Research Article

Robust and Energy-Efficient Data Gathering in Wireless Sensor Network

Juan Feng, Baowang Lian, and Hongwei Zhao

School of Electronic Information, Northwestern Polytechnical University, Xi’an 710072, China

Correspondence should be addressed to Juan Feng; fengjuankh@hotmail.com

Received 26 May 2014; Accepted 3 September 2014; Published 22 October 2014

Academic Editor: Yonghe Liu

Copyright © 2014 Juan Feng 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.

Robustness and energy efficiency are critical for sensor information system, in which an abundance of wireless sensor nodes collects useful data from the deployed field. The chain-based protocols (like PEGASIS [Lindsey and Raghavendra, 2002]) are elegant solutions where sensor node has high energy efficiency. Unfortunately, if one node in the chain is failed due to some reasons such as energy exhaust, then the information cannot be forwarded to the sink. To improve system robustness and balance the energy consumption, this paper proposes a robust and energy-efficient data gathering (REEDG) approach, which is an improvement over the chain-based and grid-based network structures, in sensor information collecting system. In REEDG, data gathering is executed by a data transmitting chain which is composed by a series of virtual grids. Each grid communicates only with its neighbor grid and takes turns transmitting the information to the base station. Furthermore, an adaptive scheduling scheme is proposed to trade off energy consumption on each node and data forwarding delay. Experimental results show that, when compared with state-of-the-art approaches, REEDG achieves network lifetime extension of at least 13% as measured in terms of 20% dead nodes and improves the data transmission ratio at lowest 24% as 20% nodes fail.

1. Introduction

Information collection is one of the most important applications of wireless sensor network (WSN) which consists of a large amount of small, low-cost, and wirelessly connected sensor nodes deployed in an unattended natural environment. In an information collection system, the sensor nodes collaboratively monitor the natural environment in the area of deployment and gather the sensed data such as audio and seismic. Since the sensor nodes are usually battery-powered and have a limited lifetime, it is infeasible to replenish energy via replacing their battery after deployment. Thus, both energy efficiency and robustness are the critical issues in the design of information collection WSN.

For the data collection and gathering, the simplest approach is that all the nodes send their data directly to the sink. Since the sink is always located far away from the deployment area, the transmitting energy from any node to the sink is high and nodes will drain out quickly. Therefore, some improved approaches are proposed to reduce the transmission data flow to the sink as few as possible. Clustering (such as LEACH [1]) is one of the energy efficient approaches for data gathering. In clustering WSN, cluster head is responsible for the data gathering from its members. However, if sensor nodes report their data to the respective cluster head, more than one cluster head is involved to transmit data to the sink, leading to multiple data transmission flows toward the sink. Moreover, the structure of cluster needs to be reformed periodically. To improve it, chain-based approaches (like PEGASIS [2]) are proposed. In chain-based data gathering, all nodes are organized into a chain by using a certain algorithm and then take turns to act as the chain leader which is responsible for forwarding the aggregated data to the sink. These approaches outperform the cluster-based approaches by eliminating the overhead of dynamic cluster formation and minimizing the number of long-distance transmissions. Unfortunately, by using the chain-based protocols, the robust problems arise. If a node in the chain dies, the sensed data cannot be forwarded to its downstream neighbor until the chain is reconstructed. In that case, the sensed data cannot arrive at the sink in time and the energy dissipation required for frequent reconstructing
the chain is high. Hence, none of the hierarchical nodes arrangement has achieved all goals of data gathering, high energy conservation, and high robustness.

To address these problems and achieve the aforementioned goals, we propose a robust and energy-efficient data gathering (REEDG) approach for information collecting WSN. In REEDG, the entire sensing field is first divided into virtual grids by using a certain algorithm. The virtual grid structure is geographically formed and it does not change over time. For example, this structure is well applied in the literature [3–5]. After that, each grid is regarded as a unit to form a data transmitting chain. One node in a grid takes the role of collection leader (CL) that gathers the sensed data from its grid members and it takes turns to act as the transmitting chain leader in order to forward the aggregated data to the sink. For each round, every CL receives the data from its grid members and upstream neighbor CL and then passes its aggregated data via its downstream neighbor CL until the designated chain leader. Each grid is regarded as a chain member; thus one node dies, and another one in the same grid is selected to replace the role of the previous one, so that the structure of the chain does not need to be restructured. As a result, it reduces the energy consumption of chain constructing so as to prolong the network lifetime. Furthermore, an adaptive sleep scheduling is proposed to keep a minimal number of sensor nodes active and guarantee the performance of data gathering. In data collecting, it is not necessary to have sensor nodes active all the time. Instead, waking up nodes which have sensed data to transmit and putting the other nodes into sleep mode are an efficient way to save energy. In summary, the contributions of this paper are three-fold.

(1) An efficient data gathering approach is proposed. A virtual grid instead of one single node is used to form the transmitting chain. In this way, the sensed data can be well aggregated on each CL and only one data flow along CLs is transmitted to the sink. After the data transmitting chain forming, the network structure does not need to reform during the data gathering period.

(2) A network combining the grid-based and chain-based framework is built. By considering the whole grid as a chain member, any sensor node in the same grid can play the same role in the transmitting chain when the previous one dies so that data gathering is not affected by the failure of some nodes.

(3) Adaptive sleep scheduling scheme is proposed to allow different grids having independent sleep schedules to maximize the sleep time of nodes. Moreover, the collaboration is established among grids leading to less communication cost because the control packets are just sent among CLs instead of all the nodes.

The rest of this paper is organized as follows. Section 2 gives an overview of related work. In Section 3, the system model is stated. Then, Section 3 presents the proposed approach in detail. The experimental results are shown in Section 4. Finally, we conclude the paper in Section 5.

2. Related Works

To achieve energy-efficient and robust data gathering, some approaches have been proposed one after another. The data gathering techniques vary with different network architectures. In general, there are mainly four network architectures for data gathering. They are cluster-based, tree-based, chain-based, and grid-based networks [6].

