAT* message.

$$\begin{aligned} E_{\text{Steady\_CH\_F}} &= E_{Rx}(l) + E_{Ax}(l) + E_{Tx}(l, d) + E_{Tx}(h, d) \\ &\quad + E_{Rx}(h)\, P_a[f] + E_{Tx}(h'(f), d), \end{aligned} \tag{6}$$

$$\begin{aligned} E_{\text{Steady\_non-CH\_F}} &= E_{Tx}(l, d) \\ &\quad + [E_{Tx}(h, d) + (h + h'(f))\, \varepsilon_{elec}]\, P_a[f], \end{aligned} \tag{7}$$

where $P_a[f]$ is the probability of a node being alive at frame f. Therefore, $P_a[f]$ converges to 0 when f goes to ∞. Let $f(rd)$ be the number of frames at round rd, then it has the following properties. That is, $P_a[f] \ge P_a[f+1]$, if $0 \le f \le f(rd)-2$ and $P_a[f] \ge 0$, if $f = f(rd)-1$. Also, h is the size of *HEARTBEAT* message such that $h << l$ while $h'(f)$ is the size of *DEAD_NODE* message at a given frame f. As we set each bit in *DEAD_NODE* message to represent each dead node and define $n(1-P_a[f])$ as an event counter, which is increased by at most one in each frame when a frame includes at least one newly dead node, the size $h'(f)$ can be expressed as:

$$h'(f) = \begin{cases} 1 & \text{, if } n(1-P_a[f]) = 0 \\ \dfrac{N}{k} & \text{, if } n(1-P_a[f]) = 1 \\ h'(f-1) - h'(f-1)\cdot(1-P_a[f]) & \text{, if } n(1-P_a[f]) > 1 \end{cases} \tag{8}$$

Therefore, it can be bounded by:

$$1 \le h'(f) \le \frac{N}{k}. \tag{9}$$

At the last frame of each round rd, the CH node spends additional energy for aggregating and broadcasting the incentive information with size $h''(rd)$. On the other hand, the non-CH nodes need to send $h+1$ bits instead of h to include the existence of incentive value. Let $P_I[rd]$ be the probability of a node having an incentive value at round rd and $h'(lf)$ is the size of the *DEAD_NODE* message of the last frame of the round rd. Similar to the $h'(f)$, $h''(rd)$ can be defined by:

$$h''(rd) = \begin{cases} 0 & \text{, if } P_I[rd] = 0 \\ h'(lf) - h'(lf)\cdot(1-P_a[lf]) & \text{, if } P_I[rd] > 0 \end{cases} \tag{10}$$

Therefore, it can also be bounded by:

$$0 \le h''(rd) \le \frac{N}{k} \tag{11}$$

Accordingly, the energy consumption of two types of nodes at the last frame of a given round rd can be expressed by Equation (12) and (13).

$$\begin{aligned} E_{\text{Steady\_CH\_LF}} &= E_{\text{Steady\_CH\_F}} + [E_{Rx}(h+1) - E_{Rx}(h) \\ &\quad + E_{Ax}(h+1) + E_{Tx}(h'', d)]\, P_a[f(rd)-1] \end{aligned} \tag{12}$$

$$\begin{aligned} E_{\text{Steady\_non-CH\_LF}} &= E_{Tx}(l, d) + [h\, \varepsilon_{elec} + E_{Tx}(h+1, d) \\ &\quad + h''\, \varepsilon_{elec}]\, P_a[f(rd)-1] \end{aligned} \tag{13}$$

We also need to consider a non-CH node that becomes dead as status right after sending a *HEARTBEAT-ACK* message to its CH node. This scenario is possible because the non-CH node had almost same amount of energy that is required to send the *HEARTBEAT-ACK* message but no more remaining energy to be spent for the next frame. In order to avoid this situation, our scheme requires each non-CH node to send the *HEARTBEAT-ACK* only when it has sufficient amount of energy even after sending the *HEARTBEAT-ACK* message. When we define $E(S_i)$ as a remaining energy of node i , this restriction can be represented by:

$$E(S_i) > \alpha \cdot E_{Tx}(h, d), \tag{14}$$

where α is a *HEARTBEAT-ACK* coefficient such that $\alpha > 2$. A node i decides its incentive value $I(i, rd)$ at round rd, with $f(rd)$ frames by considering 1) the amount of energy consumed as a non-CH in the current round rd; and 2) the average amount of energy consumption of the CH nodes in a previous round, which is available through the message h' sent by last CH of the round. As we consider the node i acting as a CH node at frame k, the incentive value $I(i, rd)$ can be expressed as shown in Equation (15).

