

Multi-level Clustering Architecture and Protocol Designs for Wireless Sensor Networks

Barnabas C. Okeke
Ryerson University
Toronto, Canada
bokeke@ryerson.ca

K. L. Eddie Law
Ryerson University
Toronto, Canada
eddie@ee.ryerson.ca

ABSTRACT

Wireless sensor network (WSN) consists of sensors for measuring and gathering data in a variety of environments. These sensors, with the size and battery constraints, usually have limited transmission ranges due to the low-power wireless radio transceivers. In a sensor network, sensed data should be collected at a centralized location, called sink, for processing and analysis. With limited transmission distances, sensed data may require multiple relays to reach the sink. In this paper, a novel multi-level clustering (MLC) wireless sensor network design and its associated operating protocol will be presented. Energy optimization is always a critical factor in the designs and deployments of wireless sensor networks. The goal is to create an energy-efficient and effective routing protocol for the networks. Cluster creation in this paper is different from the well-known Low-Energy Adaptive Clustering Hierarchy (LEACH) design. Cluster-heads in our proposed design form a tree with a goal to reach all sensor nodes in a network. Subsequently, all sensed data in the tree can be delivered to the sink while LEACH can not offer this guarantee. Energy savings may be improved with different numbers of levels in the hierarchical clustering architecture. To validate the proposed design, thorough simulations have been carried out. Upon comparing to a multi-hop LEACH protocol, the proposed design offers consistent wider coverage area and longer life span of a wireless sensor network.

Keywords

clusters, energy model, LEACH, multi-level clustering (MLC), sensors, wireless sensor networks

1. INTRODUCTION

With rapid technological advancements in both the wireless and sensing devices, the creations of wireless sensor networks (WSNs) may find a broad spectrum of applications. Examples of applications may include border surveillance, environment monitoring, industrial process monitoring, tactical

systems, etc. In general, sizes of sensors are small and their deployment environments may prohibit the uses of external power supplies. It is likely that the operating life duration of a sensor is at most equal to the life span of its installed battery. Energy optimization is then a critical factor in system designs and deployments of wireless sensor networks. The more energy effective the system design is, the longer the operating life spans the sensor nodes and the associated wireless sensor network should be.

Signal transmissions in wireless communication medium may easily be corrupted with co-channel interferences and noises. This implies that signals in wireless medium usually suffer relatively high bit error rate (BER). In fact, with the physical size limitation and a reasonably accepted signal-to-interference-and-noise ratio (SINR), the transmission range of a sensor node is limited. It is unrealistic to consider that the data collecting and processing station, the sink, is always within the transmission ranges of all sensor nodes in a wireless sensor network. As a result, it is important to design a wireless sensor network architecture which can operate with an energy effective multi-hop data forwarding capability.

Given a fixed maximum transmission range, if a data transmitting sensor is close to a receiving sensor node, then transmitted data can be sent at a lower signal transmission power to save battery energy. The communication distance between a sender and receiver can be estimated through, for example, pilot signals in the system. Consequently, the transmission range can be adjusted by tuning to an agreed upon signal strength at the receiving node. This design technique is assumed and discussed in our system energy model which helps avoid wasting energy.

In the following, some related system designs for data forwarding in wireless sensor networks are discussed in Section 2. The Low-Energy Adaptive Clustering Hierarchy (LEACH) algorithm is a popular clustering design and its operating procedures are outlined. In Section 3, the energy model used in simulations is elaborated. Then the proposed multi-level clustering (MLC) system design and associated protocol operations are discussed. Thorough simulations have been carried out; corresponding results regarding the performance of the MLC system and protocol designs are elaborated in Section 4. The proposed MLC system design has demonstrated superior performance upon comparing to the multi-hop LEACH sensor network. Finally, conclusion can be found in Section 5.

2. RELATED WORK

Low-Energy Adaptive Clustering Hierarchy (LEACH) [4] is a popular energy-efficient group-based communications for wireless sensor networks. In LEACH, sensors are partitioned into groups called clusters, and time is partitioned into fixed-length intervals called rounds. Each round begins with a setup phase which includes three sub-phases: advertisement phase is for cluster-head selection, cluster set-up phase is for cluster member selection, and schedule creation phase is for setting up and notifying TDMA schedule for the cluster members.

At the beginning of the setup phase, each sensor may become a cluster-head with some predefined a priori probability. Cluster-heads broadcast messages to neighboring sensor nodes. If a node receives messages from multiple cluster-heads, it joins a cluster by choosing the one marked in a received message with the strongest received signal strength, i.e., the nearest cluster-head. During the round, cluster members send sensed data to their respective cluster-heads, which then aggregate, compress, and route the information to the sink directly. This clustering operation procedure repeats for every round. In general, cluster-heads usually consume more energies than cluster members. This design allows dynamic setup of clusters and cluster-heads in the process.

Clustering of sensor nodes has shown to be an effective technique for reducing the transmission distances of data sensing nodes. Many other LEACH-like protocol designs always have the same assumption, that is, all nodes can send data frames directly to the sink. Unfortunately, LEACH does not consider factors such as the residual energy of sensor nodes, dead nodes, and the relative locations of sensor nodes while choosing cluster-heads. This results in variation of the number of clusters [10, 12] from the computed optimal value as discussed in [3].

Dynamic Transmission Range Adjustment Protocol (DTRAP) [11] was another LEACH-like approach. In this design, each node effectively adjusted its transmission range to keep number of its neighboring nodes at a preset maximum value, which was independent of the node distribution. Hence, cluster sizes should not exceed the preset maximum neighbor size. At the end of each round, a cluster-head simply appointed another node with the highest residual energy within its cluster to be the cluster-head in the next round. This re-clustering technique reduced the number of control messages involved in repeating a full clustering selection process. However, this design was only effective when all nodes in network are within each others' transmission ranges.