For cluster-based WSNs, a typical and classical data gathering scheme is LEACH [1], in which the network is divided into a few clusters and only cluster heads send the aggregated data to the remote sink directly. Moreover, TDMA is utilized to minimize channel contention. Although LEACH achieves the energy efficiency to some extent, the cluster structure needs to be reformed periodically to prevent the early death of cluster heads, which consumes much energy. After LEACH, many improved cluster-based protocols have been proposed for such applications. In [7] an adaptive scheme is proposed to control the degree of data aggregation with respect to the reliability requirement. In [8, 9] the authors propose to build a hierarchy for all clusters by flooding in a typical route discovery process [8] or by using a greedy heuristic [9]. The sensed data are first aggregated on cluster heads at the lowest level. Then the aggregated data are sent to a higher-level cluster head for further aggregation. In [10], the authors propose to aggregate sensed data hop by hop through a multihop path. Using this scheme the route must be established in advance. However, the way of data gathering is not robust because the sensed data cannot be collected if cluster head failed. In all cluster-based approaches the cluster hierarchy needs to be reformed periodically and dynamically. Moreover, the similar sensed data are always sent to different cluster heads not to facilitate data aggregation.

A tree can be simply used to organize the nodes of a sensor network into a hierarchy. For tree-based WSNs, TAG, for instance, is a good representative of this kind of schemes [11]. Nodes in TAG are logically organized into levels depending on their distance to the sink so that a transmission from the leaf nodes to the sink is completed within an epoch. Unfortunately, using a tree brings a high probability of disconnections in the network because the closer the distance between the node and the sink is, the heavier the workload of the node is. An obvious solution would be retransmitting lost data using alternative paths. Multipath-based approaches are proposed in several works [12, 13]. Each node is allowed to have more than one parent and to exploit all of them during data gathering phase. These approaches are obviously more robust and under certain assumptions it may be not much more energy intensive than the single path one. However, since each node has more than one parent, there is duplication of information at each level, which makes it difficult to implement duplicate sensitive aggregation function, such as computing the average or the count.

For chain-based WSNs, PEGASIS is an improved data gathering protocol. It further decreases the consumed energy by minimizing the transmission distance among sensor nodes. After PEGASIS, chain-based sensor networks have been used widely in recent works [14–18]. Jung et al. proposed an enhanced PEGASIS [14], in which the sensing area is...
circularized into several concentric levels. For each level, a chain is constructed based on the greedy algorithm. PEGASIS slightly improves the redundant transmission; however, since the distribution of sensor nodes is not even, the transmission distance between two chain leaders in different levels might be lengthy, and the data transmission consumes more energy. In [15, 19], the main idea is to split the sensing field into a number of smaller areas in order to create multiple shorter chains to reduce the data transmission delay and redundant path. COSEN in [19] is efficient in the ways that it ensures maximal utilization of network energy and it takes much lower time to complete a round. However, for large sensing areas, the chain in each smaller area would still be lengthy. In [16], the authors propose an efficient localized chain construction (ELCC) scheme, which creates several chains for the topology using Voronoi tessellation. To construct a chain in a Voronoi cell guarantees that the summation of square of the distances would be the lowest. In [17], the authors focus on building a chain to get the minimum transmitting energy. In [18], the authors establish a chain in order to meet the requirement of the network coverage. Comparing with cluster- and tree-based architectures, the clear superiority of chain is that it is not necessary to reform the structure when the head node is changed. However, if all the nodes attend one chain, the sensed data are transmitted through a long and redundant path. If sensor nodes form more than one chain, the data aggregation among chain heads is not facilitated to implement. Furthermore, the network does not have high robustness. If one node in a chain dies, the chain structure has to be reform.

For grid-based WSNs, TTDD algorithm [20] lets sensor nodes at the grid points (called dissemination nodes) forward the data. The sink issues a query for the data. The query is further propagated only by the dissemination nodes and the source responds back through the reverse path. However, considerable overhead would be involved in the dissemination nodes. A variant of TTDD is called energy-efficient data dissemination (EEDD) algorithm [21], in which the grid heads have to be frequently changed in order to maintain fairness for each sensor node. However, in existing grid-based network [3], the data of sensor node in each grid are transmitted by their own grid head through different path so that the sensed data cannot be efficiently aggregated among the different grids. In contrast, in our REEDG, each grid is regarded as a chain member so that the sensed data in different grids will be aggregated by transmitting through the chain.

3. System Model

3.1. Network Model. In this paper, we consider a static WSN which is composed of one sink and \( p \) randomly distributed sensor nodes \( n_i \), \( i \in [1, p] \) in a two-dimensional sensing field, where \( p \) is the number of the deployed nodes. Our sensor network model has the following properties and assumptions.

(i) The sink is fixed and located far from the network. It has an infinite power supply and gathers data from the sensing field.

(ii) The distribution of sensor nodes is mutually independent. Every node is homogeneous and energy constrained.

(iii) Each node knows its position by any localization algorithm (such as [22]). Let \( X_i(x_i, y_i) \) be the location of node \( n_i \).

(iv) Data collection is periodical from the network. In each round of communication, each sensor node has a packet to be sent to the sink.

3.2. Energy Consumption Model. We adopt a very widely used energy model [1, 4, 18] as described in Figure 1. \( E_{\text{TX}}(k, d) \) and \( E_{\text{RX}}(k) \) represent energy consumption of transmitting and receiving \( k \) bits data over a distance \( d \):

\[
E_{\text{TX}}(k, d) = (E_{\text{TX-elec}} + e_{\text{amp}} \cdot d^\alpha) \cdot k, \quad (1)
\]

\[
E_{\text{RX}}(k) = E_{\text{RX-elec}} \cdot k, \quad (2)
\]

where \( E_{\text{TX-elec}} \) and \( E_{\text{RX-elec}} \) are distance-independent terms that take into account overheads of transmitter and receiver electronics. \( e_{\text{amp}} \) (Joule/(bit \cdot m^3)) is a constant which represents the energy needed to transmit one bit to achieve an acceptable signal to noise ratio over a distance \( d \), and \( \alpha \) is path loss exponent (2 \( \leq \alpha \leq 5 \)) which depends on the channel characteristics. Generally, we can assume that \( E_{\text{TX-elec}} = E_{\text{RX-elec}} = E_{\text{elec}} \).