$$I(i, rd) = \begin{cases} \left\lfloor \dfrac{\sum_{j=0,\, j\ne k}^{f(rd)-1} E(i)_j}{E(CH)_0} \right\rfloor & \text{, if } rd = 0 \\[3ex] \left\lfloor \dfrac{A(i, rd-1) + \sum_{j=0,\, j\ne k}^{f(rd)-1} E(i)_j}{\dfrac{1}{f(rd-1)} \sum_{j=0}^{f(rd-1)-1} E(CH)_j} \right\rfloor & \text{, if } rd \ge 1 \end{cases} \tag{15}$$

$$A(i, rd) = \begin{cases} \displaystyle\sum_{j=0, j\neq k}^{f(rd)-1} E(i)_j - I(i, rd)E(CH)_0 & \text{, if } rd = 0 \\ \displaystyle A(i, rd{-}1) + \sum_{j=0, j\neq k}^{f(rd)-1} E(i)_j - I(i, rd)\frac{1}{f(rd-1)} \sum_{j=0}^{f(rd-1)-1} E(CH)_j & \text{, if } rd \geq 1 \end{cases} \quad (16)$$

Due to the difficulty in obtaining the actual average amount of energy consumption of the CH node for the first round, $E(CH)_0$, we use a standard energy consumption value that has been pre-calculated for each cluster. However, we use the actual amount of energy expended for the second round.

Note that $A(i, rd–1)$ is a fractional number after the decimal point of the real incentive value, which should be transferred from the previous round $rd–1$ to the current round rd for sensor node i. Therefore, this value stands for accumulated value, which is not considered to the incentive value of the sensor node i in its previous round $rd–1$, although it contributed to the corresponding sensor network where it is deployed. This accumulated value can be calculated by Equation (16).

At a given round rd, we recognize all sensor nodes in a cluster are actively involved in data detection and transmission processes especially when there are continuous significant events in a given cluster. In such case, all nodes will request at least one incentive at the end of round rd. This results in no available node willing to act as a CH node in round $rd+1$. In order to cope with this situation, the proposed scheme requires the last CH node of the current round rd to decrease incentive values of all nodes by 1, until it finds at least one node having a positive CH sequence number, which guarantees $f(rd+1) \geq 1$.

D. Semi-Reclustering

A non-CH node can recognize that the CH node of the neighboring cluster has an even stronger signal than that of its current CH node. This is a possible scenario since the cluster is initially fixed. Our semi-reclustering scheme allows such nodes to switch their CH nodes.

Even though this switching requires a small amount of energy consumption, too frequent switching negatively affects the overall network lifetime. Therefore, switching is only allowed for non-CH nodes located at the boundary of the clusters. This can be implemented by grouping the nodes in multiple groups depending upon their signal strength to their CH node during the initial set-up phase. Consequently, only the non-CH nodes, which are in the weakest group, are allowed to switch their clusters. Once the node switches its cluster, it can save transmission energy as the physical distance to its CH node of new cluster is closer than that of the previous cluster. However, this simple method alone cannot always guarantee a performance improvement in terms of network lifetime.

Let us consider the following scenario shown in Fig. 3. In this example, node 9 of cluster 2 is located in the weakest group and current CH node is node 5 in both clusters. Node 9 will switch to cluster 1 as it detects node 5 of cluster 1 has a stronger signal than that of its current cluster. However, it experiences the opposite phenomenon one frame later whenever it switches its cluster. This cluster switching is repeated until node 8 acts as a CH node, resulting in unnecessary energy consumption.

