J Supercomput (2010) 52: 23–46
DOI 10.1007/s11227-009-0268-7

Connectivity and coverage maintenance in wireless sensor networks

**Yuanyuan Zeng · Cormac J. Sreenan ·
Naixue Xiong · Laurence T. Yang · Jong Hyuk Park**

Published online: 21 February 2009
© Springer Science+Business Media, LLC 2009

Abstract One of the main design challenges for wireless sensor networks (WSNs) is to obtain long system lifetime without sacrificing system original performance such as communication connectivity and sensing coverage. A large number of sensor nodes are deployed in redundant fashion in dense sensor networks, which lead to higher energy consumption. We propose a distributed framework for energy efficient connectivity and coverage maintenance in WSNs. In our framework, each sensor makes self-scheduling to separately control the states of RF and sensing unit based on dynamic coordinated reconstruction mechanism. A novel energy-balanced distributed connected dominating set algorithm is presented to make connectivity maintenance; and also a distributed node sensing scheduling is brought forward to

---

Y. Zeng
School of Electronic Engineering, Wuhan University, Wuhan, China

Y. Zeng · C.J. Sreenan
Department of Computer Science, University College Cork, Cork, Ireland

Y. Zeng
e-mail: yz2@cs.ucc.ie

C.J. Sreenan
e-mail: cjs@cs.ucc.ie

N. Xiong
Department of Computer Science, Georgia State University, Atlanta, GA, USA
e-mail: nxiong@cs.gsu.edu

L.T. Yang
Department of Computer Science, St. Francis Xavier University, Antigonish, Canada
e-mail: ltyang@stfx.ca

J.H. Park (✉)
Department of Computer Science and Engineering, Kyungnam University, Kyungnam, Korea
e-mail: partkjonghyuk1@hotmail.com

maintain the network coverage according to the surveillance requirements. We implemented our framework by C++ programming, and the simulation results show that our framework outperforms several related work by considerably improving the energy performance of sensor networks to effectively extend network lifetime.

Keywords Wireless sensor networks · Energy efficient · Connectivity · Coverage · Connected dominating set · Self-scheduling

1 Introduction

The research on wireless sensor networks has been fueled up by many applications in various areas, such as environment monitoring, target or object tracking, industry automation and control, and so on [2, 14, 29]. Dense deployment helps to improve reliability and extends longevity of sensor networks. In practice, large scale wireless sensor networks are usually deployed randomly. However, the energy efficiency is one of the major constraints in wireless sensor networks. Wireless sensors are very limited in their processing, computing and communication capabilities as well as the storage and power supply. Moreover, due to the sheer number of sensor nodes and the potentially hostile environment, it is usually very hard to recharge the battery after the deployment of sensor nodes. So how to extend the lifetime of a sensor network under the stringent energy constraint of each individual sensor node is an important and challenging problem.

We assume that all nodes share common communication and sensing tasks in wireless sensor networks. Not all sensors are required to perform the transmission and sensing tasks during the whole system lifetime. Making some nodes sleep does not affect the overall system function as long as enough working nodes could assure the network connectivity and coverage. If we initially deploy a large number of sensors and schedule them to work alternatively, system lifetime can be prolonged correspondingly. Our design goal is to make both connectivity and coverage maintenance for data collection and extend the lifetime of the sensor nodes as long as possible.

A dominating set (DS) of a graph is a subset of nodes such that each node in the graph is either in the subset or adjacent to at least one node in that subset. A connected dominating set (CDS) is a DS, which induces a connected sub graph. A CDS is a good candidate of a virtual backbone for wireless networks, because any node in the network is less than 1-hop away from a CDS node. Only the backbone nodes are responsible for relaying messages for the network. The non-backbone nodes can thus turn off their communication module to save energy when they have no data to be transmitted out. One objective for constructing the backbone is to minimize the size of a backbone (i.e., the number of backbone nodes). Unfortunately computing a minimal CDS (denoted by MCDS) of a Unit disk graph [8] has been proved to be NP-hard [19]. When comes to network coverage, given a randomly and densely deployed wireless sensor network, it is desirable to have sensors autonomously schedule their duty cycles while satisfying the sensing coverage degree. The problem is called coverage maintenance. We try to make coverage maintenance for data collection and extend the lifetime of the sensor nodes as long as possible by selecting only a subset of nodes to be on-duty state and keeping the remaining nodes in off-duty state. Based on the above ideas, we propose a framework for energy efficient connectivity and coverage maintenance in wireless sensor networks for data sensing and transmissions to extend the network lifetime in this paper.

The main contributions of this work can be summarized as follows.

A distributed framework for energy efficient connectivity and coverage maintenance for wireless sensor networks is presented to extend the network lifetime.

Firstly, for connectivity maintenance of WSNs, we propose a novel efficient distributed approximation algorithm (ECDS) that computes a sub-optimal MCDS in polynomial time. Since the backbone nodes have heavy communication load for relaying messages for the network, their energy consumption is high. There are three major advantages of the proposed algorithm: (1) the algorithm is fully distributed, which can be easily implemented in sensor networks; (2) the constructed CDS has a small size, which reduces the overhead of maintaining the backbone and the cost in communication; (3) the constructed CDS achieves load balancing, which extends the lifetime of the network.

Secondly, for coverage maintenance, we propose an energy conservation node self-scheduling algorithm (ECSS) in sensor networks. Different from the existing work, ECSS has following advantages. (1) Our mechanism is based on a probabilistic sensing model. (2) Our mechanism guarantees a certain sensing reliability and also provides some degree of redundancy according to application requirements. (3) The mechanism considers the residual energy and detection ability of nodes.

The remainder of this paper is organized as follows. Section 2 briefly introduces the related work in the literature. Section 3 is the framework of the networks. Section 4 discusses the distributed ECDS algorithm. Section 5 discusses our ECSS mechanism. Section 6 presents simulation results. Section 7 is the conclusion and future work.

2 Related work

Minimizing energy consumption and maximizing the system lifetime have been a major design goal for wireless sensor networks. In the last few years, researchers actively explored advanced power conservation mechanisms for wireless sensor networks.

Extensive work has been done on the connectivity maintenance issues. Research in [21] focuses on energy conservation by controlling sensor transmission power in order to maintain network connectivity. It demonstrates that the network connectivity can be maintained if each sensor has at least one neighbor in every cone of $2\pi/3$. Xu et al. [25] propose two algorithms that can conserve energy by identifying redundant nodes of connectivity. In GAF [26], nodes use geographic location information to divide the world into fixed square grids. Nodes within a grid switch between sleeping and listening, with the guarantee that one node in each grid stays up to route packets. SPAN [7] is another protocol that achieves energy efficiency for wireless sensor networks by introducing off-duty and on-duty cycles for sensor nodes. Rozell et al. [15] propose an optimal power scheduling which can yield significant power savings over communication strategies that use a fixed number of bits on each communication link.

Dominating set based topology leads to a virtual backbone for the deployed ad hoc and sensor networks [3, 16, 17]. The virtual backbone is formed by representing the connected routing nodes as a connected dominating set (CDS). Since the minimal CDS problem is NP-hard, most previous work has focused on finding heuristics for reducing the size of CDS. Current MCDS approximation algorithms include centralized and distributed algorithms. Following the increased interest in wireless ad hoc and sensor networks, many distributed approaches have been proposed because of no requirements for global network topology knowledge. These algorithms contain two types. One type is to find a CDS first, then prune some redundant nodes to attain MCDS. Wu and Li proposed in [24] a distributed algorithm with $\Theta(m)$ message complexity and $O(\Delta^3)$ time complexity, the approximation factor at most $n/2$. Butenko et al. [6] constructs a CDS starting with a feasible solution, and recursively removes nodes from the solution until a MCDS is found. The other type is to form a maximal independent set (MIS) at first, and then find some connectors to make the independent nodes connected together. P.J. Wan et al. [19] proposes a distributed algorithm with performance ratio of 8. Min et al. in [13] propose an improved algorithm by employing a Steiner tree in the second step to connect the nodes in the MIS with performance ratio of 6.8. Recently, there's a great increasing focus on low cost and low energy consumption in wireless networks. Wu et al. in [23] present an algorithm for power aware connected dominating set based on [6]. Acharya et al. in [1] present a power aware MCDS construction when introducing a concept of threshold energy level for dominating nodes based on [6]. Those algorithms do not consider the balance of energy consumption in the network.