4. Robust and Energy-Efficient Data Gathering

Our idea is based on a hybrid network structure, in which the working process can be divided into three stages, network building, data collection, and network maintenance. In network building stage, the network is initialized to form a grid-based network structure and the data transmitting chain is built. Then in data collection stage, sensing data are collected along the chain. Each grid has adaptive schedule and plays a role of chain head in turn. At last, network maintenance includes grid maintenance, new nodes addition, and dead nodes exit. The following subsections present the working process of the proposed approach in detail.

4.1. Network Initialization. Initially, the network is organized as in Figure 2, in which the whole sensing field is divided...
into q small equal size grids \( G_j, j \in [1, q] \). The grids are differenced by their grid identifications: \( \text{Id}(G_j) = (u_j, v_j) \). Each node decides which grid it is in by its location \( X_i(x_i, y_i) \) as follows:

\[
u(n_i) = \frac{x_i - x_0}{\gamma}, \quad v(n_i) = \frac{y_i - y_0}{\gamma}, \quad (3)
\]

where \([\cdot]\) is a symbol which stands for the integer part of the number in it. \((x_0, y_0)\) is the location of the network origin, which is a system parameter set in the network initialization stage. \( \gamma \) is the side length of each grid. The nodes in one grid have the same grid identifications.

Usually, the sensing range of node \( n_i \) can be approximated by a circle with centre \( n_i \) and radius \( R_s \). In order to collect sensed data, all the nodes are divided into three roles, which are collection member (CM), collection leader (CL), and transmitting chain head (CH). CM reports its sensed data to collection leader in its own grid. CL aggregates the data received from its members and sends it to the neighbour collection leader. Then, CH sends the aggregated data to the sink. These three node roles can be transferred as illustrated in Figure 3. Accordingly, the transmission ranges of a node have three states: (1) \( R_{ts} \), transmitting data within one grid, (2) \( R_{ts} \), transmitting data to the neighbour grid, and (3) \( R_{tu} \), transmitting data directly to the sink. Each sensor node has power control and the ability to adjust its transmission range on the basis of its roles.

We assume that all nodes have location information about the nodes in their neighbor grids. If not, our proposal still works, and nodes can obtain the information by communicating with each other. Let \( W = (N, CL) \) denote a sensor network, where \( N \) is the set of all the sensor nodes and CL is the set of all the collection leader nodes. Thus, one grid \( G_j \) in the sensing area can be shown as follows:

\[
G_j = \{ n_i | u(n_i) = u_j \land v(n_i) = v_j \land n_i \in N, \quad i \in [1, p], j \in [1, q] \}.
\]

Furthermore, each node can adjust transmission power based on its undertook role. Nodes can estimate its transmission range \( R_{tg} \) as follows:

\[
R_{tg} = \{ d | d = \max \{ n_i, n_j \} \land n_i, n_j \in G_j, \quad i, j \in [1, q] \},
\]

where \([n_i, n_j]\) is the distance between node \( n_i \) and \( n_j \). Similarly, \( R_{tn} \) and \( R_{tu} \) are estimated:

\[
R_{tn} = \{ d | d = \max \{ n_i, n_j \} \land n_i, n_j \in G_j \land n_g \in G_k, \quad i, j \in [1, q] \},
\]

\[
R_{tu} (n_i) = \{ d | d = | n_i, s \} \land n_i \in N, i \in [1, p] \},
\]

where \([n_i, s]\) is the distance between node \( n_i \) and the sink. Therefore, each sensor node can choose appropriate transmission range and power for energy efficiency and reliable data transmission.

4.2. Selecting Collection Leaders. All the sensor nodes start with CM and the transmission range of nodes are set as \( R_{tg} \). To balance the energy consumption on each node, the node which has the most energy in a grid is selected as the CL by Algorithm 1, in which node \( n_i \) broadcasts an announcement “req.CL” after waiting for a time period \( T(n_i) \):

\[
T(n_i) = T_{min} + (T_{max} - T_{min}) \left( 1 - \frac{E_{res}(n_i)}{E_{ref}} \right) + T_{ran}(t),
\]

where \( T_{min} \) and \( T_{max} \) denote the minimum and maximum waiting time used to control the waiting time in a reasonable range. \( E_{res}(n_i) \) is the residual energy of node \( n_i \). \( E_{ref} \) is reference energy in order to avoid too long waiting time when all the nodes have a little energy left. \( T_{ran}(t) \) is a random time in \([0, t]\), which is usually an order of magnitude smaller than \( T_{max} \), and added to prevent the collision caused by the same residual energy of two or more nodes. Thus, sensor nodes can have the same or different initial energy. When the waiting time \( T(n_i) \) expires, the node \( n_i \) broadcasts the announcement “req.CL” to other nodes. If any node receives the announcement from the node which has the same grid \( \text{Id} \) with it, the node will cancel the waiting time and give up the election of CL to guarantee just one CL in one grid. In this algorithm, the node which has more energy will wait shorter time before broadcasting its CL announcement. Finally, the node which
4.3. Data Transmitting Chain Forming. When selecting CL finishes, sensor nodes adjust their sleep state, transmitting range, and power based on their roles. CM nodes can get into sleep and wake up after a time period. Then, the transmitting chain is constructed by CL nodes. According to the energy consumption model discussed in Section 3.2, to reduce the energy consumption in data transmitting we should shorten the distance between nodes or decrease the packet size. The data packets can be aggregated when they are transmitted through the chain to achieve energy saving. Moreover, the total cost of sending data throughout the chain depends on the distance $\sum d^\alpha$, where $d$ is the distance between two grids along the data transmitting chain. The distance between two grids is defined as follows:

$$ |G_j, G_k| = \sum_{i=1}^{\lambda_j} \sum_{g=1}^{\lambda_k} |n_i, n_g|^{\alpha} / \lambda_j \lambda_k, \quad n_i \in G_j, n_g \in G_k, \quad i, g \in [1, p], \quad j, k \in [1, q],$$

where $|G_J, G_k|$ is the average distance of the nodes in the grids $G_J$ and $G_k$. Algorithm 2 builds a chain with minimum $\sum d^\alpha$.