In order to avoid this effect, we design the semi-reclustering process to permit continuous activity at appropriate times due to the change in CH nodes with their associated distances and the topography of the environment. To design our semi-reclustering scheme, we consider the fact that the initial CH sequence is determined based upon the signal strength from the initial CH node. We consider the remote field, where nodes are randomly deployed. Therefore, we can conclude that if the distance between a given node and its current CH node is much longer than the average distance between arbitrary sensor nodes and their CH nodes, the probability that its next CH node in a same cluster is close enough to the given node is extremely high.

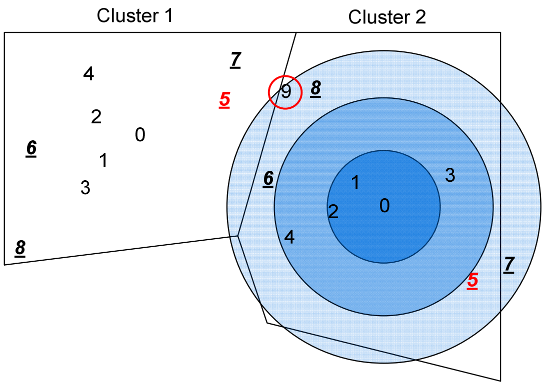

Fig. 3. A Scenario of frequent cluster switching

Once the CH node at the center of the cluster is assumed and the nodes are deployed in the remote area with distribution $\rho(x, y)$, the expected squared distance, $E[d^2]$, from the non-CH node to the CH node is given by

$$E[d^2] = \iint (x^2 + y^2)\rho(x, y)dxdy = \iint r^2 \rho(r, \theta) r dr d\theta \; . \quad (17)$$

If we also assume the area occupied by a cluster is a circle-shaped region with radius R and $\rho(r, \theta)$ is constant for r and θ, Equation (17) can be simplified into:

$$E[d^2] = \rho \int_{\theta=0}^{2\pi} \int_{r=0}^{R} r^3 dr d\theta = \frac{\rho M^4}{2\pi k^2} \; . \quad (18)$$

When the nodes are uniformly distributed in a cluster area M^2/k, ρ is equal to $1/(M^2/k)$, which results in:

$$E[d^2] = \frac{M^2}{2\pi k} \approx d^2 . \tag{19}$$

Therefore, the amount of energy consumption for short distance transmission, which is defined in Equation (1), can be rewritten by:

$$E_{Tx}(l, d) = l\varepsilon_{elec} + l\varepsilon_{fs}\frac{M^2}{2\pi k} \approx l\varepsilon_{elec} + l\varepsilon_{fs}d^2 \tag{20}$$

Based on these observations, we can define the cluster switching threshold T_S as $\beta E[d^2]$ with switching coefficient β. Let us consider two involved CH nodes, which are CH i and CH j in cluster i and j, respectively. The qualified non-CH node, located in the weakest group of the cluster i, decides to switch its CH node if the following condition is satisfied.

$$\begin{aligned} &0 < (\varepsilon_{fs}d_i^2 - \varepsilon_{fs}d_j^2) \le \beta\varepsilon_{fs}E[d^2] \\ &\rightarrow 0 < (d_i^2 - d_j^2) \le \beta E[d^2], \end{aligned} \tag{21}$$

where d_i^2 and d_j^2 is a square distance from a given node to CH i and CH j, respectively, and $d_i^2 > d_j^2$.

Since the expected square distance is constant in a given cluster, we can allow more non-CH nodes, having larger square distance gap between two CH nodes, to switch their cluster by increasing the value of β. In order to prevent a non-CH node from frequently switching between two cluster i and j, we can adjust the value of β by considering the maximum distance gap between two CH nodes, which should be satisfied to switch its cluster as follows.

$$\begin{aligned} \beta &= 1.6\cdot\max\{d_i - d_j | \, d_i > d_j\}\cdot\frac{\gamma\cdot 10}{\max\{d_i - d_j \mid d_i > d_j\}} \\ &= 16\cdot\gamma \end{aligned} \tag{22}$$

That is, γ represents the relative distance gap to the given two CH nodes compared to the maximum distance gap between the two CH nodes and this is a percentage unit. For example, if the radius R of the two clusters is 10, the maximum distance gap is equal to 20. When we set the γ to 10%, the β becomes 1.6. Also, when d_i is 11 and d_j is 9, the square distance gap is equal to 40. The expected square distance is fairly assumed to 25, the node is allowed to switch its cluster since $40 \le 1.6\cdot 25$.