The other issue, coverage maintenance, has also driven lots of research efforts recently. Target detection and field surveillance are among the most prominent applications of Sensor Networks. Tian et al. [18] present a node-scheduling algorithm to turn off redundant sensors if their sensing areas are covered by their neighbors. In [9], Hsin et al. propose a random scheduling scheme and a coordinated sleep scheduling scheme to maintain network coverage using low duty-cycle sensors. In [28], a sensor node uses a probing mechanism to determine whether it should sleep. In [27], Yan et al. divide the whole monitored field into grids and transformed the area coverage problem into the grid intersection point coverage problem. Each node is able to dynamically decide a schedule for itself, which guarantees the grid intersection points within its sensing range to be monitored by itself or by its neighbors at any time. Lazos et al. [10] propose to evaluate the detection probability of mobile targets when n sensors are stochastically deployed to monitor a field of interest. The authors map the target detection problem to a line-set intersection problem and derive analytical formulas using tools from Integral Geometry and Geometric Probability. The detection probability depends on the length of the perimeters of the sensing areas of the sensors and not their shape.

Many recent literatures consider both communication connectivity and sensing coverage. In [20], the proposed approach makes a clear distinction between coverage and connectivity, and it derives the same conditions under which connectivity can be obtained without compromising sensing coverage. In [30], a distributed coverage and connectivity centric technique for the selection of active sensor nodes in dense sensor networks is proposed. The active node selection procedure is aimed at providing the highest possible coverage of the sensor field, and also assures network connectivity for routing based on the concept of a CDS. A CDS that satisfies both sensing coverage and connectivity constraints acts as a backbone for the sensor network. In [11], the authors consider wireless sensor networks satisfying the case that each node either monitors one target or is just for connection. It is assumed that the wireless sensor network has l targets, and that each is monitored by k sensor nodes. If $k = 2$ and the graph G corresponding to the wireless sensor network is $(l + \max\{1, l - 4\})$-connected, or $k \geq 3$ and G is $(l(k - 1) + 1)$-connected, then find k(the maximum number) disjoint sets, each of which completely covers all the targets and remains connected to one of the central processing nodes. The disjoint sets are activated successively, and only the sensor nodes from the active set are responsible for monitoring the targets and connectivity; all other nodes are in a sleep mode. In addition, they give the related algorithms to find the k disjoint sets. Woehrle et al. [22] address the deployment problem of WSN using a multiobjective evolutionary algorithm which allows identifying the tradeoffs between low-cost and highly reliable deployments to provide the decision maker with a set of good solutions to choose from. Bai et al. [5] study the issue of optimal deployment to achieve four connectivity and full coverage for wireless sensor networks under different ratios of sensors' communication range to their sensing range. And then Bai et al. [4] propose deployment patterns to achieve full coverage and three-connectivity and full coverage under different ratios of sensor communication range over sensing range for wireless sensor networks.

3 General network framework

3.1 Connectivity and coverage maintenance

The self-organized sensor networks involve two different topology issues, the connectivity maintenance and the coverage maintenance. Consider both connectivity-based and coverage-based metrics, we propose a framework to assign different roles for wireless sensor networks, i.e., to separate connectivity from the coverage management. In our framework, we consider connectivity and coverage maintenance separately by different mechanisms. It implies that the connectivity maintenance will decide on the node on/off state of its RF units and the coverage maintenance will decide on the on/off state of sensing units. We assume that each sensor can separately control the states of RF and sensing unit, i.e., the state of the RF unit is independent from the sensing unit. Then, there will be four possible sensor states: (1) *sleep*, i.e., both sensing units and RF units are off-duty. (2) *active*, i.e., both sensing units and RF units are on-duty. (3) *sensing*, i.e., sensing units are on-duty and RF units are off-duty. (4) *relaying*, i.e., sensing units are off-duty and RF units are on-duty.

The aim of our connectivity and coverage maintenance framework is to construct a connected dominating set based backbone to take responsibility of communication and then use the self-scheduling sensing mechanism for nodes to provide enough coverage. The connectivity maintenance provides a network backbone to support

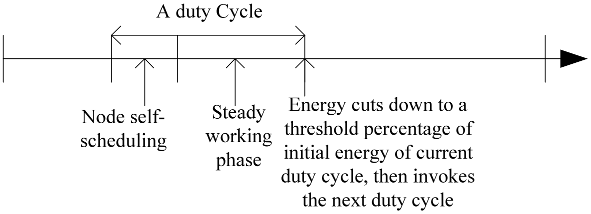


Fig. 1 A duty cycle of our framework

network-wide routing functionality. It supports application specific sensing queries and data gathering of the sensors. We induce the connected dominating set to form a virtual backbone in the networks to maintain network connectivity. To cut down communication overhead at best, we try to find a minimum connected dominating set of the network graph. An MCDS satisfies: (1) each node is either a backbone node or is one-hop connected to a backbone node. (2) The backbone nodes are connected. After the backbone construction, the backbone nodes will turn on the RF units for connectivity need, and the non-backbone nodes will turn off the RF units to save energy. The coverage maintenance provides the network event detection capability. On the other hand, node self-scheduling sensing mechanism can reduce system overall energy consumption by identifying application-specific redundant nodes in respect of sensing coverage redundancy threshold and then assigning them an off-duty sensing state. The nodes in off-duty sensing state consume lower energy than the normal on-duty sensing one. It should preserve network coverage with a certain fault tolerant requirements of the applications (bound by a coverage redundancy threshold value).

To balance the energy consumption of sensor nodes, the framework operation time is divided into duty cycles to make dynamic reconstruction of connectivity and coverage topology control. Each duty cycle runs in two phases: self-scheduling phase and steady working phase. Figure 1 shows the duty cycle of our framework.

In Fig. 1, during the node self-scheduling phase, nodes investigate the mechanisms and determine the RF and sensing state of nodes. During the working phase, some nodes do not work in the duty cycle and turn off RF and sensing unit. The other nodes will turn on and work during the steady working phase to maintain network connectivity and coverage. We take dynamic reconstruction strategy to balance energy consumption in the networks. Each cycle when a CDS is constructed as backbone, the length of operating time of this CDS for this round will be determined according to the residual energy of the CDS nodes. If the minimal residual energy of nodes in CDS is cut down to a certain percentage (such as 50%) of the initial energy of current cycle, the operating period of this cycle is due. When the operating period of current round expires, the next duty cycle will be invoked.

3.2 System models and basic ideas

We assume that all nodes in wireless sensor networks are distributed in a two-dimensional plane and have an equal maximum transmission range of one unit. The network topology is modeled as a unit disk graph, UDG in Short. We use graph $G = (V, E)$ to represent such networks, where V is the set of sensor nodes and E is the set of edges.