First, the CL which is farthest away from the sink is chosen to initiate the process of chain constructing (line 1). The CL is considered as the original CH and its grid is added to the chain as CH grid $G_{CH}$ (line 3). The CH creates a chain head token. Then, $cl_j$, which is not in the chain and its related grid $G_j$ is the neighbor of the grids which have been already in the chain, calculates the minimum increase of $\sum d^\alpha$ when $G_j$ is added to the chain (lines 6–8). After that, choose the grid which has the minimum increase of $\sum d^\alpha$ as the new chain head token. This process is repeated until all nodes have been added to the chain (lines 9–12). The algorithm ends when the number of nodes in the chain reaches $q$ (line 16).

Algorithm 2: Data transmitting chain construction.

1. $cl_j = \{cl_j||cl_j, s| = \max|cl_j, s|, cl_j \in CL, j \in [1, q]\}$
2. $n = 1$
3. $CH \leftarrow cl_j, C_n \leftarrow G_j, CHAIN \leftarrow \{G_j\}$
4. while ($n < q$)
5. $Min_V = Max_V$
6. for each ($cl_j \in CL - CHAIN, j \in [1, q])$
7. (if ($G_j \in \{neighbor grids of G_j, G_k \in CHAIN\}$))
8. do $V_j = \min \left\{ \left( \frac{G_j, C_i}{\alpha}\right)^2 + \left( \frac{G_j, C_{i+1}}{\alpha}\right)^2 - \left( \frac{C_i, C_{i+1}}{\alpha}\right)^2 \right\}, i \in [1, n]$
9. (if ($V_j < Min_V$))
10. $Min_V = V_j$
11. $next = j$
12. end if
13. end for
14. Insert (CHAIN, $C_i, G_{next}$)
15. $n = n + 1$
16. end while
In order to balance the energy consumption, grids take turns playing the role as CH grid \((G_{CH})\) and transmitting the grid is selected as the new chain member and the process is repeated until all the grids are in the chain. In the process, the chain can be broken to insert a grid for the shortest \(\sum d^r\). Figure 4 illustrates that a chain \(G_1-G_2-G_7-G_6-G_{12}\) is constructed in a deployed sensing field.

From the line 7 in Algorithm 2, we can see that the grids which are the neighbors of the current chain are just considered instead of all the grids when choosing the new grid to add to the chain. Since the grids are distributed by the geographical location, it is impossible that the minimum increase of \(\sum d^r\) is obtained when \(G_j\) is added to the chain and \(G_j\) is not the neighbour of the chain (as Figure 4 shows), so that a sensor node does not need to know location information from all nodes, especially from distant nodes.

4.4. Data Gathering through a Data Transmitting Chain.

After data transmitting chain forming, the network starts to collect information of the sensing area. The CH passes the control message through the chain to initiate the data transmission from the ends of the chain. The control message is just transmitted among the CLs instead of all nodes and it is just transmitted one time in a period of CH lifetime. Therefore, the control message costs very small. In each data collection round, CM reports sensed data to its CL in each grid at the time slot they negotiated. When the CL in the ends of the chain receives the control message from CH, it sends the aggregated data of its own grid to the downstream neighbour CL in the chain so that the sensed data is forwarded to CH along the opposite direction of the control message. Moreover, the sensed data are aggregated at every CL. Each CL aggregates its grid’s data with the received data from its upstream CL to generate a single packet of the same length. At last, CH sends the aggregated data to the sink.

Figure 5 takes a part of network from Figure 4 as an example. In Figure 5, \(cl_1\) is considered as CH and the control message is passed along the chain until to \(cl_{11}\). CL gathers the sensed data from its CMs, respectively. When \(cl_{11}\) receives the control message, it sends the aggregated data of its grid to \(cl_6\). Then, \(cl_6\) aggregates the received data with its own grid’s data and sends it to \(cl_1\) until \(cl_{11}\). Finally, \(cl_1\) sends the aggregated data to the sink.

Since the radio module is the most energy consuming part in a sensor node, the nodes which do not need to send data should shut off their radio and go into sleep state. In this paper we propose an adaptive sleep scheduling scheme based on the data transmitting chain for the information collection system. Each grid has different sleep scheduling and the global synchronization is not required. The timeline of the information collection is shown in Figure 6. CH assigns time slots for CL in every grid by the control message passing in order to send data along the chain by CL. The time period of a data collection round is assumed as \(t_r\), which is divided into \(t_c\) and \(q\) time slots. \(t_c\) is the time interval when CL collects data from CMs in the grids of the end of the chain. One time slot \(t_s\) can be obtained as follows:

\[
t_s = \frac{t_r - t_c}{q}.
\]

In a data collection round, the CLs in the end grids of the chain have already acquired the sensed data from their CMs after time period \(t_c\). Then the CL aggregated the data and sends it to its downstream CL at the next time slot. Since the CLs in the end grids of the chain just have one neighbor to send data, one time slot is assigned to it, while the other CLs are assigned two time slots. One time slot is used for receiving data from its upstream CL and the other is used for sending data to its downstream CL. There is one overlapped time slot (like in Figure 6) in two CLs which are the neighbor grids in the chain so that CL can communicate with its neighbor grids. The information of the assigned time slots for CL is included in the control message which is passed through the chain by CH. In each grid, CL is assigned an appropriate time slot during \(t_{ic}\) (time of information collection as in Figure 6) for its CM to collect sensed information because CM should report its sensed data before its CL sends data to the downstream CL. The CM in different grid can collect information in different time periods and short distance communications are used within a grid so as to increase the channel reusability and the network throughput. Furthermore, the nodes without sending tasks can get into sleep, which can reduce energy cost significantly.