In [9], comparisons between the effect of residual energies and relative positions of nodes in selecting cluster-heads were made. The simulation results in the paper showed that selecting cluster-heads with larger number of neighbors would be more important than those with more residual energies in creating energy-efficient clusters. However, neglecting the residual energy of a potential cluster-head might create *gray zones* in network. The death of a cluster-head within a round indicated no sensed information sent from that cluster to the sink.

In [5, 6, 7], the sink used genetic algorithm to coordinate the formation of energy-efficient clusters and the election of cluster-heads. In [5, 6], all nodes had identical properties and each node could be a cluster-head. In each round, a number of either 1 or 0 was assigned to each node. A node assigned with 1 was a cluster-head. It was a regular sensor node if assigned with 0.

Recalling that in LEACH design, an assumption is made that all sensor nodes can reach the sink directly. But in reality, all sensor nodes have limited transmission ranges, and sensed data may have to be relayed over multiple hops to reach the sink. As a result, some hierarchical clustering designs [1, 7, 8] have been developed.

In [7], a wireless sensor network maintained four nodal states in each genetic generation: **powered off**, **cluster-head**, **inter-cluster router** and **sensor node**. The inter-cluster routers formed a packet routing path to the sink for cluster-heads. They did not participate in the sensing operations nor did they belong to any clusters. In [1, 8], locations of sensor nodes were modeled as an independent homogeneous spatial Poisson process. The model was used to compute an optimal probability, P_0 , of a node of becoming a cluster-head as in the LEACH protocol. In [1], the cluster-heads elected themselves to a new level of cluster-heads with probability, P_1 and the process continued to at most the n -th level. Sensed data were fused at each level and forwarded to the upper level. The n -th level cluster-heads then sent directly to the sink. At the first level of the hierarchy, nodes forwarded cluster-head advertisement messages to their neighbors up to a predetermined at most k number of hops away. The proper values set for the n and k parameters were not clearly defined. This design was a bottom-up approach. Although it reduced the number of clusters forwarding signals to the sink, it did not work in cases that not all nodes can reach the sink. In this design, the multi-level cluster-heads might form a pyramid structure, the cluster-heads at the top of the pyramid might not be able to reach the sink.

In [8], the functionality of the normal single level cluster was extended with a proxy node. This proxy was used by a cluster-head when it did not have enough power to transmit the fused signal to the sink directly. The proxy node approach improved the efficiency of the network by ensuring that signal from cluster-heads could reach the sink.

3. MULTI-LEVEL CLUSTERING WSN

In this section, features from different designs may be integrated, and these are explored in our new design. In this paper, a novel top-down tree-based architectural design for creating multi-level clustering (MLC) wireless sensor networks is proposed. As shown in Fig. 1, the root of the MLC tree is always the sink of a wireless sensor network. Depending on the nature of an application, sensor nodes may be randomly and geographically located. And with limited transmission ranges of all sensor nodes and the sink, likely only some nodes can send data directly to the sink. Then in this group of sensor nodes, some of them can be picked to operate as the Level-1 cluster-heads. In other words, the sink is the lone Level-0 cluster-head, which is the node always at the highest level in the tree.

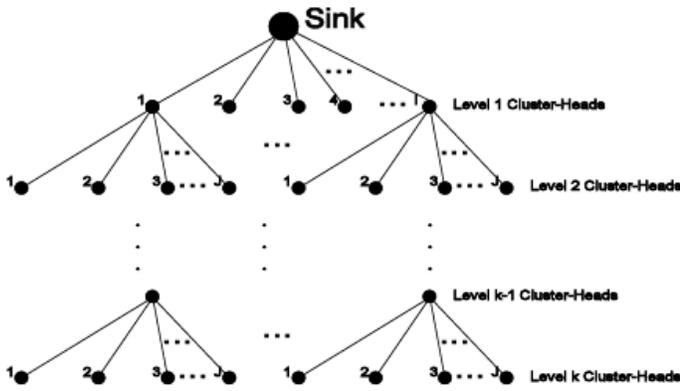


Figure 1: Multi-level cluster-heads.

Geographically, sensor nodes may be spread across a wide area. Multi-level clustering tree-based architecture can be constructed for relaying information from distant node to the sink. In the current simulation model, a Level-1 cluster-head is responsible for relaying, if exist, the information from the Level-2 clusters to the sink. Obviously, the traditional clustering techniques used in LEACH and its variants do not work effectively in networks requiring relaying. This is because the LEACH design is based on an assumption that all nodes can communicate with each others and the sink. But for most applications, it is almost impossible to make all nodes to reach the sink or every other node in the network. In such scenarios, LEACH-like protocols create *island clusters* which can be completely isolated, and form the *gray zones* in sensing environments.

The communications within a cluster in MLC protocol run in rounds. Each round consists of four operational procedures, and they are the **discovery**, **cluster-head selection**, **cluster-head member admittance**, and **transmission** processes. The first three processes repeat every round and they are followed with T number of transmission operations. The value of T is specified in the NOTIFY (Table 1) message sent by the sink, and it is subsequently copied into the PARAM (Table 2) messages forwarded down the tree by the cluster-heads. A combination of TDMA/CDMA schemes can be used to minimize inter- and intra-cluster interferences.

Table 1: NOTIFY message.

NODEID(1)	...	NODEID(I)	T	A	Z	F	J	CLUSTERID
-----------	-----	-----------	---	---	---	---	---	-----------

Table 2: PARAM message.

CLUSTERID	SLOT	SLOT_TIME	CODE	T	SIZE
-----------	------	-----------	------	---	------

3.1 Energy Model

Before going in-depth the MLC protocol design, we would like to elaborate the derived energy model in our simulations. The model is based on the one discussed in [3]. Power attenuation depends on the traveling distance, d , from a sender to a receiver. In free space or short distance without any obstacles, signal power follows the inverse-square law, $\propto 1/d^2$; but for longer distance with multi-path fading, signal power follows $\propto 1/d^4$.