Each node u has a weight $w(u)$ of being in the backbone. Here $w(u)$ can be the value computed based on a combination of its remaining battery power *energy*(u) and its effective degree *degree*(u) in the communication graph, and so on. Thus, each time the constructed CDS backbone could has higher energy level and smaller size under the condition of energy efficiency. Note that the definition effective degree is different in each phase. Each node knows its own location, which can be obtained at a low cost from Global Positioning System (GPS) or through location discovery algorithms. Each node can know its neighbors' information by simple neighbor information exchange process. The surveillance field can be represented by a 2D grid. Let $G = \{g_1, g_2, \ldots, g_m\}$ be the set of all grid points in surveillance field. Let g_i be the location vector for grid point g_i, i.e., $g_i = \langle x_i, y_i \rangle$, where x_i and y_i are the centroid coordinates for grid point g_i. We use s to denote the set of n sensor nodes that have been placed in the sensor field. A node with id k is referred as s_k. Let l_k be the location vector of node s_k. Assume that all sensor nodes are equipped with the same type of sensing and communication hardware, i.e., they have the same maximum sensing range R_s. Let $d_{i,k}$ be the distance between the grid point g_i and the sensor s_k. $d_{i,k} = \|g_i - l_k\|$.

4 Distributed energy balanced CDS backbone

The aim for energy balance CDS algorithm is to compute a sub-optimal MCDS as a backbone for wireless sensor networks. Our distributed energy balanced CDS backbone construction algorithm includes two phases: MIS construction and then to select connectors to make the MIS nodes connected into a CDS construction. In the first phase, we compute a maximal independent set (MIS) of the network graph. An independent set (IS) of a graph is a subset of V that no two nodes in the subset have an edge. An MIS of a graph is an independent set that cannot include any more nodes in V. Thus an MIS is a DS of a graph. Note that this DS (obtained as the MIS) may not be connected. The second phase of the algorithm is to choose the minimal number of nodes (called connectors) to make the DS connected, i.e., a CDS.

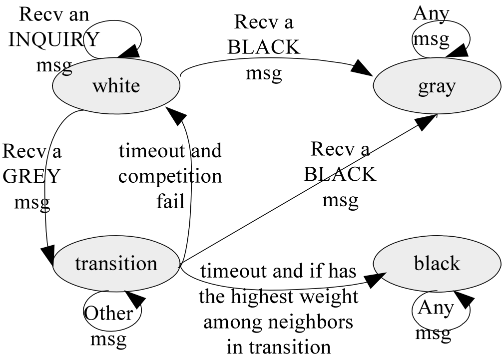
4.1 Phase 1: MIS construction

The algorithm always starts from a node that initiates the execution. We call this node as initiator. The initiator can be designated randomly beforehand. We use colors to indicate if a node is in MIS or not, i.e., use black to indicate the nodes in MIS and grey to indicate non-MIS nodes.

Figure 2 can describe the algorithm execution of each node. Each node is in one of the four states: white, black, grey and transition. Initially, all nodes are white, and at the completion of the algorithm all nodes in the network must be either in black (become a MIS nodes) or in grey (become a non-MIS nodes). The transition state is an intermediate state.

There are three types of messages in phase 1: (1) BLACK message: sent out when a node becomes a black node; (2) GREY message: sent out when a node becomes a grey node; (3) INQUIRY message: sent out when a node inquires the weights and states of its neighbors.

Fig. 2 State transition diagram of MIS construction



Each message contains node state and id, i.e., $state(i)$, $id(i)$. The replies of INQUIRY message contains the weight of nodes, i.e., $w(i)$. Node calculates its $w(i)$ by battery power and its effective degree. The effective degree is the number of its neighbors in the white and transition state.

The initiator starts the construction of a new CDS by executing the following procedure that shows in Algorithm 1. In Algorithm 1, firstly, the initiator colors itself in black. A node that colors itself in black will broadcast a BLACK message to its neighbors immediately to indicate itself as an MIS node. A white neighbor that receives the BLACK becomes a grey node (i.e., a non-MIS node), and broadcasts a GREY message to indicate its neighbors simultaneously. A white node that receives a GREY message is a neighbor of the non-MIS node. It needs to compete to become a black node. So it broadcasts an INQUIRY message toward its neighbors to inquire their states and weights. The node sets a timeout to wait for the replies of the INQUIRY message, and then enters the transition state. It will stay in the transition state until timeout expires. During the timeout, it may receive a BLACK, INQUIRY or GREY message. If receives a BLACK message, the node becomes grey and broadcast a GREY message. If receives other messages, the node ignores them and only stays in the transition. When timeout is due, if the node finds that it has the highest weight among all neighbors in transition (or the node has no neighbors in transition state) based on the replies of INQUIRY, it will become black node. Then the node in the black state does the same as the other black ones do. Otherwise, the node enters back into white state after the timeout. During the coloring process, each grey node will keep a list of all the adjacent black (MIS) nodes. The same operation continues from node to node as the BLACK or GREY messages propagate, until nodes have entered the state of either black or grey.

Theorem 1 *The set of black nodes that computed by the phase* 1 *forms a maximal independent set of the network graph.*

Proof We donate the set of black nodes that computed by phase 1 algorithm as B. The MIS algorithm colors the nodes of the graph layer by layer, and propagates out from the initiator to reach all nodes in the network, with one layer of black and the

Algorithm 1 MIS construction

```
initiator () {
    Color itself black;
    Broadcast a BLACK message;
    }

Each node i, performs the following function:
MIS-construction () {
    Recv a msg;
    If state(i) is black/grey then   //final state of phase 1
            Ignore the msg and return;
    If state(i) is white then   //initial state
            Switch on msg-type
                        BLACK:   // the sender is an MIS node
                                    state(i) ← grey;
                                    Broadcast a GREY msg;
                        GREY:  // the sender is a non-MIS node
                                    Broadcast an INQUIRY msg;
                                    Set a timeout;
                                    state(i) ←  transition;
                        INQUIRY:   // the sender is in transition
                                    Reply its state(i) and w(i);
            endSwitch
    If state(i) is transition then   //state during timeout period
            Switch on msg-type
                        BLACK:// the sender is an MIS node
                                    state(i) ←  grey;
                                    Broadcast a GREY msg;
                        INQUIRY://sender is in transition
                                    Reply its state(i) and w(i);
                        GREY:   //the sender is a non-MIS node
                                    Ignore the msg;
            endSwitch
    }

Upon the expiration of timeout for transition state, it executes the following procedure:
    transition-timeout () {//timeout expires
            If w(i) is highest among transition neighbors then
                    state(i) ← black;
                    Broadcast a BLACK msg;
            else
                    state(i) ←white; //competition fails
    }
```

next layer as grey. At each layer (except initiator), black nodes are selected by grey nodes of previous layer and are marked black. The construction incrementally enlarges the black node set by adding black nodes 2 hops away from the previous black nodes set. Also the newly colored black nodes could not be adjacent to each other, for the interleaving coloring layer of black and grey nodes. Hence every black node is disjoint from other black nodes. This implies that B forms an independent set. Further, the algorithm will end up with black or grey nodes only. Each grey node must have at least one black neighbor, so if coloring any grey node black, B will not be disjoint anymore. Thus, B is the maximal independent set. □

4.2 Phase 2: CDS construction

Since an MIS is a dominating set with non-adjacent edges, a CDS can be constructed by making the DS connected together, i.e., by connecting the nodes in an MIS through some nodes (called connectors) not in the MIS. A localized approximation of minimum spanning tree may perform well enough. We call it a dominating tree. We design a greedy approximation algorithm that every MIS node selects the non-MIS node with the highest weight which are equivalent to 2-hop to interconnect two or more MIS nodes, as a connector.