To join the new cluster, a node follows the steps described in Section III.2. However, this switching is only allowed at the beginning of each round for the new cluster. The most straightforward manner to implement this would be to require the non-CH nodes to inform their current CH node using a *HEARTBEAT-NAK* message at the point they decide to switch to another cluster. However, we recognize this method will not allow the non-CH nodes to join the new cluster until the new cluster starts its next round. This is because each cluster has a different number of frames in each round, resulting in non-identical periods of rounds. Therefore, the non-CH node, which decides to switch its cluster, will not transmit detected data to its current CH node after sending a *HEARTBEAT-NAK* message. It also needs to wait for termination of the current round of its new cluster. During this period, the events of the area covered by the node are not sent to the current CH node.

Another method to minimize this critical period would be to require the non-CH node to send the sensed data to its current CH node even after it decides to switch its cluster. Instead of *HEARTBEAT-ACK* or *HEARTBEAT-NAK* message, it will send a *HEARTBEAT-NAK* message with a flag bit. The content of the flag bit indicates the node will be detached from its current cluster and attached to another cluster. Once the last CH node of the current round receives a [*HEARTBEAT-NAK*, 1] message from its non-CH node, the node is not considered for scheduling the CH sequence for the next round.

However, this method requires a new type of message. Therefore, our scheme requires the non-CH node, which decided to switch its cluster, to send its acquired data to its current CH node until it detaches itself from the current cluster. Also, in order not to be considered as a CH node for the next round in its current cluster, the non-CH node sets its incentive value to 1 in every round. This procedure is repeated until it attaches itself to the new cluster when it detects the start of the new round for its new cluster. Next, the two CH nodes of the two involved clusters let the sensor nodes adjust their CH sequence by sending the most up to date {node_id} list. This can be done by node_id comparisons.

IV. PERFORMANCE

The performance of the proposed scheme was evaluated by conducting comprehensive simulations via a simulator that we developed using a Java platform. We compared our scheme with two other schemes, the low-energy, adaptive, clustering, hierarchy scheme [3], and the round-robin clustering hierarchy scheme [4]. Hereafter, we refer to these schemes as LEACH and RRCH, respectively. The proposed self-incentive and semi-reclustering scheme is abbreviated to SISR.

We performed all of our simulation experiments using 100 sensor nodes and the location of each node is randomly generated over a grid area of 100m×100m. The average link distance between CH nodes and BS is set to 85m. We also assume that each sensor node initially has 2J of energy and each knows the location of other nodes by using a clustering algorithm such as [3]. The simulation parameters are summarized in Table 2.

TABLE 2
SIMULATION PARAMETERS

Parameter	Value	Unit	Parameter	Value	Unit
ε_{elec}	50	nJ/bit	K	5	
ε_{fs}	10	pJ/bit/m^2	L	2000	bit
ε_{mp}	0.0013	pJ/bit/m^4	l_A, l_I, l_J	200	bit
ε_{sch}	5	nJ/bit/signal	H	8	bit
E_{Ax}	5	nJ/bit/signal	D	85	m
N	100		M	100	m

Three scenarios were established to test the efficiency of our SISR scheme, each constructed with a distinct set of obstacle events.

- Scenario 0 (Detect all): The obstacles always exist over the all remote area where the sensor nodes are deployed. Therefore, all non-CH nodes always have acquired data to be sent to their CH node.
- Scenario 1 (Detect none): The obstacle never exists. Hence, all sensor nodes repeat the sleep/wake cycles and contain no acquired data to be sent their CH node. The only interaction between the non-CH nodes and their CH node is for exchanging *HEARTBEAT* and *HEARTBEAT-ACK*.
- Scenario 2 (Detect randomly): The obstacles randomly exist over the remote area and the events also randomly occur over time. Therefore, each node has a different frequency of the sleep/wake cycles and different amount of acquired/transmitted data.