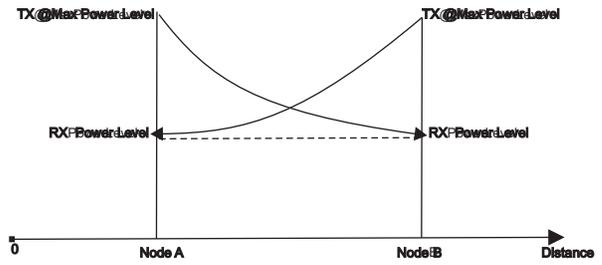


Figure 2: Neighbor distance estimation using the power level received from the neighbor's HELLO message

For the simulations carried out as described in Section 4, both Friss free space model and multi-path fading model are used based on the distance parameter, d . The received power $P_r(d)$ is

$$P_r(d) = \begin{cases} \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} & \text{for } d < d_{crossover} \\ \frac{P_t G_t G_r h_t^2 h_r^2}{d^4} & \text{for } d \geq d_{crossover} \end{cases} \quad (1)$$

where P_t is transmitted signal power, G_t and G_r are the transmitter and receiver antenna gains, λ is the wavelength of the carrier frequency, and $L \geq 1$ is the system loss factor not relating to propagation loss. The h_t and h_r are the transmitter and receiver antenna heights, respectively. And the crossover point of the two models is set at

$$d_{crossover} = \frac{4\pi\sqrt{L}h_t h_r}{\lambda}. \quad (2)$$

At a sensor node, energy $E_{tx}(l, d)$ is consumed for transmitting l -bit frame through a distance, d , and it is:

$$E_{tx}(l, d) = \begin{cases} l(E_e + d^2 \epsilon_{friss}) & \text{for } d < d_{crossover} \\ l(E_e + d^4 \epsilon_{two-ray}) & \text{for } d \geq d_{crossover} \end{cases} \quad (3)$$

where E_e is energy consumed per bit (J/bit) by the transceiver electronics, ϵ_{friss} is the free space power amplification coefficient in $J/bit/m^2$ and $\epsilon_{two-ray}$ is the multi-path fading power amplification coefficient in $J/bit/m^4$. Similarly, the energy $E_{rx}(l)$, consumed upon receiving l -bit frame, is:

$$E_{rx}(l) = l \times E_e.$$

If data rate is R_b , the transmit power P_t is:

$$P_t = \frac{E_{tx}(l, d)}{l} R_b. \quad (4)$$

Using Eqn. (1), (3), and (4), we can obtain parameters ϵ_{friss} and $\epsilon_{two-ray}$,

$$\epsilon_{friss} = \frac{P_r-sensitivity(4\pi)^2}{R_b G_t G_r \lambda^2} \quad (5)$$

$$\epsilon_{two-ray} = \frac{P_r-sensitivity}{R_b G_t G_r h_t^2 h_r^2} \quad (6)$$

With the parameter settings¹ used in the simulations, from Eqn. (5) and (6), we obtain: $\epsilon_{friss} = 6 \text{ fJ/bit}/m^2$ and $\epsilon_{two-ray} = 0.815 \text{ aJ/bit}/m^4$.

¹ $G_t = G_r = 1 \text{ m}$, $h_t = h_r = 1.5 \text{ m}$, $R_b = 38.4 \text{ kbps}$, $\lambda = 0.328 \text{ m}$, $P_r-sensitivity = -98 \text{ dbm}$ or 158 fW .

In general, a cluster-head consumes more energy than its members to execute as it administers, receives transmissions from all cluster members, and aggregate them into a frame for forwarding. If $E_{CH}(l, n)$ is the energy consumed by a cluster-head with n clustered members, then it is:

$$E_{CH}(l, n) = n \times [E_{rx}(l) + E_f] \quad (7)$$

where E_f is the energy required for the fusing operation. Furthermore, a cluster-head may need to relay fused information frames from cluster-heads at lower levels to the sink. Suppose that there are in total F fused data flows including the local cluster required forwarding. Then the total energy, $E_T(l, n)$ required by a cluster-head in one transmission process is:

$$E_T(l, n) = E_{CH}(l, n) + (F - 1)E_{rx}(l) + \sum_{f=1}^F E_{tx}(l, d_f),$$

where the d_f is the transmission distance for a cluster-head forwarding flow f . If a cluster-head always has energy to reach its NEXT_HOP cluster-head, then the d_f is constant throughout a transmission process.

In the design, the required minimum energy operating level, MIN_ENERGY of a node is fixed at twice the amount of energy required to transmit the maximum frame size at the maximum transmission range. It switches off itself when its residual energy is below this level. This ensures that a node, low in stored energy, does not get involved in the neighbor discovery process (Section 3.2) and end up polluting the NEIGHBOR_TABLE of other nodes. This is because such node with low energy level ends up sending HELLO message with a transmission power lower than what is required to reach the maximum transmission range. Such a situation causes those nodes, receiving the HELLO message, make incorrect estimates of the distances that this node is away from. Thus,

$$\text{MIN_ENERGY} = 2 \times E_{tx}(l, \text{TX_RANGE}). \quad (8)$$

where TX_RANGE is the maximum transmission range.

3.2 MLC Neighbor Discovery Process

In the proposed multi-level clustering protocol design, a sensor node selects a random number between $0-u$ to determine the number of seconds to wait before sending a HELLO broadcast message covering its maximum transmission range². u is set to 9 in simulations. This operation is to notify all neighboring nodes regarding the existence of this specific node. During the waiting period, a sensor node listens for neighboring nodes' discovery messages.

The fields in HELLO message is shown in Table 5. The NODEID field is the unique identity of the node sending the message. The ENERGY field indicates the current residual energy level of the node. The NODEID of a node is unique but the ENERGY level of a node decreases with time elapses. However, this condition does not apply to the sink, because the sink is assumed to have energy renewing source.