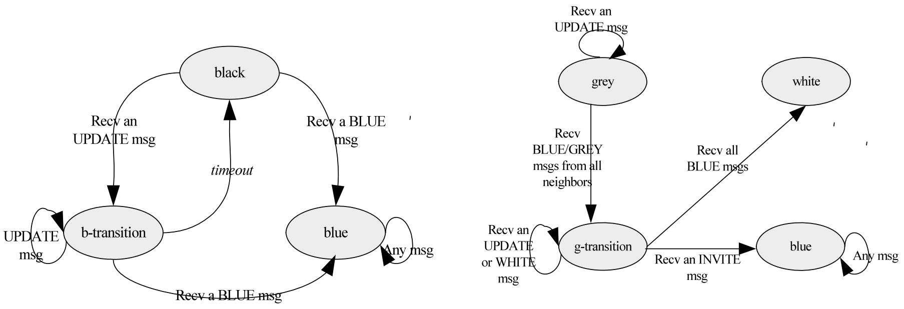
Figure 3 can describe the CDS construction. During the process, an MIS node can be in one of the three states: black, b-transition and blue, as shown in Fig. 3(a). A non-MIS node can be in one of the four states: grey, g-transition, blue and white, as shown in Fig. 3(b). When the CDS construction is completed, all the nodes in the network graph are either blue or white. And all the blue nodes are the CDS nodes.

There are three types of messages: (1) BLUE message, sent out when a node becomes a CDS node. (2) INVITE message, sent out by an MIS node to invite a non-MIS neighbor to be as a connector. (3) UPDATE message, sent out by a non-MIS node to notify all its neighbors about its current weight. In UPDATE message, the node will calculate its weight $w(i)$ by battery power and its effective degree. The effective degree is defined as the number of its current black neighbors (i.e., the number of its MIS neighbors in black state). (4) WHITE message, sent out by a node in white.

After finishes phase 1 algorithm, a node in the network graph is either in black (i.e., become a MIS node) or in grey (i.e., become a non-MIS node), and each grey node keep a list of its black neighbors. Then the node will check to see whether it could begin the second phase of CDS algorithm. If all its neighbors become grey or black, the node will begin phase 2.

The phase 2 can be described as CMIS-construction process that shown in Algorithm 2. In Algorithm 2, a node that colors itself blue will broadcast a BLUE message to indicate itself as a CDS node. We assume that all messages are delivered in order.

When an MIS node receives GREY messages from all its neighbors, it begins phase 2 by entering the b-transition state from the black state. Nodes in the b-transition state will detect current node information among its neighborhood to select



(a) state transition diagram for MIS nodes (b) state transition diagram for non-MIS nodes

Fig. 3 State transition diagram of CDS algorithm

Algorithm 2 CMIS construction

```
CMIS-construction {
  If state is black and recv GREY msgs from all neighbors then
  //MIS node begins phase 2
              state(i)← b-transition;
  If state is grey and recv BLACK msgs from all neighbors then
  //non-MIS node begins phase 2
              state(i)← g-transition;
  Recv a msg;
  Switch on state(i){
    blue:
       Ignore the msg and return; //terminates
    b-transition:
       Set a timeout;
       If recv a BLUE msg then
              state(i) ← blue;
       Broadcast a BLUE msg;
       If recv an UPDATE msg then
              Get the w(i) of UPDATE senders;
       else
              Ignore the msg;
    g-transition:
       Broadcast a UPDATE msg;
       If recv an INVITE msg then
              state(i) ← blue;
              Broadcast a BLUE msg;
       If recv a BLUE msg then
              Broadcast a WHITE msg;
              state(i) ← white;
              return; // process terminates
       If recv an UPDATE or a WHITE msg then
              Ignore the msg;
       }
  endSwitch
}
```

Before a node enters the b-transition state, it sets a timeout overhearing UPDATE message from its effective neighbors. Upon the expiration of this timeout, it executes the following procedure:

```
b-transition-timeout() {//timeout expires
  find the highest w(k) from the senders of UPDATE;
  send out an INVITE msg to k; //invite k to be a connector
  state(i) ← black;
  }
```

connectors. When a node enters in the b-transition state, it will set up a *timeout* timer immediately. During the *timeout* period, it stays for UPDATE messages from neighbors and collect the neighborhood non-MIS nodes information. During the *timeout* period, it may receive a BLUE message. Then that implies a connector is already chosen among its neighborhood, then the timeout stopped and the node enters into the blue state. When the node enters in the blue state, it broadcasts a BLUE message to indicate itself as a CDS node as the other blue nodes do. Otherwise, when timeout for the b-transition state expires, the node will select a connector from the effective non-MIS neighbors with the highest weight, and then sends out an INVITE message to the selected one. The effective neighbor is defined as: the non-MIS node must have a blue neighbor. This definition is used to guarantee the constructed structure is connected together, not becomes disconnected areas. When a node sends out an INVITE, it will enter back into the black state.

When a non-MIS node receives GREY and BLACK messages from all its neighbors, it begins phase 2 by enters into the g-transition state from the grey state. A grey node enters in the g-transition state will send out an UPDATE message. The intuition of the g-transition state for a non-MIS node is to probe the network and compete to see if it is suitable to behave as a connector. The node in the g-transition state may receive four types of message: UPDATE, BLUE, INVITE and WHITE. If a node in the g-transition state receives an UPDTE or WHITE message, it only ignores the message. Otherwise, if a node in the g-transition state receives an INVITE message, it implies that the competition succeeds and this node is invited to be a connector. Then the node will enter into the blue state to become a CDS node. When the node becomes blue as a CDS node, it broadcasts a BLUE message. If a node in the g-transition state receives BLUE messages from all its neighbors, it implies that all neighbors finish the CDS construction. Then the non-MIS node fails for becoming a connector and enters into the white state.

The Algorithm 2 continues until: 1) Any MIS node colored blue (i.e., becoming CDS node) terminates the algorithm. 2) Any non-MIS node terminates when it becomes blue (as a connector) or becomes white (enters back into initial state for the next duty cycle).

4.3 Performance evaluation

Lemma 1 *For an MIS node calculated by our algorithm in a network graph, there always exists that it has a non-MIS neighbor connecting with at least another MIS node.*

Proof Considering the propagation layer of our MIS algorithm, let B_i and G_i be the set of MIS and non-MIS nodes at ith layer. For any node $g \in G_i$ is a non-MIS node formed at the ith layer. In the phase 1 algorithm, it is marked grey (non-MIS node) from white state on receiving a BLACK message from its black neighbor (MIS node) in B_i. Next, after determining its state, the grey node g sends out a GREY message to all its neighbors in the $(i+1)$th layer. The neighbor finds itself with the highest weight among all its transition neighbors will become a black node in B_{i+1}. This implies that there always exists a non-MIS neighbor node $g \in G_i$ has at least two MIS neighbor nodes in B_i and B_{i+1} respectively. So for an MIS node in B_i, there always exists that it has a neighbor in G_i connecting at least another MIS node in B_{i+1}. □

Theorem 2 *The set of blue nodes computed by the phase* 2 *algorithm is a CDS of the network graph.*

Proof The set of blue nodes include MIS nodes and connectors. MIS is a dominating set, so we only need to proof the connectivity. Let $\{b_0, b_1, \ldots, b_n\}$ be the independent set, which elements are arranged one by one in the construction order. Let H_i be the graph over $\{b_0, b_1, \ldots, b_i\}$, $(1 \le i < n)$ in which pairs of nodes are interconnected by connectors. We prove connectivity by induction on j that H_j is connected. Since H_1 consists of a single node, it is connected trivially. Assume that H_{j-1} is connected for some $j \ge 2$. Considering message propagation layer in phase 1 algorithm, let B_{i-1} and G_{i-1} be the set of MIS and non-MIS nodes at the $(i-1)$th layer, respectively. The non-MIS node in G_{i-1} with maximal weight is selected as connectors in phase 2 algorithm. According to Lemma 1, it's enough to find non-MIS nodes, which interconnect B_{i-1} nodes at $(i-1)$th layer with B_i nodes in the ith layer. As H_{j-1} is connected, so must be H_j. Therefore the set of blue nodes computed by phase 2 algorithm is a CDS. □

Theorem 3 *Our distributed algorithm has* $O(n)$ *message complexity, and* $O(n)$ *time complexity.*

Proof In phase 1, for BLACK, GREY, and INQUIRY message, each node at most sends out once this kind of messages. Thus, the total number of these messages is $O(n)$. In phase 2, for BLUE, UPDATE, and INVITE messages, since each node sends a constant number of messages, the total number of messages is also $O(n)$. The time complexity of this algorithm is bounded by MIS construction, which has the worst time complexity $O(n)$. The worst case occurs when all nodes are distributed in a line and in either ascending or descending order of their weight. The rest of the process have time complexity at most $O(n)$. □

The following two important properties are listed in [17].