In order to balance the energy consumption, grids take turns playing the role as CH grid \((G_{CH})\) and transmitting the
aggregated data to the sink. If the lifetime of $G_{CH}$ elapses, the
CL in $G_{CH}$ sends the CH token to its downstream grid, which
is converted to head grid of the chain. CH grid transformation
is shown in Figure 7. The lifetime information of $G_{CH}$ is also
included in the control message. When a grid is considered
as CH grid, the CL in the grid creates a control message and
passes it along the chain. Therefore, each grid knows which
one is the CH and how long the CH lifetime is. The lifetime
of $G_{CH}$ is calculated by the CH as follows:

\[
lt_{CH,grid} = \sum_{n \in G_{CH}} lt_{CH}(n_i),
\]

\[
lt_{CH}(n_i) = \left[ \delta_{min} + \delta_{d,ref} \max \left[ n_i, s_i^{\alpha} \right] + \delta_{e,ref} \frac{E_{res}(n_i)}{E_{ref}} \right],
\]

where $lt_{CH,grid}$ is the lifetime of $G_{CH}$ and $lt_{CH}(n_i)$ is the
lifetime of node $n_i$ as CH, $lt_{CH,grid}(n_i) \in [0, 1, 2, \ldots]$. Symbo
$[\ ]$ stands for the integer part of the number in it. $\delta_{min}$,
$\delta_{d,ref}$, and $\delta_{e,ref}$ are the minimum lifetime value, distance
related reference, and energy-related reference, respectively,
$\delta_{min}, \delta_{d,ref}, \delta_{e,ref} \in [0, 1, 2, \ldots]$, which can be set based
on requirement. Since the transmitting cost is proportional to
distance in the light of (1), the node which is farthest away from
the sink is chosen as reference because it has to dissipate the
maximum amount of energy during its turn as CH compared to
the other nodes. The lifetime of $G_{CH}$ is the sum of the
lifetime of each node as CH in $G_{CH}$. The node in $G_{CH}$ is
considered as CH one after another according to its CH
time $lt_{CH}(n_i)$ until every node in $G_{CH}$ has taken the role
of CH during one $G_{CH}$’s turn. When the current CH finishes
its lifetime, it will broadcast a declaration to the other nodes
in $G_{CH}$. The node received the declaration and its $lt_{CH}(n_i) > 0$
will wait for a short random time to avoid collision and
broadcast a reply to the nodes in $G_{CH}$ to declare its CH
role. If the current CH does not receive a reply after a time
period $t_{CH}$, it passes the CH token to the downstream grid.
When a grid is converted to $G_{CH}$ from a general grid, the
nodes in $G_{CH}$ will calculate $lt_{CH}$ and $lt_{CH}(n_i)$ again. Moreover,
during a period of $lt_{CH,grid}$, the control message just needs
to pass through the chain one time so that the transmission
costs of control message be very small. Equation (II) allows
the individual grids to become CH grid variable number of
times depending on their distances from the sink and residual
energy, which not only reduces control message transmission,
but also leads to more balanced energy costs.

When grids receive the control message from the CH, CLs
in the grids adjust the sleep state for their CMs and send the
aggregated data in the assigned time slot. Each time CLs send
data, they set $lt_{CH} = lt_{CH} - 1$. If $lt_{CH} = 0$, they wait for the new
control message from a new CH.

4.5. Grid Maintenance and Robustness Analysis. To avoid
excess energy consuming on CL, a lifetime of CL collecting
data from its CMs is set in the CL announcement message
when broadcasting it to its CMs. The $cl_j$ lifetime value is
mainly dependent on its residual energy and calculated as follows:

\[
lt(cl_j) = \phi_{min} + \phi_{e,ref} \frac{E_{res}(cl_j)}{\sum_{n_i \in G_j} E_{res}(n_i)},
\]

where $lt(cl_j) \in [0, 1, 2, \ldots]$. $\phi_{min}$ is the minimum lifetime
value and $\phi_{e,ref}$ is the energy-related reference value of CL
lifetime. $E_{res}(cl_j)$ is the residual energy of $cl_j$, $\sum E_{res}(n_i), n_i \in
G_j$ is the average residual energy of nodes which are in the
same grid with $cl_j$. CMs send the sensed data each time and
update $lt(cl_j) = lt(cl_j) - 1$. If $lt(cl_j) = 1$, the CMs send the last
sensed data with their residual energy to $cl_j$; then $lt(cl_j) = 0$.
Thus, $cl_j$ informs the one member who has the most residual
energy as a new CL, which broadcasts a CL announcement
with new CL lifetime values.

For any WSN, it is critical to handle unexpected sensor
nodes failures for robustness. In REEDG, instead of one
sensor node, a whole grid is employed as a member of the
transmitting chain. All the nodes in a grid play a role as
CL in turn. In one time instant, CL responds for the data
transmitting along the chain. When a CL fails, the other
nodes in the same grid with the CL take the place of the
current CL to collect and transmit the sensed information.
The CM which replaces the current CL is chosen according
to its residual energy. When CMs do not receive the answer
message from their CL for a time threshold $th_f$, a new CL electing process will start following the same rule as selecting CL initially in Algorithm 1. Since each node in a grid stores the information of the upstream and downstream grid when the chain forms, this new CL simply forwards data to the next CL toward the CH, until finally the aggregated data reach the sink. These techniques improve the robustness of REEDG against unexpected node failures. Even though multiple sensor nodes fail simultaneously along the forwarding path, our scheme also handles well because the data gathering can be achieved if only one node in a grid works well. In addition, it is not necessary to reconstruct the chain in the presence of a few nodes failures, which also enhances system robustness.

Furthermore, in REEDG, the updated control message is triggered and passed though the chain when a CH grid changes. Every node in CH grid can play as CH and the control message does not need to be updated during the grid lifetime as a CH grid. Compared with a single node as CH, which frequently refreshes control message when head node changes, REEDG reduces the number of control messages so as to conserve the scarce energy supply of sensor nodes.

4.6. Energy Consumption and Delay Analysis. This section presents the theoretical analysis of energy consumption at CM, CL, and CH nodes, delay during cluster formation, and energy consumption during data transmission.

4.6.1. Energy Consumption. On Chain Formation. In both PEGASIS and the hierarchical chain-based scheme, nodes need the global location information of all the nodes to form a chain. When a new node will be added to the chain, the protocols need to find out the node, which is nearest to the current CH or which adds the minimum $\sum d^a$, from all the nodes which are not in the current chain. They need to calculate the distance for these nodes and communication with them. Since the number of nodes is large, the process of chain forming consumes a lot of energy and time whereas REEDG just searches for a new chain member added to the chain in the neighbor of the current chain, which reduces energy and time consumption in the chain formation.

On CM. Our data transmission scheme always guarantees that CMs send their data to CL in their own grid so that the transmission distance is relatively small as it is in other chain-based approaches. Moreover, CMs do not aggregate data in REEDG due to data aggregation at CL. Since the information from the same grid is already related, data aggregating by CL is more efficient. Also, CM is just active to send data at the time instant negotiated with its CL and gets into sleep in the other time to save energy.