²A sensor node transmits at its maximum power to reach sensor nodes at its maximum coverage area with acceptable receive sensitivity. The coverage area of a node is modeled as a perfect circle.

Table 3: Important Node Properties

Property	Description
NODEID	The unique identity of a node.
ENERGY	Current residual energy of a node.
TX_RANGE	Maximum Transmission Range of a node. This value is the same for all nodes including the sink.
CLUSTERID	The unique identity of a cluster. This is equal to the NODEID of the cluster-head of a cluster.
NEXT_HOP	The next-hop cluster-head in the cluster-head's route to the sink.

Table 4: NEIGHBOR_TABLE

NODEID	ENERGY	DISTANCE	CLUSTERID
--------	--------	----------	-----------

Upon receiving a HELLO message from another node, a node registers the received NODEID and the latest ENERGY current in its NEIGHBOR_TABLE. The fields for each entry in the NEIGHBOR_TABLE is shown in table 4. If a HELLO with a new NODEID is received for the first time, it randomly picks a number between $0 - u$, and waits for that length of time before replying with a HELLO message. If the received HELLO message with a registered node, it simply updates the latest ENERGY in the NEIGHBOR_TABLE. The reason for replying with a HELLO message is to create a bidirectional link between the two nodes and secondly to update other neighbors of its current battery status.

Since HELLO message is sent with transmission powers to reach the maximum transmission ranges, a message receiving node can thus make an estimate of the DISTANCE, d , of the message sending neighbor. For example, from Fig. 2, the distance can be interpreted with the energy model discussed in Section 3.1. With the values of parameters used in simulations, the received power at the crossover point distance is $171pW$. With d 's set to TX_RANGE in Eqn. (1) and (4), we can obtain the DISTANCE, d , as

$$d = \begin{cases} \sqrt{\frac{P_t G_t G_r \lambda^2}{P_r L(4\pi)^2}} & \text{for } P_r > 171pW \\ \sqrt[4]{\frac{P_t G_t G_r h_t^2 h_r^2}{P_r}} & \text{for } P_r \leq 171pW \end{cases} \quad (9)$$

At the end of each round, every node including the sink purges its NEIGHBORS_TABLE and begins a new neighbor discovery process. This takes care the scenario that some nodes might have switched off due to low battery powers based on Eqn. (8).

3.3 MLC Cluster-head Selections

After the neighbor discovery process, a cluster-head can select cluster-heads at the next level. For example, at the Level-0, the sink picks I number of nodes from its NEIGHBOR_TABLE as the next level cluster-heads and broadcasts the NOTIFY message. The transmission power level of sending a NOTIFY message depends on the DISTANCE estimates of selected nodes recorded in the sink's NEIGHBOR_TABLE. The selected node with the largest DISTANCE, d , value is used for setting the transmission power of the NOTIFY message, that

Table 5: HELLO Message

NODEID	ENERGY
--------	--------

is,

$$\text{NOTIFY}_{tx_range} = \max_{1 \leq i \leq I} [\text{DISTANCE}]. \quad (10)$$

Upon receiving the NOTIFY message, a node firstly checks to see if its NODEID is listed in the message. It simply discards the message if it is not selected.

The design can control the transmission power upon sending the NOTIFY message and help conserve overall network energy. Certainly, reducing the number of nodes that receive the message may potentially extend the life of a node, but may also minimize the coverage area of the sensor network. We will explore the system performance with regard to the number of cluster-heads selected in the next level in later section.

A node is selected as a cluster-head using a *minimum cost model* which will be discussed in detail in Section 3.3.1. For example, at the Level-0, the sink selects the first I nodes with the lowest COST values. These I nodes form the Level-1 cluster-heads as shown in Fig. 1. If the sink has fewer than I neighboring nodes, then the sink may select all of them. In the protocol design, the sink has choices to require confirmation reply messages from these selected I nodes which indicating the acceptance or rejection of the task. The F field in the NOTIFY message is used for this purpose. When F is set to 1, a reply acknowledgement is required. When it is set to 0, the acknowledgement is not needed. An $F = 0$ also indicates that the selected nodes must accept the task of being cluster-heads. The CLUSTERID field is the NODEID of the node initiating the NOTIFY message.

A listed node, upon receiving the NOTIFY message, registers the CLUSTERID field value as the NODEID of its NEXT_HOP node to the sink. Then, it broadcasts a REQUEST message at its peak power to all its neighbors with the CLUSTERID field updated with its NODEID. That is, the CLUSTERID of a cluster-head is the same as its NODEID.

Notice that some neighbors of the Level-1 cluster-heads may include the neighbors of the sink. Therefore, for the nodes within this cluster-head's neighborhood that have not received any REQUEST, NOTIFY or ACK messages from any other nodes, they responds with REPLY messages. The NODEID in the REQUEST message is that of the cluster-head sending it. The NODEID and CLUSTERID fields in the REPLY message are the identities of the node replying it and the cluster-head that has initiated the REQUEST message, respectively. The REPLY message also contains the current ENERGY level of the replying node, and th cluster-head can update it properly in its NEIGHBOR_TABLE.

A Level- k cluster-head ($k \neq 0$) may select up to J number of next level Level- $(k+1)$ cluster-heads. The settings of both I and J parameters will be studied in Section 4. The value of J is specified in the J field of the NOTIFY message. Using the minimum cost model, J number of nodes are selected from the set of replied nodes, and a NOTIFY message is broadcast

Table 6: REQUEST/ADVERT Message

NODEID	TYPE
--------	------

TYPE	Message	Description
0	REQUEST	Used by a k th level cluster-head to send invitation for $(k + 1)$ level cluster-heads.
1	ADVERT	Used by a cluster-head to send invitation for member nodes.