Lemma 2 *In a unit disk graph, every node is adjacent to at most five independent nodes.*

Lemma 3 *In any unit disk graph, the size of every maximal independent set is upper-bounded by* $3.8opt + 1.2$ *where opt is the size of minimum connected dominating set in this unit disk graph.*

Lemma 4 *In the phase* 2 *algorithm, the number of the connectors will not exceed* $3.8opt$, *where opt is the size of MCDS.*

Proof Let B be the independent set and S be the connector set of a graph. From Lemma 2: $|B| \leq 3.8opt + 1.2$. From Lemmas 1 and 2, it can be deduced that the number of MIS neighbors for a connector is ranged from 2 to 5. Let T be the dominating tree spanning black and blue nodes found by our algorithm. The worst case for T′ size occurs when all nodes are distributed in a line. By analyzing the utmost situation, the number of connectors must be less than the number of MIS nodes, i.e., $|S| \leq |B| - 1 \leq 3.8opt$. So the number of output connecting nodes will not exceed $3.8opt$. □

Theorem 4 *Our distributed algorithm has an approximation factor of not exceeding* 7.6.

Proof Our distributed algorithm includes two phases. Phase 1 is MIS construction, and phase 2 is CDS construction. From Lemma 3, the performance ratio in the first phase is 3.8. From Lemma 4, the performance ratio is 3.8 in the second phase, so the resulting CDS will have size bounded by 7.6. □

In our algorithm, the residual energy and size of nodes are both considered into CDS based backbone construction. The reconstruction makes the balance of energy consumption in networks as energy level changes. Our algorithm guarantees that the CDS nodes have good energy efficiency and extend the network lifetime.

5 Energy conservation nodes self-scheduling mechanism

5.1 Coverage metrics

We express the sensing coverage over the sensor field using a sensing probability model [12]. It denotes the sensing coverage of a sensor as a function of the distance between the sensor and the event. The probability sensing model can better reflect a sensor's sensing behavior than previous disk sensing model (i.e., sensor can only detect an event happening within a certain range). We use $p(g_i, s_k)$ to describe sensor s_k's sensing ability at the surveillance grid g_i. It implies that the coverage as the detection probability of a target at grid point g_i being detected by a node s_k. In the probability sensing model, $p(g_i, s_k)$ is a function of distance between s_k and g_i.

$$f(g_i, s_k) = \begin{cases} f(d_{i,k}) & d_{i,k} \le R_\mathrm{s} \\ 0 & d_{i,k} > R_\mathrm{s}. \end{cases} \tag{1}$$

The probability model coveys the intuition that the closer location to the sensing node, a higher signal-to-noise ratio is obtained, and a higher confidence level of the location being detected. Areas beyond the maximal sensing area R_s will be too noisy to be detected. We use the following function to represent the confidence level in the received sensing signal:

$$f(d_{i,k}) = 1/(1 + \alpha d_{i,k})^\beta. \tag{2}$$

In formula (2), α and β are parameters representing the physical characteristics of sensing unit.

Assume that S is the set of nodes that can detect grid pint g_i within node s_k's sensing range R_s. The sensing coverage of sensor set S in specific grid g_i is:

$$p(g_i, S) = 1 - \prod_{S_k \in S} (1 - p(g_i, s_k)), \tag{3}$$

A_k is the specific surveillance area. The sensing coverage of sensor nodes set S' (the set of nodes within A_k) in all grids of area of A_k is defined as the summation of the sensing coverage over the target gird from $g_0(0, 0)$ to $g_i(x, y)$ in area A_k:

$$p(A_k, S') = \int_0^x \int_0^y p(g_i, S')\, dx\, dy. \tag{4}$$

For the coverage requirement in the surveillance field G, the coverage conservation node self-scheduling problem is to schedule the sensor nodes and find a subset of sensing on-duty nodes to cover the whole sensor field of interest without losing detection probability on each grid point, i.e., for any A_k in G, S_a in S, $p(A_k, S_\mathrm{a}) \ge p_\mathrm{th}$.

p_{th} is the given event detection degree based on applications, and S_{a} is the on-duty sensor nodes within A_k. Only nodes in the subset S_{a} are actively performing the sensing task and the surveillance area is still covered with detection probability no lower than p_{th}.

5.2 Coverage redundancy

In the densely deployed sensor networks, the sensing areas may overlap with each other. In general, the larger the overlap of the sensing areas, the more redundant data may be generated and more power will be consumed. Since the existence of a sensor only affects the area covered by it, its coverage redundancy can be calculated by only considering the area of its sensing range (within R_{s}). So we consider A_k as specific surveillance area of local R_{s} area of sensor. We define the coverage redundancy $\xi_{i,k}$ for sensor node s_k in area A_k as the ratio of sensing coverage from all nodes that can detect in A_k except s_k itself to the coverage provided by s_k alone, when $p(A_k, \{s_k\}) > 0$:

$$\xi_{i,k} = \frac{p(A_k, S' \backslash \{s_k\})}{p(A_k, \{s_k\})}. \tag{5}$$

When $\xi_{i,k} = 0$, it implies there's no redundancy of event detection for sensor s_k in A_k. When $\xi_{i,k} > 0$, it implies grid g_i is partially redundant covered by sensor s_k. When $\xi_{i,k} \geq 1$, sensor s_k has complete coverage redundancy in A_k.

Due to application requirements, coverage maintenance problem still needs to consider certain fault tolerance. The wireless sensor network should be operated with a certain coverage redundancy. The required coverage redundancy is bounded by a threshold, denoted as ξ_{th}. For sensor s_k, if $\xi_{i,k}$ is no less than threshold ξ_{th}, it has the intuition that sensor s_k has enough coverage redundancy for detecting events provided by its neighborhood. So the working state of sensor s_k won't increase network coverage capability any more. If $\xi_{i,k}$ is smaller than threshold ξ_{th}, sensor s_k doesn't has the enough fault tolerance for the application and it should join the network event detection to preserve network coverage.

5.2.1 Sensor nodes self-scheduling

As discussed above, the objective of nodes self-scheduling is to minimize the number of working nodes, as well as maintain the sensing coverage and also with a certain redundancy for fault tolerance based on applications. Periodically, a node will determine whether it is eligible to turn off sensing units to save energy. So the nodes with sensing off-duty state eligibility rule should ensure that the entire network is covered with original detection probability and enough fault tolerance.

Definition The sensing off-duty state eligibility rule of nodes:

$$state_k = \begin{cases} \mathit{off} & \text{if } \xi_{i,k} \geq \xi_{\mathrm{th}} \text{ and } p(A_k, S_{\mathrm{a}}) \geq p_{\mathrm{th}} \\ \mathit{on} & \text{otherwise.} \end{cases} \tag{6}$$

Here A_k is the sensing area of sensor node s_k. After turning off the sensing units of candidate nodes, the original detection probability and application required coverage redundant degree are still ensured.

The election rule forms a network that roughly contain a subset of nodes maintain the network coverage with enough detection probability. The node with eligibility rule will be in sensing off-duty state and take the form of powering off its sensing unit.