On CL. In REEDG, although CL spends more energy than CM due to sending aggregated data to the neighbor CL, CL is rotated in a grid and the node which has the most residual energy takes the role of CL each time, which balances the energy consumption on every node. If any CL fails, the other nodes in the same grid with the CL will replace it and the chain structure does not need to reform. Reducing the frequency of reforming of the chain implies reduced energy consumption and improved robustness in REEDG as compared to PEGASIS and the hierarchical chain-based scheme.

On CH. The energy consumed in transmitting the data from CH to the sink is the same in both PEGASIS and the hierarchical chain-based scheme, since they directly transmit aggregated data to the sink. In our approach, each node can be CH different times according to its location and residual energy so that the node near to the sink takes the role of CH more times than the node far away from the sink. Therefore, the energy consumption is well balanced on each node and this algorithm controls the data to flowing toward the node near to the sink.

4.6.2. Data Collection Delay. Another important issue in information collection WSN is data collection delay. In case of PEGASIS, a chain for data transmitting is constructed by all the nodes. During each round, a node takes turn to collect and send data to the sink. If approximately $t_{tra}$ time delay is required for one node to transmit information to the next node and about $t_{agg}$ time delay for data aggregation, then for $p$ nodes network, delay in one round can be estimated as follows:

$$t_{del}(\text{PEGASIS}) = p\left(t_{tra} + t_{agg}\right).$$

Since the number of nodes $p$ is always massive, although PEGASIS reduces the long distance transmission, at the same time, it introduces an excessive data collection delay.

In the hierarchical chain-based scheme, such as COSEN and ELCC, the data collection delay depends on the number of the network layers. If the network is divided into $n_{layer}$ layers, collecting data from all the nodes in one layer requires $p/n_{layer} - 1$ hops and from $n_{layer}$ layer heads to the sink transmission requires extra $n_{layer}$ hops. In total the time delay of the data collection is approximately obtained as follows:

$$t_{del}(\text{hier-chain}) = \left(\frac{p}{n_{layer}} - 1\right)\left(t_{tra} + t_{agg}\right) + n_{layer}\left(t_{tra} + t_{agg}\right).$$

In this scheme we take a network of $p = 100$ nodes for instance. According to COSEN, there are 5 chains and each one contains average 20 nodes. Therefore, $19(t_{tra} + t_{agg})$ time delay is required in each layer and $5(t_{tra} + t_{agg})$ additional delay occurs in the higher level chain. In total there is $24(t_{tra} + t_{agg})$ time delay to collect the data by the sink. That is much less as compared to PEGASIS. However, in some case, the number of nodes in each chain is not even. In order to wait for data collection of the longest chain, there is always excessive delay.

In REEDG, the data collection delay can be estimated as follows:

$$t_{del}(\text{REEDG}) = \left(\frac{p}{q} - 1\right)t_{tra} + q\left(t_{tra} + t_{agg}\right).$$
where the number of grids \( q \) should be under restriction and neither too large nor too small. In order to minimize the delay, let \( dt_{del}(\text{REEDG})/dq = 0 \); thus \( q \) is obtained as in (18):

\[
dt_{del}(\text{REEDG})/dq = \frac{-p}{q^2} t_{tra} + t_{tra} + t_{agg}.
\]

From (18), we can see that the value of \( q \) is related to the ratio of \( t_{tra} \) to \( t_{tra} + t_{agg} \). Generally, \( t_{agg} \) is much smaller than \( t_{tra} \) so that \( q \) is always set as close as possible to \( \sqrt{p} \) and the nodes deployed area should be considered when \( q \) is set. Also, a network of 100 nodes deployed in 100 m \( \times \) 100 m area is taken for instance and it is divided into 9 grids. Therefore, 10\( t_{tra} \) time delay is required in each grid and 9\( (t_{tra} + t_{agg}) \) additional delay occurs among CLs. This delay is less than the hierarchical chain-based scheme and PEGASIS. Moreover, the delay is not affected if the number of nodes in each grid is not equal. Because CM has enough time to report its sensed data in each grid before its CL sends data to the downstream CL as Figure 6 shows, REEDG is more delay-efficient.

5. Performance Evaluation

To evaluate the performance of the proposed approach REEDG, we implement the simulations using Castalia, which provides realistic channel models, radio models, and MAC layer protocols based on OMNet++4.1 [5, 23]. We also compare our simulation results with PEGASIS, COSEN, and ELCC in terms of total energy consumption, average data collection delay, network robustness, and network lifetime.

5.1. Experiment Environment. We assume that there are 256 sensor nodes distributed randomly over an area of 200 m \( \times \) 200 m and a fixed sink node located at (100, 600). The network is divided into 16 grids. We also assume each node has an initial energy of 1.5 J (Joules). The values for energy parameters adopted in our simulations are the same as that used in PEGASIS and ELCC; that is, \( E_{elec} = 50 \text{ nJ/b}, \) \( E_{amp} = 100 \text{ pJ/(b \cdot m^2)} \), and \( \alpha = 2 \). Moreover, the energy cost for data aggregation is considered as 5 nJ/bit/message. The power consumption of radio in idle and sleep modes is 0.22 mW and 0.000006 mW, respectively. The bandwidth of wireless channel is 1 Mbps and we adopted the MAC model of IEEE 802.15.4. In our experiments, a data packet size is 2000 bits and a control message is 64 bits long.

For CL selection, we set \( T_{max} \) and \( T_{run} \) to be \( 10^{-4} \), \( 10^{-2} \), and [0, \( 10^{-5} \)] second, respectively. \( E_{ref} \) is set as the same as the initial energy of a node. This ensures that the proposed approach has lower delay and high probability selecting the CL with the most residual energy. In addition, we set one time slot \( t_s = 5 \text{ ms} \), \( t_h = 3 \text{ ms} \) and information-processing time in a node is taken between 1 and 3 ms. For the lifetime of CH and CL, we set \( \delta_{min} = 1 \), \( \delta_{d_ref} = 2 \), \( \delta_{r_ref} = 5 \), \( \phi_{min} = 1 \), and \( \phi_{d_ref} = 2 \). For each simulation setup, we run at least 5 times with different random node distributions. Each result is averaged over these runs.