Table 7: REPLY/JOIN/ACK/NACK Message

NODEID	CLUSTERID	ENERGY	TYPE
--------	-----------	--------	------

TYPE	Message	Description
00	ACK	Used by a node to accept the task of cluster-head.
01	NACK	Used by a node to reject the task of cluster-head.
10	REPLY	Used by a node to respond to a REQUEST message indicating that it is available to accept the task of cluster-head.
11	JOIN	Sent by a node to a cluster-head to indicate that it is willing to join its cluster.

with a power level determined based on Eqn. (10), where in this case I is changed to J . When the F field in the NOTIFY message is set to 1, an ACK or NACK message is replied to the cluster-head that has sent the NOTIFY message to the intended next level cluster-heads.

When a selected node finds its NODEID in the list within the NOTIFY message, it replies an ACK message to the cluster-head if and only if it has not overheard ACKs from other nodes from the time that it sent the REPLY message to the time that it received the NOTIFY message. This ensures that the selected cluster-heads are not concentrating in one location. Also, this design ensures that proper number of nodes are selected as members of k -th level cluster-heads by a $(k - 1)$ -th level cluster-head, as described in Fig. 1. This guarantees effective area coverage of a network that can reach the sink with a good number of cluster-heads.

If some nodes replied with NACK messages, a cluster-head then has to select another set of nodes equal to the number of receiving NACK messages using the cost model. The cluster-head then sends a NOTIFY message to them. Indeed, ACK and NACK messages may be returned by some or all the nodes again. This process continues until J number of nodes have accepted the task of next level cluster-heads or the list of replied nodes is exhausted. Notice that even though the J field is the same in all NOTIFY messages, the number of lower level cluster-heads created by each cluster-head in the same level is not always the same.

The newly selected k -th level cluster-heads repeats the same process of selecting $(k + 1)$ -th level cluster-heads until such a time that no more node replies to a cluster-head's RE-

QUEST message. At each level, every Level- $(k + 1)$ cluster-head registers the NODEID³ of its k -th level cluster-head in its NEXT_HOP field. This protocol design has an advantage of potentially creating a spanning tree that can forward all sensed data in a network through the tree to the sink consistently.

3.3.1 Minimum Cost Model

The COST(node) function of a node, node located at an estimated distance, DISTANCE from a certain cluster-head which is its neighbor is:

$$\text{COST}(\text{node}) = \frac{E_{ini}}{\text{ENERGY}(\text{node})} \quad (11)$$

where E_{ini} is the initial energy state of a node when it is firstly deployed. The ENERGY(node) is the latest residual energy level of the node. To simplify the computation, the numerator and denominator in Eqn. (11) can be switched, and the division can become a simple bit-wise shift operation; in this case, the node with the maximum residual energy should be picked.

Suppose that a Level- k cluster-head has sent out a REQUEST message, if G nodes out of the P neighbors of the cluster-head responded with REPLY messages, then J among these G sensor nodes can be selected to become the Level- $(k + 1)$ cluster-heads.

3.4 MLC Cluster Member Admittance

Once the cluster-head selection process has been completed, each selected cluster-head broadcasts an ADVERT message at its peak power to all its neighbors. Node then replies with a JOIN message to the cluster-head that it has received with the strongest signal strength. This likely should be the closest cluster-head. At the current implementation of simulation model, a cluster-head simply accepts all JOIN messages by sending node-specific ACCEPT messages to all joining nodes.

For the simulation model in future, the parameters A and Z in NOTIFY messages will be used to control the minimum and maximum member nodes a cluster-head can support. A cluster-head may reject some JOIN replies. With fewer member nodes than the preset minimum number, a cluster-head informs its current members to join another cluster-head. But the cluster-head cannot abandon its role unless its higher level cluster-head disappears in the tree. This technique may reduce the waiting time of cluster-heads with small member size before initiating aggregate message forwarding as described in the next section. Furthermore, it may improve the life span of the network. Smaller number of flows is expected to be forwarded by cluster-heads at the lower levels.

Cluster-heads should cooperate with the immediate upstream, immediate downstream, and same level neighboring cluster-heads to create orthogonal CDMA codewords for their respective clusters. Each cluster-head uses a specific codeword for all intra-cluster communications, while a dedicated codeword is used for cluster-head to cluster-head for multi-

³This is the CLUSTERID field in the NOTIFY message.

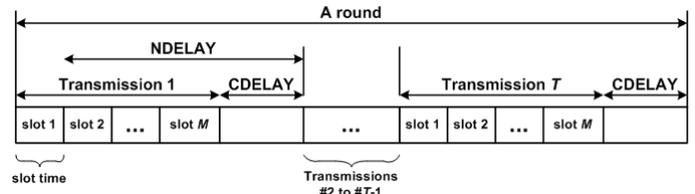


Figure 3: The transmission time line of a round

hop frame forwarding. This dedicated codeword should be known to all nodes in network.

After the establishments of intra-cluster codewords, cluster-heads set up TMDA schedules for all associated members and notify them through broadcast PARAM messages. The transmission power level of an intra-cluster communication is based on the Eqn. (10), where I is changed to SIZE, the membership size of the cluster. The CLUSTERID field in the PARAM message is the NODEID of cluster-head which the nodes use to update their respective CLUSTERID fields.

3.5 MLC Transmission Process

In a transmission cycle, each member node transmits its sensed data to its cluster-head in its allocated TDMA SLOT. SLOT_TIME is the time allocated to a member node by its cluster-head. The SLOT_TIME is constant in simulations because all member nodes are set to send identical l -bit data frame per transmission operation.

Once a cluster-head has received data from all its member sensor nodes, it aggregates the sensed data. It waits for a pre-assigned time until its NEXT_HOP cluster-head is ready to receive its aggregate data frame with the cluster-head to cluster-head dedicated codeword. Also with possibility, cluster-head may receive aggregate data from upstream cluster-heads towards the sink, and need to schedule them for forwarding. In the design, its own signal always takes precedence. In the simulations, all nodes and cluster-heads are assumed to operate at constant data rate for simplicity.