A. *Self-Incentive*

As scenario 2 reflects the most practical situation, we first evaluate the total incentive values, which have been requested by each node under scenario 2. In the preliminary version [12] of this paper, each sensor node i only considered the integer value to request its own incentive value $I\,(i)$ in every round. As the sensor network involves RD rounds in a lifetime and the sensor node i acts as CH node in frame k in every round, this could be expressed as Equation (23).

$$I\,(i)=\sum_{rd=0}^{RD}\left[\frac{\sum_{j=0,j\neq k}^{f(rd)-1}E(i)_j}{\frac{1}{f(rd-1)}\sum_{j=0}^{f(rd-1)-1}E(CH)_j}\right] \tag{23}$$

However, we recognized the remaining fractional number after the decimal point of the real incentive value can significantly contribute the total incentive value over the RD rounds. Therefore, our new SISR requires each sensor node i to consider the accumulated value $A(i)$ to count its total incentive value $I(i)$. Both $I(i)$ and $A(i)$ are represented in Equation (24) and (25) and recall that $I(i, 0)$ and $A(i, 0)$ has been defined in Equation (15) and (16), respectively.

$$I\,(i)=\sum_{rd=0}^{RD}\left[\frac{A(i,rd-1)+\sum_{j=0,j\neq k}^{f(rd)-1}E(i)_j}{\frac{1}{f(rd-1)}\sum_{j=0}^{f(rd-1)-1}E(CH)_j}\right] \tag{24}$$

$$A(i)=\sum_{rd=0}^{RD}\ \{A(i,rd\!-\!1)+\sum_{j=0,j\neq k}^{f(rd)-1}E(i)_j - I(i,rd)\frac{1}{f(rd-1)}\sum_{j=0}^{f(rd-1)-1}E(CH)_j\ \}. \tag{25}$$

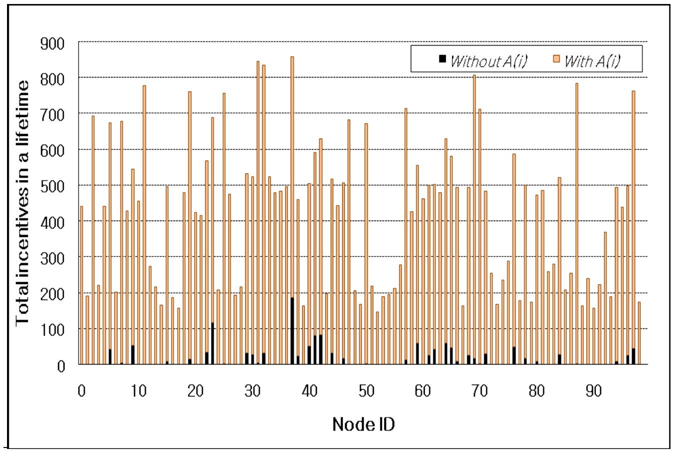

Fig. 4. The total requested incentive values in a lifetime

Fig. 4 shows the significant affect our aggressive incentive-calculation involving the accumulated value $A(i)$ has on the total incentive value $I(i)$. If we use the passive approach under scenario 2, many nodes, which are deployed in a relatively inactive area, did not request incentive values, because they consumed lesser energy than other nodes, which are located in active area. The maximum incentive value difference between the nodes is 186 in the case of the passive approach while the same is 695 for the aggressive approach. Also, for a specific node, the minimum and maximum difference between the passive and aggressive incentive-calculation approach is 146 and 836, respectively. These results indicate our aggressive incentive-calculation allows the sensor network to maintain balance among the nodes in terms of energy consumption. In addition, the different number of incentive values requested by the nodes allows both types of nodes to consume a significantly smaller energy quantity than the other two schemes.

B. *Semi-Reclustering*

Our semi-reclustering scheme allows non-CH nodes to switch their CH nodes. However, this is not always allowed for all non-CH nodes. We defined switching conditions the non-CH node should satisfy to switch their cluster. In a cluster, we first divided the sensor nodes in different groups based on their initial signal strengths to their CH node during the initial set-up phase. We also defined relative distance gap between the non-CH node and its two available CH nodes, namely threshold γ, to consider the scenario depicted in Fig. 3.