ECSS applies a greedy strategy by gradually selecting sleep sensors in ascending order of detection ability and their residual energy. The gradual selection is to avoid that nods make the state decisions simultaneously and coverage blind points [17] may appear. We resolve the contention of nodes' off-duty sensing state selection with a random back-off delay time, and consider the above variety of factors in the derivation of the back-off delay. The first considered factor is the sensing coverage of sensor nodes. The nodes with higher detection capability have higher probability to be in on-duty state. The sensor nodes with less coverage detection ability should have shorter back-off delay. Another factor is that of unequal energy left at each node. The on-duty nodes consume more energy than off-duty ones. To balance energy load, the nodes with less energy left should be more reluctant for working and therefore should have a shorter delay. Let *E(k)* denote the energy already consumed at node s_k and $E_m(k)$ donate the initial energy of s_k. We define back-off delay function as follows, and γ is tunable weights subject to $0 \le \gamma \le 1$:

$$delay = \gamma \times \frac{p(A_k, S)}{p_{\text{th}}} + (1-\gamma) \times \left(1 - \frac{E(k)}{E_m(k)}\right). \tag{7}$$

In the self-scheduling phase, nodes investigate the eligibility rule and determine the sensing state of nodes. In the working phase, eligible nodes do not work in the duty cycle and turn off sensing unit. Each self-scheduling phase contain two steps. First, each node advertises its position and state. Only neighbors within a node's sensing range are considered in the basic model, so each node broadcast an ADVERTISE message with the minimum power as long as it reaches its sensing range. Each node listens to ADVERTISE messages from other nodes to obtain neighboring nodes' position and current sensing state, and maintains the above information in the neighbor list. Second, each node decides the state of sensing unit by the eligibility rule. The eligible nodes will broadcasts an ANNOUNCE message to notify its neighbors about the result. The ANNOUNCE message contention is resolved by delaying with a randomized back-off delay. Each node delays its determination for a random back-off time delay according to formula (7). Initially, all nodes are in the sensing on-duty state. A node collects all its neighbor sensing neighbor information will begin the initial duty cycle. The node begins a new duty cycle will set out a back-off delay time. During the delay time, the nodes are in the competition state. When the delay is due, the node will check the eligibility rule to decide its sensing state. If the node is eligible, it will turn off its sensing units entering off-duty state by broadcasts an ANNOUNCE message to notify all its neighbors. If the node is not eligible, it will maintain its original working state. When a node is in the competition state, if it receives an ANNOUNCE message from its neighbors, the node will recalculate its coverage metric, and adjust the back-off delay accordingly.

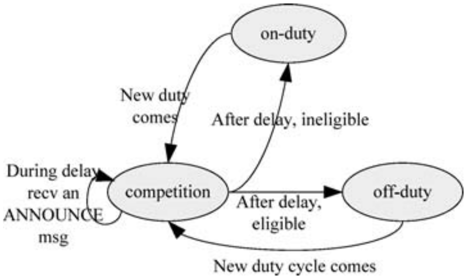


Fig. 4 Flowchart for ECSS scheme

When a new cycle begins, the node will set a new back-off delay and enter into the competition state. The flowchart of the whole ECSS scheme is illustrated in Fig. 4.

6 Simulations

In this section, we verify our framework in simulation by C++ programming and evaluation the performance on random network scenarios in terms of energy efficiency. We use ECDS based backbone to maintain connectivity, and integrate ECSS into the network framework to maintain coverage. We compare ECDS with Wu and Li's algorithm (WLA) and Wan, Alzoubi, and Frieder's algorithm (WAA), and compare ECSS with existing classic node-scheduling proposed by Tian et. al in [18].

The simulation network size is 100–300 numbers of nodes in increments of 50 nodes respectively, which are randomly placed in a 160 × 160 square area to generate connected graphs. Radio transmission range is 30 to 50 m. For the purpose to balance the choice of nodes in dominating set between those with high degree and high remaining energy, giving importance to both. We investigate the weight function as: $w(u) = \sqrt{degree(u)} * energy(u)$. Each node is assigned initial energy level 1 Joule (J). A simple radio model is used: E_{elec} is energy of actuation, sensing and signal emission/reception. E_{amp} is energy for communication, varies according to the distance d between a sender and a receiver. $E_{\mathrm{amp}} = \xi_{\mathrm{fs}}$, when $d < d_{\mathrm{crossover}}$, and $E_{\mathrm{amp}} = \xi_{\mathrm{mp}}$ when $d \geq d_{\mathrm{crossover}}$. The transceiver energy model: mimics a "sensor radio" with E_{elec} 50 nJ/bit, ξ_{fs} 10 pJ/bit/m^2, ξ_{mp} 0.0013 pJ/bit/m^4. Data fusion is omitted. The broadcast packet size is 25 bytes, data packet size 100 bytes, and packet header size is 25 bytes. The data routing takes flooding protocol. We take parameter all timeout in our algorithm as 100 ms, and when minimal energy of backbone cut down to 50% incurs a new backbone construction. In (2) parameters are chosen as $\alpha = 0.1$ and $\beta = 3$ respectively. We regard the sensing detection ability less than 6% as negligible. In (7) parameter are chosen as $\gamma = 0.5$. And the threshold parameter: $p_{\mathrm{th}} = 1.0$. The simulation makes average solutions over 50 iterations of random generating scenes.

Figures 5 and 6 show the size of the dominating set with the increasing number of nodes in the network for a certain transmission radius. ECDS has a good performance with smaller CDS size when comparing with WAA and WLA as the network size increases. Figure 7 shows the average CDS residual energy as the network size increases with $r = 50$ m for 300 working periods. ECDS achieves better energy efficiency with much higher residual power comparing with WAA and WLA. And WLA

Fig. 5 Size of CDS as the network size increases ($r = 30$ m)

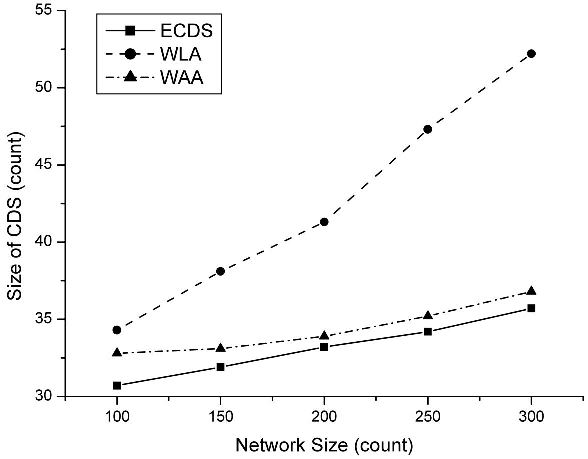
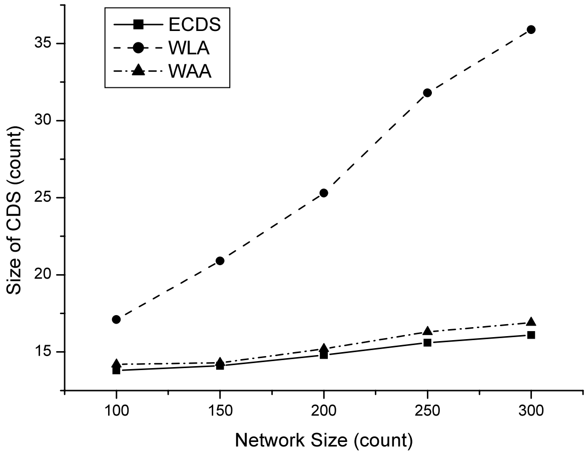


Fig. 6 Size of CDS as the network size increases ($r = 50$ m)



has the worst energy efficiency for its big size of dominating set. Figure 8 shows the network lifetime (the number of periods that the network survives until can't construct a backbone for the network) as the network size increases from 100 to 300 nodes when $r = 50$ m. ECDS has much better performance comparing with WAA and WLA. It can work with longest time until can't construct a backbone any more. Apparently, ECDS has better and more proper energy consumption when compared with the other two algorithms.