For a sensor node, the transmission process may involve T number of transmissions, which is specified in the T field of PARAM message in a round, r . As depicted in Eqn. (12), a node waits for NDELAY(node, r) after each frame transmission inside the round r before transmitting another frame as shown in Fig. 3, i.e.,

$$\text{NDELAY}(\text{node}, r) = (\text{SIZE}(h) - 1)\text{SLOT_TIME} + \text{CDELAY}(h) \quad (12)$$

where CDELAY(h) is the time required to propagate an aggregate signal from a cluster-head h to the sink, and SIZE(h) is the number of member nodes of h . The CDELAY(h) largely depends on the estimate of the cluster-sizes of all clusters along the route to the sink and number of flows on the route.

At the end of the T -th transmission, all nodes including the cluster-head purge the NIEGHBOR_TABLES and switch to the common codeword. A new neighbor discovery process begins which is followed by a new multi-level clustering process. This cycle continues until the time that the network is no longer functional. Recall that a node switches off when its

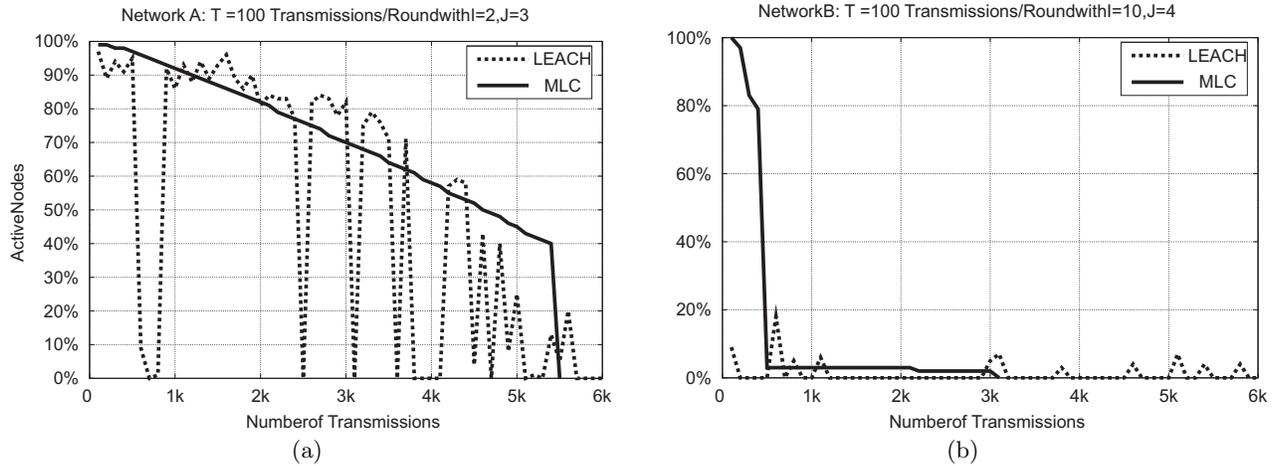


Figure 4: Change in active node percentage versus number of transmissions per round (time).

Table 8: Parameters for network A and network B.

Parameter	Network A	Network B
Total number of nodes, N	2000	200
Network dimension (in meters)	1000×1000	1000×1000
Node density (nodes/m^2)	0.002	0.0002
Average number of neighbor nodes	146	15
(x, y) coordinates of the sink (in meters)	(1000, 500)	(1000, 500)
Number of sink's neighboring nodes	63	4
TX_RANGE (in meters)	152.4	152.4

residual energy is lower than the MIN_ENERGY in Eqn. (8).

4. SIMULATION AND DISCUSSION

As shown in Table 8, two wireless sensor network models are used in simulations. The network A is a relatively dense network, while network B is a relatively sparse network model. In network A, a total of $N = 2000$ randomly generated sensor nodes are uniformly distributed across an $1000 \times 1000 m^2$ square-shaped area. The node density is $0.002 \text{ nodes}/m^2$. The maximum transmission radius is $152.4m$ [2], and an average number of neighboring nodes is 146 nodes. Similarly, in network B, a total of $N = 200$ sensor nodes are randomly generated and uniformly distributed on the $1000 \times 1000 m^2$ square-shaped area. The node density of $0.0002 \text{ nodes}/m^2$. With maximum transmission range of $152.4 m$, the average number of neighboring nodes is 15 nodes. In both simulation models, the sinks are located at the $(x, y) = (1000 m, 500 m)$ coordinates. The sinks in network A and network B have 63 and 4 neighboring nodes, respectively.

An initial energy of 2 Joules is assigned to each sensor node. Other parameters are based on the Mica2dot⁴ specifications [2]: $\epsilon_{friss-amp} = 6 fJ/bit/m^2$, $\epsilon_{two-ray-amp} =$

⁴Mica2dot is a third generation wireless micro-sensor node designed specifically for deeply embedded wireless sensor network.

$0.815 aJ/bit/m^4$, $E_e = 50 nJ/bit$, $E_f = 5 nJ/bit$, $G_t = G_r = 1 m$, $h_r = h_t = 1.5 m$, $L = 1$, $R_b = 38.4 kbps$, $\lambda = 0.328 m$, $d_{crossover}^5 = 87 m$, TX_RANGE = $152.4 m$, and $l = 4000 bits$.

In our simulations, a network is considered unusable if there are fewer than 10% sensor nodes alive, or no nodes can reach the sink. A sensor node switches off when its residual energy is less than the MIN_ENERGY stated in Eqn. (8), and it is considered dead.

In a round, a *live* node is on but may be *inactive*. An *active* node implies that its sensed data can always reach the sink. But an *inactive* node may not be connected to any cluster-head, and hence, does not participate in the sensing operations. Indeed, an *inactive* node can select to go into *sleep mode* and wake up at the end of the round. With Fig. 3, which shows the operating timeline of transmission operations, a node can estimate the length of time it can spend in the sleep mode.