As a result, in order to switch its cluster, the non-CH node should 1) recognize that the CH node of the neighboring cluster has an even stronger signal than that of its current CH node; 2) be located in the weakest group in terms of initial signal strength from the CH node; 3) satisfy the threshold γ, which was defined in Equation (22).

Fig. 5 shows the simulation results with different γ values. We performed this simulation with 5 groups and each group has equal length of interval. As we can see, the network operational lifetime is varied with different γ values. This is because the nodes are randomly deployed in a remote area.

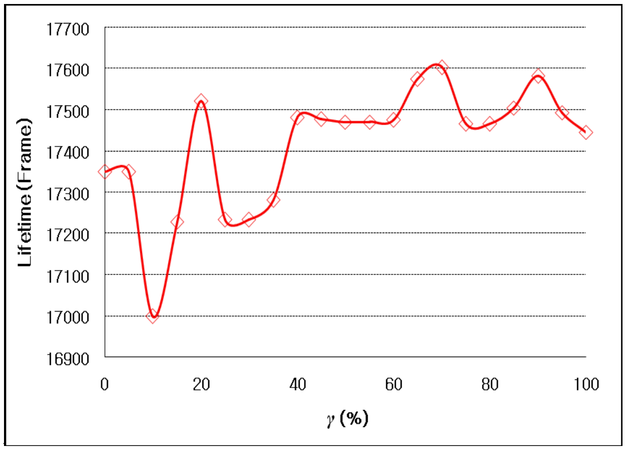

Fig. 5. The network lifetime vs. γ

There is no constant relationship between the lifetime and γ values, especially if the γ value is set to below 40%. In contrast, it shows the minimum lifetime is bounded by γ values as long as the γ value is larger than 40%. In our simulation, it shows the best performance, say 17604 frames, when γ value is set to 70%. However, we need to mention that the lifetime is varies significantly as a factor of the network topology rather than γ value. Therefore, the γ value can be adaptively set depending on the topology.

C. *Network Lifetime*

Owing to our self-incentive and semi-reclustering properties, the proposed SISR scheme enables the nodes to efficiently use their limited energy resource, resulting in a much longer lifetime than other compared schemes. In our simulation, we evaluated the network life time with three different scenarios. For the scenario 0, Fig. 6 illustrates simulation results of the network lifetime by showing the number of nodes alive as a function of frames for the three different schemes. The SISR allows the network to prolong 1425 and 605 more frames than LEACH and RRCH, respectively. However, we can see our SISR does not significantly outperform the RRCH. This is because SISR requires each node to spend extra energy to transmit and receive the incentive information in addition to the *HEARTBEAT* message.

Due to this overhead, SISR temporarily shows even worse performance than RRCH in terms of number of alive nodes especially between frame 6686 and 6713, and between frame 7521 and 7655, although it comprehensively shows better performance than other two schemes. However, this overhead is sufficiently and quickly compensated with well balanced energy consumption among the nodes. Of course, our semi-reclustering contributes such a quick compensation. After frame 8805, we can see three nodes are continuously alive for a long time. There is no alive node except for these three nodes. It means they become CH and there is no need to exchange the *HEARTBEAT* message among the three nodes, resulting in long lifetime.

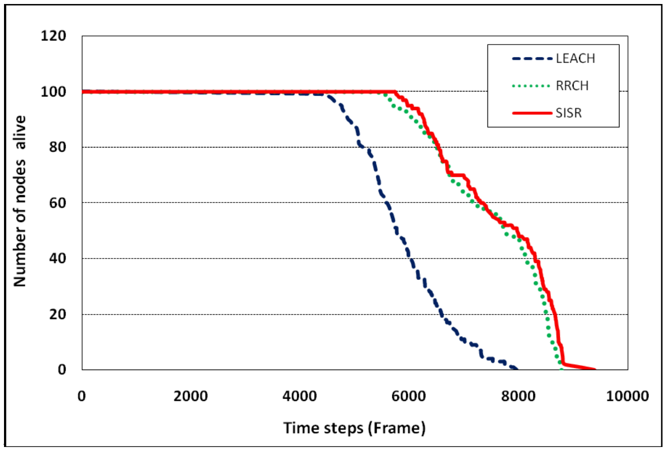

Fig. 6. The number of sensor nodes alive over the frame (Scenario 0)

