TTS: a two-tiered scheduling mechanism for energy conservation in wireless sensor networks

Nurcan Tezcan* and Wenye Wang

Department of Electrical and Computer Engineering,
North Carolina State University,
Raleigh, NC 27695, USA
E-mail: ntezcan@ncsu.edu E-mail: wwang@ncsu.edu
*Corresponding author

Abstract: In this paper, we present a two-tiered scheduling approach for *effective* energy conservation in wireless sensor networks. The effectiveness of this mechanism relies on *dynamically* updated *two-tiered* scheduling architecture. We aim to prolong network lifetime, while preserving the major requirements of wireless sensor networks: coverage and connectivity. In this approach, sensors are periodically scheduled into sleep in two phases using weighted greedy algorithms that can be deployed either centralised or distributed. First, we establish a *coverage-tier* by selecting a set of sensors that fully covers the sensing field. Thus, sensors that are not selected for the coverage-tier, are put into sleep immediately. However, the coverage-tier sensors do not necessarily stay active all the time when events are not reported. Therefore, a second tier, called *connectivity-tier*, is formed to deliver data traffic to a sink node. Thus sensors, essential to coverage-tier but not in connectivity-tier may periodically sleep and become active only for sending new sensing measurement and receiving queries from the sink to preserve coverage for energy savings. In addition, periodically rotating the coverage and connectivity tiers is performed in order to maximise network lifetime and achieve fairness of energy consumption. Through extensive simulations in ns2, we demonstrate that the two-tier scheduling can reduce average energy consumption up to 40% while balancing the residual energy of sensors.

Keywords: wireless sensor networks; coverage; connectivity; energy conservation.

Reference to this paper should be made as follows: Tezcan, N. and Wang, W. (2006) 'TTS: a two-tiered scheduling mechanism for energy conservation in wireless sensor networks', *Int. J. Sensor Networks*, Vol. 1, Nos. 3/4, pp.213–228.

Biographical notes: Nurcan Tezcan received the BS and MS degrees in Computer Engineering from Yeditepe University, Turkey in 2001 and Bogazici University, Turkey, in 2004, respectively. She is currently working towards the PhD in Computer Engineering at the North Carolina State University, Raleigh. Her current research interests are in transport layer protocols, energy-efficient algorithms and performance analysis of wireless networks.

Wenye Wang received the BS and MS degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 1986 and 1991, respectively. She also received the MSEE and PhD from Georgia Institute of Technology, Atlanta, Georgia in 1999 and 2002, respectively. She is now an Assistant Professor with the Department of Electrical and Computer Engineering, North Carolina State University. Her research interests are in mobile and secure computing, Quality-of-Service (QoS) sensitive networking protocols in single- and multi-hop networks.

1 Introduction

In a Wireless Sensor Network (WSN), a large number of sensor nodes, each having limited battery power, monitor the events of interest queried by the sink. In many applications, sensor nodes are densely deployed and transmit their data to the sink in an event-driven or continuous manner. In such dense networks, energy-efficient scheduling is a key factor to extend the functionality and lifetime of the network. That means, only the nodes maintaining the functionality stay active whereas others are scheduled to sleep, for example, switching to power saving mode. Therefore, the energy dissipation in sending/receiving and idle time can be significantly reduced and by updating the sleeping nodes, network lifetime can be prolonged.

The fundamental challenge of scheduling is to maximise the number of sleeping nodes to conserve more energy while maintaining the functionality of the WSN. For this purpose, several approaches have been proposed that make use of topological information which can be categorised into three groups:

1 *connectivity preserving* scheduling schemes (Cerpa and Estrin, 2002; Chen et al., 2001; Heinzelman et al., 2002; Xu et al., 2001)
2 *coverage preserving* scheduling schemes (Cardei et al., 2005; Slijepcevic et al., 2001; Tian and Georganas, 2002) and
3 *connectivity and coverage* preserving scheduling (Gupta et al., 2003; Wang et al., 2003).

Connectivity preserving schemes have been proposed to put nodes into sleep mode based on their communication neighbourhood. On the other hand, coverage preserving scheduling mechanisms have selected nodes for full coverage based on their sensing ranges. For example, the sensing range of a sensor node might be approximately in between *1* and *30 m*, whereas the transmission range of that sensor might be in between *150* and *300 m* (Zhang and Hou, 2004). Even though the coverage might imply connectivity under given conditions (Wang et al., 2003), more nodes stay active in coverage schemes than in connectivity preserving schemes, because the nodes which are essential to coverage are not necessarily to be active all the time. Instead, some may wake up periodically to send their sensing measurements and receive queries, and then go back to sleep. Similarly, when we integrate connectivity and coverage for scheduling, at least, the minimum number of nodes preserving coverage must stay active (Gupta et al., 2003; Wang et al., 2003).