In multi-hop LEACH-like algorithms designs, clusters may form but sensed data from the associated member nodes may not be able to reach the sink. This is because path simply can not be established to the sink. These clusters are *island clusters*, and the nodes within are considered *inactive*. That is, the highest level cluster-head in an island cluster does not have a NEXT_HOP cluster-head, and the cluster-head is not in the NEIGHBOR_TABLE of the sink. Island cluster is a worse scenario because energy has been consumed but wasted in sensing information, as the sensed signal does not reach the sink.

From the plotted results shown in Fig. 4, island clustering does not occur in the proposed MLC protocol. The proposed clustering technique forms a tree from the sink, and clusters are generated as branching points in the process. But the island clustering problem is noticeable in LEACH, and the effect is prominent in a relatively sparse network, as shown in Fig. 4(b). Due to the randomized creations of cluster-heads, multi-hop LEACH can not guarantee that a

⁵ $d_{crossover} = 86.2 m$ is a calculated value.

node within range of the sink is selected as cluster-head in a round. In sparse network, the probability of a node in the sink NEIGHBOR_TABLE becoming a cluster-head is much smaller.

Even if a few nodes within the range of the sink is selected, they might still be not enough to sustain the network for the entire period of a round. There are multiple occurrences that multi-hop LEACH protocol recorded zero number of transmissions in Fig. 4(a). There are cases that there are no cluster-heads in the neighborhood of the sink. The percentage of active nodes is zero in this case. As shown in Fig. 4(b), this happens often for the multi-hop LEACH protocol in network *B*. The simulated results indicate that the MLC protocol ensures good coverage of the network, and it makes good use of the sensor nodes within the transmission range of the sink.

Certainly, the performance of MLC protocol may depend on the *I* and *J* parameters. Simulations have been carried for different settings of *I* and *J*. As shown in Fig. 5 for both sparse and dense networks, *J* = 1 gives erratic results with higher variation in dense networks. In Fig. 5(a), smaller values of *I* perform better in dense networks. But in Fig. 5(d) and 5(f), larger values of *I* perform better in sparse networks.

Observing that Fig. 5(d) and 5(f) are identical despite using different values of *I*, this is because there are only 4 nodes in the transmission range of the sink (see Table 8) in network *B*. Based on the cost model used in Section 3.3.1, all neighboring nodes of the sink are Level-1 cluster-heads because $I \geq P$, where *P* is the number of nodes in the NEIGHBOR_TABLE of the sink. Hence, *I* = 4 is the only possible value used by the protocol for both Fig. 5(d) and 5(f) simulations.

In Fig. 6, the number of aggregate frames sent from active nodes received by sink is shown. The result further depicts that the multi-hop LEACH protocol is unreliable. In some rounds with large number of live nodes, no frame can arrive at the sink at all. The performance improvement is 17% in terms of the total number of frames received at the sink by the MLC protocol in network *A* upon comparing to the multi-hop LEACH protocol (see Fig. 6(a)). A 300% performance improvement is achieved in network *B* (see Fig. 6(b)) for the MLC design. The difference in the area under the graphs indicates that the MLC protocol guarantees better coverage and higher reliability in delivering sensed data to the sink.

5. CONCLUSIONS

In this paper, novel designs of multi-level clustering (MLC) architecture and protocol have been proposed for interconnecting sensor nodes in a wireless sensor network. The design is especially useful in WSN when some sensors do not reach the sink directly. With the energy constraints in WSNs, traditional wireless multi-hop routing protocols is not suitable. But the proposed design can create a highly interconnected tree graph in network, and the sensed data can get to the sink from far away. Simulation results indicate that the uses of other LEACH alike designs are not appropriate because majority of the frames generated by clusters can

not reach the sink. The proposed design offers a sensor network a wide-area coverage, and generated frames can reach the sink. It is our goal to create a spanning tree structure that all sensor nodes can reach the sink.

6. ACKNOWLEDGEMENTS

The authors would like to make an appreciation to Dr. Sajid Hussain at Acadia University, Nova Scotia, Canada, who made the simulation tool for Wireless Sensor Networks available and allowed us to extend and develop the research work carried out in this paper.

7. REFERENCES

- [1] S. Bandyopadhyay and E. J. Coyle. An energy-efficient hierarchical clustering algorithm for wireless sensor networks. In *Proc. IEEE INFOCOM'03*, 2003.
- [2] Crossbow Technology, Inc., http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2DOT_Datasheet.pdf. *Mica2dot Datasheet*, document part number: 6020-0043-05 rev a edition, June 2008.
- [3] W. R. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. Ph.d., Massachusetts Institute of Technology, Cambridge, MA, June 2000.
- [4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *33rd ICSS*, January 2000.
- [5] S. Hussain, A. W. Matin, and O. Islam. Genetic algorithm for hierarchical wireless sensor networks. *Journal of Networks*, 2(5):87–97, Sept. 2007.
- [6] S. Jin, M. Zhou, and A. S. Wu. Sensor network optimization using genetic algorithm. In *7th WMSCI'03*, 2003.
- [7] R. Khanna, H. Liu, and H. Chen. Self-organization of sensor networks using genetic algorithms. *IEEE ICC '06*, 8:3377–3382, June 2006.
- [8] K. T. Kim, H. Kim, and H. Y. Youn. Optimized clustering for maximal lifetime of wireless sensor networks. In *EUC Workshops 2006*, pages 465–474, 2006.
- [9] H. S. Lee, K. T. Kim, and H. Y. Youn. A new cluster head selection scheme for long lifetime of wireless sensor networks. In *ICCSA 2006*, pages 519–528, 2006.
- [10] H. Liu, L. Li, and S. Jin. Cluster number variability problem in leach. In *Lecture Notes in Computer Science*, 2006.
- [11] J. Liu and C. R. Lin. Energy-efficient clustering protocol in wireless sensor networks. *Ad Hoc Networks*, 3, 2005.
- [12] Q. Wang, K. Xu, H. Hassanein, and G. Takahara. Swatch: A stepwise adaptive clustering hierarchy in wireless sensor networks. In *Proc. 4th IFIP-TC6 International Conferences on Networking*, 2005.