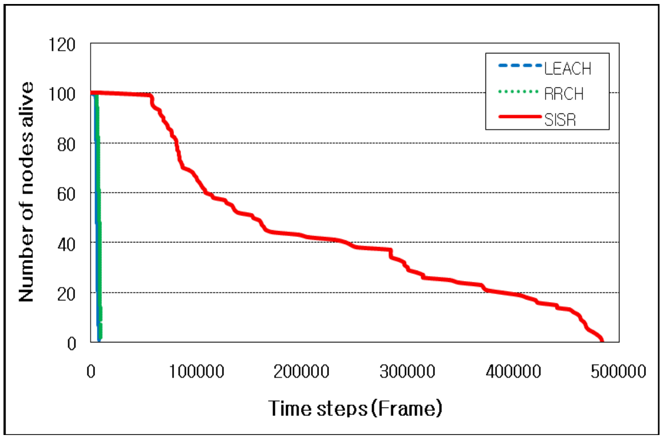

Fig. 7. The number of sensor nodes alive over the frame (Scenario 1)

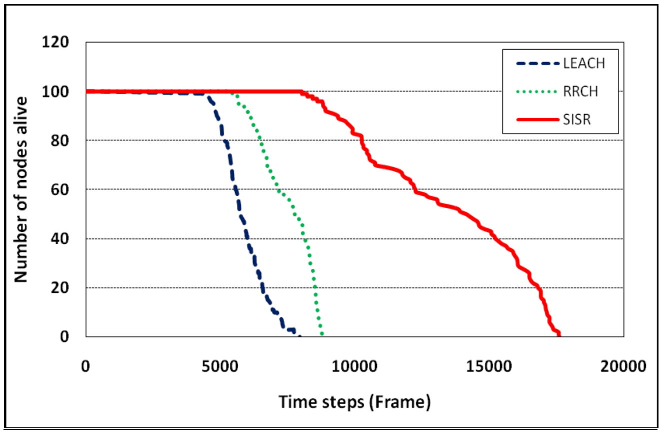

Fig. 8. The number of sensor nodes alive over the frame (Scenario 2)

On the other hand, Fig. 7 and 8 show the results of the number of nodes alive as a function of frames for scenario 1 and 2, respectively. For scenario 1, as our SISR allows the nodes to send data only when they have detected data of interest, it can save a lot of energy required 1) for long distance transmission for CH-to-BS communications; 2) for short distance transmission for sharing the incentive information among the nodes in a same cluster; and 3) for data aggregation at the CH. The only required communication is 1) for exchanging *HEARTBEAT* messages between the nodes and their CH; and 2) rare long distance transmission from CH to BS. Therefore, the network lifetime is prolonged to 484191 frames. For scenario 2 in Fig. 8, the lifetime of LEACH, RRCH, and SISR is 7960, 8780, and 17603 frames, respectively. The SISR numerically outperforms LEACH by 121.1% and 100.4% for RRCH. Note that the lines of LEACH and RRCH rapidly drop at their maximum number of frames. This indicates that both schemes offer balanced energy consumption among the nodes since all the nodes stay alive and die relatively simultaneously compared to the SISR. However, we need to point out the drop point occurs much earlier than in SISR.

V. CONCLUSION

In order to accomplish the elongation of the network lifetime, we have shown that dynamic cluster definition can reduce the power consumption, and then allowing the CH node to rotate through the nodes of the cluster can likewise prolong network life. With our scheme, it is possible for the nodes in the cluster with increased power loads to switch to a cluster that is more favorable.

In the simulation for comparative value, we examined two schemes against the one we proposed over three scenarios characterized by the amount of activity the nodes were detecting in their environments. These three environments were no detected activity, random activity, and full activity. We have shown how these three environments compare with the three network management schemes. We believe that the proposed scheme we have developed is significantly better than the other two schemes, and also believe that it has sufficient merit to be considered as the scheme for managing such WSNs.