5.2. Simulation Results. Figure 8 shows the comparison of the energy consumption of network initializing and chain forming. It is shown that the amount of energy consumed in PEGASIS and COSEN is approximately the same since they both adopt the same chain forming approaches, greedy algorithm. The energy consumption in ELCC is smaller than that in PEGASIS and COSEN because several chains are constructed in small areas in ELCC and the nodes do not need to communicate with distant nodes to get their information. REEDG consumes the least energy for chain forming because in each grid only CL attends in the process of chain forming and each time CH just communicates with the neighbor grids of the current chain to choose the next grid to be added to the chain.

Figure 9 shows the comparison of the time delay of network initializing and chain forming. The time delay of chain forming in ELCC is slightly longer than the others because the network is divided into small areas in initialization and when every node adds into the chain it needs to calculate the minimum \( \sum d^2 \). Although in REEDG, a little time is spent on the CL selection, a grid is regarded as one unit so as to greatly shorten the length of the chain and decrease the calculation drastically. Therefore, the chain forming delay is the shortest in REEDG.

Figure 10 shows that average data collection delay changed with the number of the sensor nodes in the different approaches. The number of nodes is varied from 64 to 256. If the other parameters are fixed, the average delay increases when node density increases because more nodes create more data packets. The delay of PEGASIS sharp rises in pace with the nodes increasing and it has the largest time delay among these approaches because the chain is much too long with a large number of sensor nodes. REEDG achieves lower delays compared to the other approaches for all node densities because the length of the chain is shorter than others and data collection in each grid does not affect the time delay of the chain. The delay of COSEN and ELCC is shorter than that in PEGASIS because they divided the chain into several
Figure 9: Time delay of network initializing and chain forming.

Figure 10: Average data collection time (ms)

Information collecting approaches

Time (ms)

PEGASIS | COSEN | ELCC | REEDG

Average data collection time (ms)

Number of nodes (p)

PEGASIS | ELCC | COSEN | REEDG

Figure 11: Data transmission ratio versus certain percent of node failures.

Figure 12: Average energy consumption one each node after 100 runs varies with respect to the node failure rate. Clearly it tends to increase, for all the approaches, since the chain needs to be reconstructed when node fails and fewer nodes participate in the data collection as the node failure rate increases so that the distances between the nodes become greater, and nodes have to become leaders more often. The curve representing average energy consumption on each node in REEDG, at first, shows a slowly increase because the data transmitting chain is not necessary to be reformed when a few nodes fail; later on, the average energy consumption increases quickly, since less and less nodes can be used due to the higher number of failures. Although the average energy consumption on each node also increases as the node failure rate keeps increasing, our approach still outperforms the other approaches.

Figure 13 shows the total energy cost of the four approaches after several hundreds of rounds when 30% nodes fail. We can see that REEDG consumes less energy than that in the other approaches due to less communication cost for chain reforming. REEDG only reconstructs the chain when no node is alive in one grid. Moreover REEDG adopts the adaptive sleep schedule which allowed the radio of nodes sleep in the most of the time and only work when they need send data in order to save energy further. The total energy consumption of ELCC is smaller than that in PEGASIS and COSEN because it constructs small chain in local area and has the minimum sum of the square of transmission distance \( \sum d^2 \).

Figure 14 plots the ratio of the previous two quantities, average energy consumption and data transmission ratio, and thus illustrates the energy efficiency (the energy cost of small chains to reduce the length of transmission chains; nevertheless, the delay is prolonged if the number of nodes is not even in each small chain.

In order to investigate the performance of system robustness, the experiments were carried out keeping the other parameters fixed and progressively increasing the number of node failures; specifically, the percent of node failure ranges from 0 to 50% of the total number of nodes. We set nodes failure in random position and averaged our results in each run since the simulation results are affected by the location of those nodes. Figure 11 shows data transmission ratio versus the percent of node failures for the different approaches. Not surprisingly, all the approaches obtain almost 100% transmission rate with 0 node failures and the success rate reduces with the increasing node failures. Remarkably, because one grid is regarded as the chain member and all nodes in the grid can relay data in REEDG, our approach is more robust and manages to deliver 31% of the packets sent, even with 50% failure rate. In the other approaches, the data transmission ratio drops to approximately 23% when half of the total number of nodes fails. When 20% nodes fail, REEDG improves the data transmission ratio at lowest 24%.

Figure 12 shows a chart plotting the average energy consumption on each node after 100 runs varies with respect to the node failure rate. Clearly it tends to increase, for all the approaches, since the chain needs to be reconstructed when node fails and fewer nodes participate in the data collection as the node failure rate increases so that the distances between the nodes become greater, and nodes have to become leaders more often. The curve representing average energy consumption on each node in REEDG, at first, shows a slowly increase because the data transmitting chain is not necessary to be reformed when a few nodes fail; later on, the average energy consumption increases quickly, since less and less nodes can be used due to the higher number of failures. Although the average energy consumption on each node also increases as the node failure rate keeps increasing, our approach still outperforms the other approaches.

Figure 13 shows the total energy cost of the four approaches after several hundreds of rounds when 30% nodes fail. We can see that REEDG consumes less energy than that in the other approaches due to less communication cost for chain reforming. REEDG only reconstructs the chain when no node is alive in one grid. Moreover REEDG adopts the adaptive sleep schedule which allowed the radio of nodes sleep in the most of the time and only work when they need send data in order to save energy further. The total energy consumption of ELCC is smaller than that in PEGASIS and COSEN because it constructs small chain in local area and has the minimum sum of the square of transmission distance \( \sum d^2 \).

Figure 14 plots the ratio of the previous two quantities, average energy consumption and data transmission ratio, and thus illustrates the energy efficiency (the energy cost of...
successful transmission of one packet). When there is no node failure, all approaches almost start from the same value, but the energy efficiency in PEGASIS and COSEN decreases quickly as the node failure rate keeps increasing. However, in REEDG the energy efficiency reduces slowly so that REEDG has higher energy efficiency in improving system robustness. Our approach manages to keep the energy consumption low and still maintains a remarkable degree of robustness at the same time.