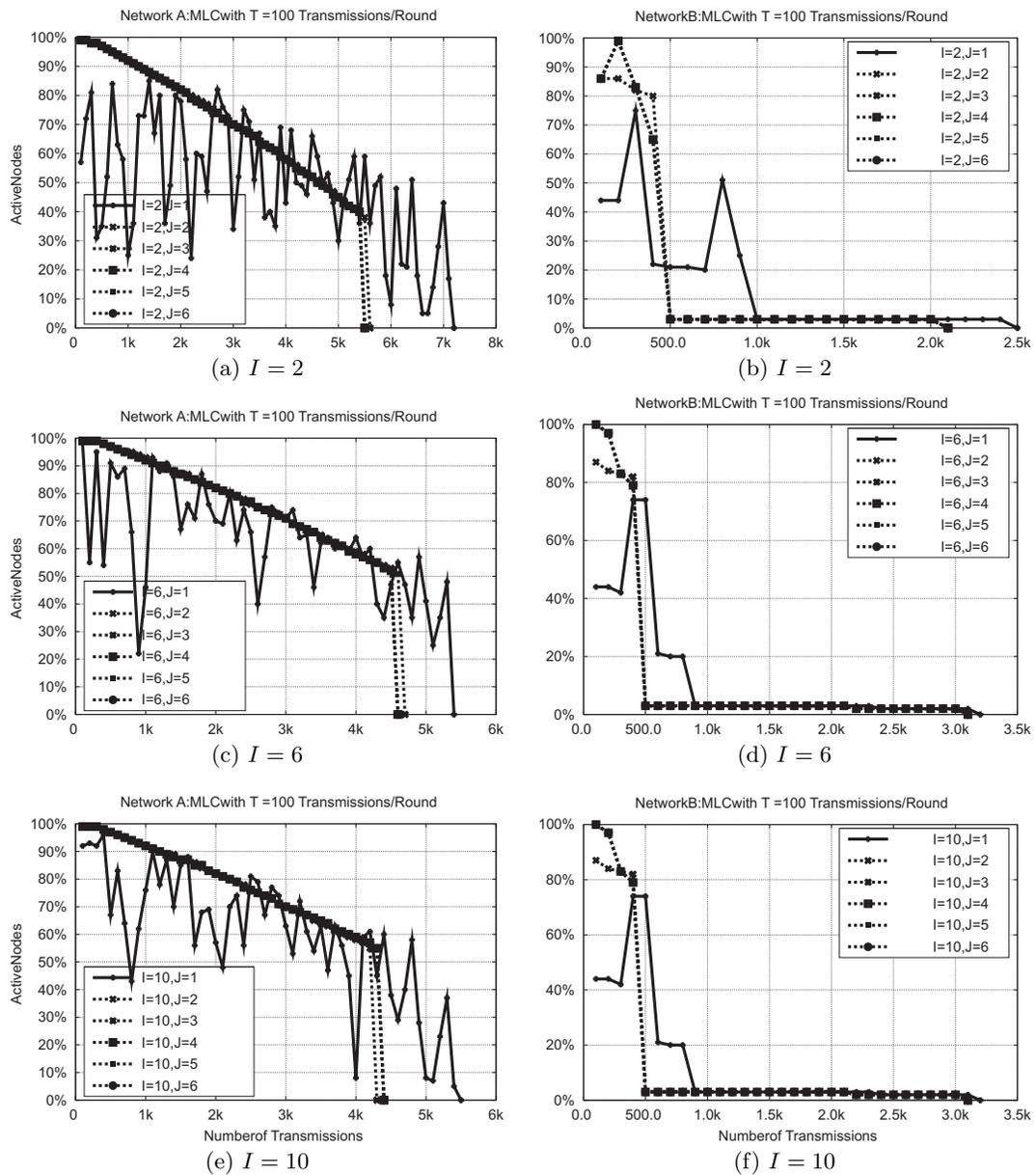


Figure 5: MLC WSNs with different settings of I and J , $T = 100$ in simulations.

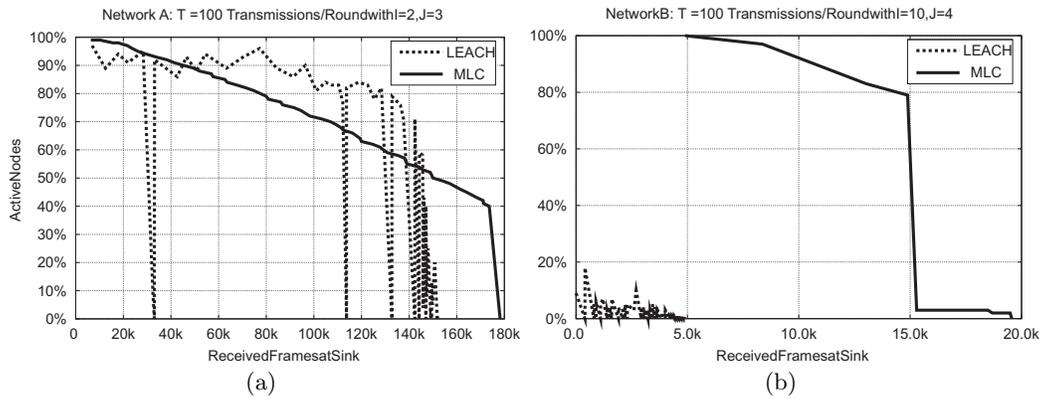


Figure 6: Frames received at the sink versus amount of active nodes in %.