Figure 9 shows a plot of the off-duty node number. We can see that increasing the number of the original deployed nodes and decreasing the coverage redundancy will result in more nodes being off-duty. Figure 10 illustrates the average energy dissipation per node with 100-node in different coverage redundancy. The energy dissipation with ECSS scheme is slower than the situation without ECSS. And with the decreasing coverage redundancy according to application requirements, the energy dissipation is slower. It shows that our ECSS scheme do big help to save node

Fig. 7 Average CDS residual energy as the network size increase ($r = 50$ m)

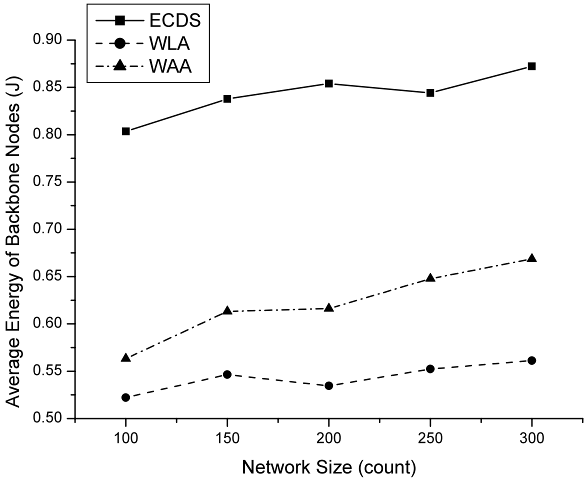
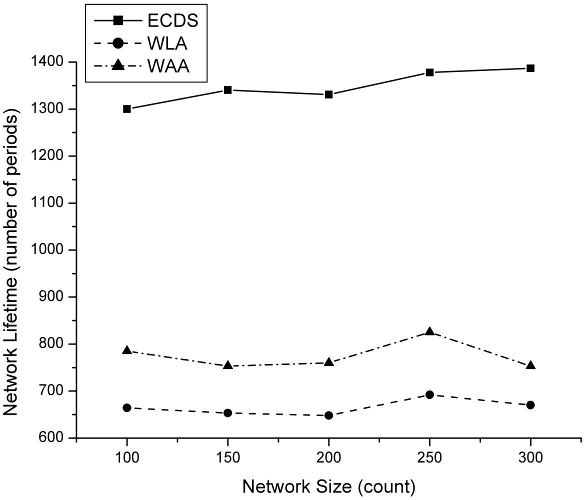


Fig. 8 Network lifetime as network size increases ($r = 50$ m)



energy. Figure 11 makes a comparison of the off-duty nodes ratio between ECSS and Tian et al.'s scheme. It shows that more nodes are in sensing off-duty state with ECSS scheme. So ECSS scheme can use the coverage redundancy efficiently and save more energy.

7 Conclusion and future work

In this paper, a distributed energy efficient connectivity and coverage preserved framework for wireless sensor networks is presented. The framework takes dynamically periodic reconstruction strategies to select new nodes for communication and sensing tasks when nods residual energy of network cut down to a threshold. Firstly,

Fig. 9 Off-duty size of ECSS in different coverage redundancy

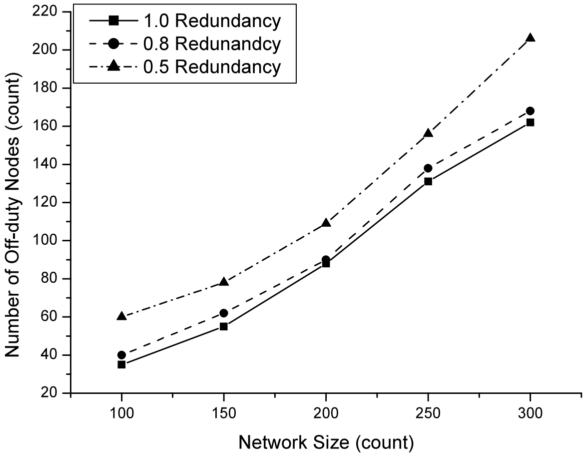
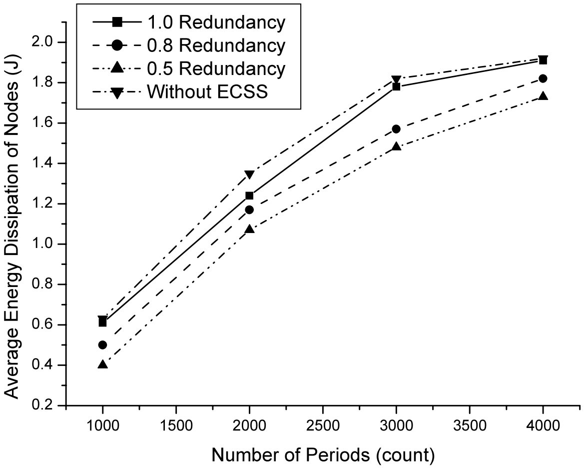
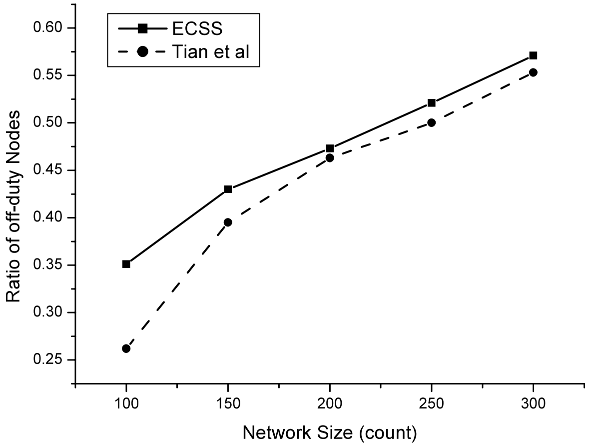


Fig. 10 Ave energy dissipation per node with 100-node in different coverage redundancy



we propose an energy-balanced connected dominating set backbone to extend network lifetime when maintaining network connectivity. Our designation can quickly construct a small-scaled backbone size. The time complexity and message complexity of this algorithm are both $O(n)$. The performance ratio is 7.6. Moreover, the algorithm is fully distributed, only uses simple local node behavior to achieve a desired global objective. Secondly, we also propose a distributed energy conservation node self-scheduling sensing coverage mechanism, which can reduce energy consumption when satisfying the application specific sensing coverage and redundancy. Our mechanism preserves the adequate surveillance degree according to the specific application when only selecting a subset of nodes in sensing on-duty state, and the other nodes will turn off sensing unit to save energy. The simulation results show that ECDS can efficiently prolong network lifetime and balance node energy consumption with a smaller backbone size, comparing with existing classic algorithms; and

Fig. 11 Comparison of the percentage of off-duty nodes for ECSS and Tian et al.'s scheme



ECSS scheme achieves considerable improvements on energy efficiency. The future work will focus on simulations and experiments under various settings.

Acknowledgement This work was supported by Kyungnam University Foundation Grant.