REFERENCES

[1] W. B. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proc. of the 5th ACM/IEEE MCN*, pp. 174–185, August 1999.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. of the 6th ACM/IEEE MCN*, pp. 56–67, August 2000.

[3] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, 1(4):660–670, October 2002.

[4] D. Nam, and H. Min, "An Energy-Efficient Clustering Using a Round-Robin Method in a Wireless Sensor Network," *Proc. of the 5th SERA*, pp. 54–60, October 2007.

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[6] Y. Chen, and S.H. Son, "A fault tolerant topology control in wireless sensor networks," *Proc. of ACS/IEEE AICCSA*, pp. 57–I, January 2005.

[7] R. Ghosh, and S. Basagni, "Napping backbones: energy efficient topology control for wireless sensor networks," *Proc. of IEEE RWS*, pp. 611–614, January 2006.

[8] W. P. Chen, J.C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, 3(3), pp. 258–271, July 2004.

[9] J. Y. Cheng, S.Ruan, R.Cheng, and T.Hsu, "PADCP: Power-Aware Dynamic Clustering Protocol for Wireless Sensor Network," *Proc. of IFIP WOCN*, pp. 1–6, April 2006.

[10] S. Yi, G. Park, J. Heo, J. Hong, G. Jeon, and Y. Cho, "Energy aware routing based on adaptive clustering mechanism for wireless sensor networks," *Proc. of International Conference on Networking and Mobile Computing*, pp. 1115–1124, February 2005.

[11] T. Rappaport, Wireless Communication, Principles & Practice. Prentice Hall, 1996.

[12] J. Baek, S. An, P. S. Fisher, and E. J. Jones, "Dynamic cluster header selection with self-incentive for wireless sensor networks," Proc. of IEEE Sarnoff Symposium, pp. 1–5, April 2009.

BIOGRAPHIES

Jinsuk Baek (M'04) is Associate Professor of Computer Science at the Winston-Salem State University (WSSU), Winston-Salem, NC. He is the director of Network Protocols Group at the WSSU. He received his B.S. and M.S. degrees in Computer Science and Engineering from Hankuk University of Foreign Studies (HUFS), Korea, in 1996 and 1998, respectively and his Ph.D. in Computer Science from the University of Houston (UH) in 2004. Dr. Baek was a post doctorate research associate of the Distributed Multimedia Research Group at the UH. He acted as a consulting expert on behalf of Apple Computer, Inc in connection with Rong and Gabello Law Firm which serves as legal counsel to Apple computer. His research interests include wireless sensor networks, scalable reliable communication protocols, mobile computing, network security protocols, proxy caching systems, and formal verification of communication protocols. He is a member of the IEEE.

Sun Kyong An is a software developer of Gabriels Technology Solution Inc, New York, NY. She received her B.S. degree in Computer Science from Winston-Salem State University, Winston-Salem, NC with a Magna Cum Laude in 2009. She participated in pseudo dual port 1T-RAM project with Msyslab Inc. as an assistant research engineer and Home Networking System project with LG Electronics as a software tester in Korea. He also worked for Wake Forest University Baptist Medical Center as an Intern. Her research interests include Wireless Sensor Networking, Mobile Computing, and Network Security.

Paul S. Fisher is R. J. Reynolds Distinguished Professor of Computer Science at the Winston-Salem State University (WSSU), Winston-Salem, NC. He is the director of High Performance Computing Group at the WSSU. He received his B.A. and M.A. degrees in Mathematics from University of Utah and his Ph.D. in Computer Science from Arizona State University. He has written and managed more than 100 proposal efforts for corporations and DoD involving teams of 1 to 15 people. He worked as consultant to the U.S Army, U.S Navy, U.S Air Force and several companies over the years. In the 1990's he commercialized an SBIR funded effort and built Lightning Strike, a wavelet compression codec, then sold the company to return to academe. His current research interests include wired/wireless communication protocols, image processing and pattern recognition.