Finally, the last chart in Figure 15 shows the comparison of lifetime among PEGASIS, COSEN, ELCC, and our proposed algorithm REEDG. We can see that REEDG has the longest lifetime among these approaches. This is caused by the following reasons. (1) REEDG reduces the frequency of reconstructing the data transmitting chain when some nodes fail. (2) REEDG adopts the adaptive node sleep scheduling to save energy. (3) REEDG allows each node to take the role of CH different times according to the distance from the node to sink and the residual energy of the node so that the energy cost is balanced. As 20% nodes die, REEDG prolongs the network lifetime by at least 13%. In all the approaches, nodes die quickly after 30% nodes death due to the greater distances between the nodes. Also, more frequency of nodes as leaders causes the energy to drain rapidly. COSEN has longer lifetime than that in PEGASIS when 1% nodes die, since it makes the energy cost balance on each node. Moreover, ELCC outperforms PEGASIS by avoiding the length of the chain much too long and minimizing the transmission distance of the chain.
6. Conclusions

This paper proposed a novel robust and energy-efficient data-gathering (REEDG) approach for sensor information collecting system. REEDG is based on a combination of grid and chain network structure. It can improve the system robustness and reduce energy consumption so as to increase the network lifetime without degrading the data-gathering performance. REEDG outperforms the other approaches by allowing the radios of nodes to sleep in most of the time and work only when they need to send data REEDG. Besides, each node takes the role of CH different times according to the location and the residual energy of the node so that the energy cost is balanced in the whole network. Moreover, in REEDG, one grid is regarded as a unit to form a data transmitting chain, which reduces the frequency of the chain reconstructing and improves the system robustness. Our experiments proved that REEDG outperformed the state-of-the-art approaches by improving the data transmission ratio at lowest 24% when 20% nodes fail as well as prolonging the network lifetime by at least 13% as 20% nodes die.

Conflict of Interests

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61301094), NPU Foundation for Fundamental Research (NPU-FFR-JCY20130135), and the Post-doctoral Science Foundation of China (2014M552490).

References

[1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS-33 ’00), pp. 3005–3014, IEEE Computer Society, Maui, Hawaii, USA, January 2000.

[2] S. Lindsey and C. S. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in Proceedings of the IEEE Aerospace Conference, pp. 1125–1130, March 2002.

[3] K. Maraiya, K. Kant, and N. Gupta, "Architectural based data aggregation techniques in wireless sensor network: a comparative study," International Journal on Computer Science & Engineering, vol. 3, no. 3, p. 1131, 2011.

[4] J. Ben-Othman, K. Bessaoud, A. Bui, and L. Pilard, "Self-stabilizing algorithm for energy saving in wireless sensor networks," in Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC ’11), pp. 68–73, July 2011.

[5] "National ICT australia-castalia," http://castalia.npc.nicta.com.au/.

[6] H. Luo, Y. Liu, and S. K. Das, "Routing correlated data in wireless sensor networks: a survey," IEEE Network, vol. 21, no. 6, pp. 40–47, 2007.

[7] H. Chen, H. Mineno, and T. Mizuno, "Adaptive data aggregation scheme in clustered wireless sensor networks," Computer Communications, vol. 31, no. 15, pp. 3579–3585, 2008.

[8] W. Lou, "An efficient N-to-1 multipath routing protocol in wireless sensor networks," in Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS ’05), pp. 1–8, Washington, DC, USA, November 2005.

[9] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’03), vol. 3, pp. 1948–1953, New Orleans, LA, USA, March 2003.

[10] C. Liu and G. Cao, "Distributed monitoring and aggregation in wireless sensor networks," in Proceedings of the IEEE INFOCOM, March 2010.

[11] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," in Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI ’02), pp. 131–146, 2002.

[12] M. Bagaa, M. Younis, A. Ouadjaout, and N. Badache, "Efficient multi-path data aggregation scheduling in wireless sensor networks," in Proceedings of the IEEE International Conference on Communications (ICC ’13), pp. 1560–1564, IEEE, June 2013.

[13] M. Rati, R. Dezfooli, K. A. Bakar, and M. Lee, "Multipath routing in wireless sensor networks: survey and research challenges," Sensors, vol. 12, no. 1, pp. 650–685, 2012.

[14] S.-M. Jung, Y.-J. Han, and T.-M. Chung, "The concentric clustering scheme for efficient energy consumption in the PEGASIS," in Proceedings of the 9th International Conference on Advanced Communication Technology (ICACT ’07), vol. 1, pp. 260–265, February 2007.

[15] K.-H. Chen, J.-M. Huang, and C.-C. Hsiao, "CHIRON: an energy-efficient Chain-based hierarchical routing protocol in wireless sensor networks," in Proceedings of the Wireless Telecommunications Symposium (WTS ’09), April 2009.

[16] Q. Mamun, S. Ramakrishnan, and B. Srinivasan, "An efficient localized chain construction scheme for chain oriented wireless sensor networks," in Proceedings of the 10th International Symposium on Autonomous Decentralized Systems (IADS’11), pp. 3–9, March 2011.

[17] M. N. Mehanathan, "use of tree traversal algorithms for chain formation in the PEGASIS data gathering protocol for wireless sensor networks," Transactions on Internet and Information Systems, vol. 3, pp. 612–627, 2009.

[18] Q. Mamun, S. Ramakrishnan, and B. Srinivasan, "Selecting member nodes in a chain oriented WSN," in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’10), pp. 1–6, April 2010.

[19] N. Tabassum, Q. E. K. Mamun, and Y. Urano, "COSEN: a chain oriented sensor network for efficient data collection," in Proceedings of the 3rd International Conference on Information Technology: New Generations (ITNG ’06), pp. 262–267, April 2006.

[20] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCOM ’02), pp. 148–159, September 2002.

[21] Z. Zehua, X. Xiaoqing, W. Xin, and P. Jianping, "An energy-efficient data-dissemination protocol in wireless sensor networks," in Proceedings of the International Symposium on a
World of Wireless, Mobile and Multimedia Networks (WoWMoM ’06), pp. 13–22, June 2006.

[22] M. A.-T. Fadi, S. H. Hossam, and A. I. Mohamed, “Quantifying connectivity of grid-based Wireless Sensor Networks under practical errors,” in Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN ’10), pp. 220–223, October 2010.

[23] “OMNeT++ Network simulator,” http://www.omnetpp.org/.
Submit your manuscripts at
http://www.hindawi.com