References

1. Acharya T, Roy R (2005) Distributed algorithm for power aware minimum connected dominating set for routing in wireless ad hoc networks. In: ICPP workshops, 2005, pp 387–394
2. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Commun Mag 40(8):102–114
3. Alzoubi KM, Wan PJ, Frieder O (2002) Distributed heuristics for connected dominating sets in wireless ad hoc networks. J Commun Netw 4(1):1–8
4. Bai X, Xuan D, Yun Z, Lai TH, Jia W (2008) Complete optimal deployment patterns for full-coverage and k-connectivity ($k \leq 6$) wireless sensor networks. In: MobiHoc'08, May 2008
5. Bai X, Yun Z, Xuan D, Lai TH, Jia W (2008) Deploying four-connectivity and full-coverage wireless sensor networks. In: IEEE INFOCOM, 2008
6. Butenko S, Cheng X, Oliveira CAS, Pardalos PM (2004) A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks. In: Cooperative control and optimization, 2004, pp 61–73
7. Chen B, Jamieson K, Balakrishnan H, Morris R (2001) Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In: MobiCom, 2001, pp 85–96
8. Clark BN, Colbourn CJ, Johnson DS (1990) Unit disk graphs. Discrete Math 86:165–177
9. Hsin C, Liu M (2004) Network coverage using low duty-cycled sensors: random & coordinated sleep algorithm. In: The 3rd international symposium on information processing in sensor networks, Berkeley, CA, April 2004
10. Lazos L, Poovendran R, Ritcey JA (2007) Probabilistic detection of mobile targets in heterogeneous sensor networks. In: IPSN, April 2007
11. Li H, Miao H, Liu L, Li L, Zhang H (2008) Energy conservation in wireless sensor networks and connectivity and connectivity of graphs. Theor Comput Sci 393:81–89
12. Lu J, Bao L, Suda T (2005) Probabilistic self-scheduling for coverage configuration in sensor networks. In: The international conference on sensing technology (ICST), Palmerston North, New Zealand, Nov. 2005

13. Min M, Huang CX, Huang SC-H, Wu W, Du H, Jia X (2004) Improving construction for connected dominating set with Steiner tree in wireless sensor networks. In: Global optimization, 2004
14. Pottie GJ, Kaiser WJ (2000) Wireless integrated network sensors. Commun ACM 43(5):51–58
15. Rozell CJ, Johnson DH (2007) Power scheduling for wireless sensor and actuator networks. In: IPSN, April 2007
16. Sivakumar R, Das B, Bharghavan V (1998) Spine-based routing in ad hoc networks. ACM/Baltzer Cluster Comput J 1:237–249
17. Stojmenovic I, Seddigh M, Zunic J (2002) Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. IEEE Int Conf System Sci 13(1):14–15
18. Tian D, Georganas ND (2002) A coverage-preserving node scheduling scheme for large wireless sensor networks. In: WSNA, 2002
19. Wan PJ, Alzoubi K, Frieder O (2002) Distributed well connected dominating set in wireless ad hoc networks. In: IEEE INFOCOM, 2002
20. Wang XR, Xing GL, Zhang YF, Lu CY, Pless R, Gill C (2003) Integrated coverage and connectivity and configuration in wireless sensor networks. In: ACM SenSys, 2003, pp 28–39
21. Wattenhofer R, Li L, Bahl P, Wang Y (2001) Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In: IEEE InfoCom, 2001
22. Woehrle M, Brockhoff D, Hohm T, Bleuler S (2008) Investigating coverage and connectivity trade-offs in wireless sensor networks. TIK-report No. 294, Oct. 2008
23. Wu J, Dai F, Gao M, Stojmenovic I (2001) On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. In: IEEE/KICS J Commun Netw, pp 346–356
24. Wu J, Li H (1999) On calculating connected dominating set for efficient routing in ad hoc wireless networks. In: The 3rd ACM int'l workshop on discrete algorithms and methods for mobile computing and communications, 1999, pp 7–14
25. Xu Y, Bien S, Mori Y, Heidemann J, Estrin D (2003) Topology control protocols to conserve energy in wireless ad hoc networks. Technical report 6, University of California, Los Angeles
26. Xu Y, Heidemann J, Estrin D (2001) Geography-informed energy conservation for ad hoc routing. In: MobiCom, Rome, Italy, July 2001, pp 70–84
27. Yan T, He T, Stankovic JA (2002) Differentiated surveillance for sensor networks. In: ACM workshop on wireless sensor networks and applications, Atlanta, GA, October 2002
28. Ye F, Zhong G, Cheng J, Lu S, Zhang L (2002) PEAS: A robust energy conserving protocol for long-lived sensor networks. In: The 10th IEEE international conference on network protocols, Paris, France, November 2002
29. Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. Comput Netw 52(12):2292–2330
30. Zou Y, Chakrabarty K (2005) A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks. IEEE Trans Comput 54(8):978–991

Yuanyuan Zeng received her B.Sc. and Ph.D. degree in Computer Science from Wuhan University, China in 2002 and 2007, respectively. She is currently a postdoctoral researcher in department of Computer Science, University College Cork, Ireland. Her research interests include sensor networks and wireless networks.

Cormac J. Sreenan joined UCC as a Professor in the Department of Computer Science in August 1999, and was Head of Department from 2000–2004. Prior to that he was a Principal Technical Staff Member with AT&T Labs Research (in Florham Park, NJ, USA), and he also worked at Bell Labs Research (in Murray Hill, NJ, USA). He studied for his Ph.D. at the University of Cambridge Computer Laboratory/Christ's College. During 2005 he completed a sabbatical at the University of New South Wales in Sydney where he was researching wireless and sensor networking with members of the Network Research Lab.

Naixue Xiong is a research scientist in Department of Computer Science, Georgia State University, USA. His research interests include Communication Protocols, Network Architecture and Design, Network Technologies, and Distributed and parallel Systems. Until now, Dr. Xiong published over 90 research articles (including over 30 international journal articles). Some of his works were published or submitted in IEEE or ACM transactions and IEEE INFOCOM. He has been a Program Chair, General Chair, Publicity Chair, PC member and OC member of over 60 international conferences, and was invited to serve as a reviewer for over 20 international journals. Now, he is serving as an Associate Editor, Editorial Board Member, and Guest Editor for 4 international journals.

Laurence T. Yang is a professor at Department of Computer Science, St. Francis Xavier University, Canada. His research includes high performance computing and networking, embedded systems, ubiquitous /pervasive computing and intelligence, autonomic and trusted computing, computational science and engineering. He has published around 240 papers in refereed journals, conference proceedings and book chapters in these areas. He has been involved in more than 100 conferences and workshops as the program/general conference chair and more than 200 conference and workshops as a program committee member. He served as the vice-chair of IEEE Technical Committee of Supercomputing Applications (TCSA) until 2004, currently is in the executive committee of IEEE Technical Committee of Scalable Computing (TCSC), and of IEEE Technical Committee of Self-Organization and Cybernetics for Informatics, and of IFIP Working Group 10.2 on Embedded Systems. He is also the cochair of IEEE Task force on Intelligent Ubiquitous Computing. In addition, he is the editors-in-chief of 10 international journals and few book series. He is serving as an editor for 14 international journals. He has been acting as the author or editor/co-editor of 30 books from Kluwer, Springer, Nova Science, American Scientific Publishers and John Wiley Sons.

Jong Hyuk Park received his Ph.D. degree in the Graduate School of Information Security from Korea University, Korea. He is now a professor at the Department of Computer Science and Engineering, Kyungnam University, Korea. Dr. Park has published any research papers in international journals and conferences. Dr. Park has been served as chairs, program committee, or organizing committee chair for many international conferences and workshops. Dr. Park is the founder of the International Conference on Multimedia and Ubiquitous Engineering (MUE), International Conference on Intelligent Pervasive Computing (IPC), and the International Symposium on Smart Home (SH). Dr. Park is editor-in-chief of the International Journal of Multimedia and Ubiquitous Engineering (IJMUE), the managing editor of the International Journal of Smart Home (IJSH). In addition, he has been served as a guest editor for international journals by some publishers. Dr. Park' s research interests include Digital Forensics, Security, Ubiquitous and Pervasive Computing, Context Awareness, Multimedia Services, etc.