This work differs from existing scheduling mechanisms in various aspects. Recent scheduling schemes have classified sensors as either active or sleeping nodes. In this work, we integrate coverage and connectivity by a tiered approach; thus, nodes having been used for connectivity or coverage have different sleeping behaviours during an update interval. Nodes, which are not selected for coverage or connectivity-tier, are put into *sleep* immediately; nodes in the coverage-tier are put into *semi-sleep* because they can wake up for sending data and can go back to sleep mode periodically; nodes in the connectivity-tier stay *active* in order to forward data traffic. Hence, we enable more nodes to sleep while maintaining the coverage and connectivity of the network.

The contributions of this paper can be summarised as follows. We propose a two-tiered scheduling mechanism using *weighted-greedy* algorithms for efficient energy conservation, which can be deployed either centralised or distributed. First, *coverage set* is established; in each step of the algorithm of establishing the coverage set, an unused sensor, covering the largest uncovered area and having higher residual energy, is chosen as an *Essential node* (E-node) for the coverage set. Nodes in the coverage set can monitor the entire sensing field and periodically wake up to send and receive to/from the sink. Therefore, we guarantee that an event can be detected by at least one node in the coverage set and queries sent by the sink can affect the entire sensing field. The nodes that are not selected for coverage set are called Non-Essential nodes (N-nodes) and put into sleep mode. Second, *connected dominating set* is selected among the nodes in the coverage set according to the residual energy and the network connectivity. The *Essential Dominating nodes* (ED-nodes), selected from the coverage set, stay active to forward the traffic, whereas others are in sleep mode. In this process, we attempt to reduce maximum number of nodes while ensuring that the remaining network is connected.

Further, to balance the energy consumption while scheduling, coverage and connectivity tiers are updated dynamically. In every round, greedy algorithms are reperformed to establish the new set of coverage set and dominating set by maximising the total residual energy of selected nodes. Then, a dominating node, whose energy consumption is high, might be a non-essential node for the next round. This dynamic update process also helps handling the topology changes due to unexpected node failures. However, the cost for deploying the centralised algorithm in sensor network may be more involved than the distributed approach and may vary greatly between applications. Hence, we discuss an alternative *distributed* implementation of the algorithms where sensors use local neighbouring information to establish the coverage and connectivity sets.

The remainder of the paper is organised as follows. In Section 2, we summarise existing research works related to energy-conservation scheduling. The problem formulation is given in Section 3. We describe the two-tiered scheduling mechanism in detail in Section 4, whereas an alternative distribution implementation is discussed in Section 5. Following, simulation results are presented in Section 6. Finally, Section 7 concludes the paper.

2 Related work

Energy conservation and power management is one of the most important design issues in sensor networks. Thus, there exist many research work on energy-efficient MAC protocols (Ye et al., 2002), energy-aware routing protocols (Chang and Tassiulas, 2004) and network topology control by scheduling nodes to switch on/off their radios. Our approach takes the advantage of topological information to decide the set of nodes to accomplish the necessary functions of sensor networks.

The prior works that use topological information for scheduling can be classified into three groups as

1 *connectivity preserving* scheduling schemes

2 *coverage preserving* scheduling and

3 *connectivity and coverage* preserving scheduling schemes.

In *connectivity* preserving scheduling, network topology is formed based on the connectivity of the network. For example, in GAF (Xu et al., 2001), sensing area is divided into grids, thus one sensor stays active for each grid, whereas other sensors are put into the sleep mode. Grid size is defined based on the transmission range of nodes. SPAN (Chen et al., 2001) is presented as a distributed algorithm for ad hoc networks to form a coordinator backbone of active nodes. It attempts to minimise the number of coordinators ensuring that enough coordinators are elected so that every node is in the radio range of at least one coordinator. Another energy-efficient approach in this group is Low-Energy Adaptive Clustering Hierarchy (LEACH) (Heinzelman et al., 2002), where sensors in a cluster may sleep in predefined time slots. LEACH includes distributed cluster formation as well as rotation to distribute the energy load among all the nodes.

Furthermore, scheduling of nodes based on *coverage* has been addressed as a way of conserving energy (Cardei et al., 2005; Slijepcevic et al., 2001; Tian and Georganas, 2002). The goal of these methods is to organise sensors to preserve the sensing coverage without leaving blind points in the sensing field. Therefore, only the sensors covering the field stay awake while others are put into sleep mode.

In Cardei et al. (2005) and Slijepcevic et al. (2001), coverage is achieved by forming multiple set-covers that are activated consecutively. To achieve this, sensor networks should be dense enough to form multiple independent set of sensors that can monitor to entire field. In Cardei et al. (2005), rather than sensing field, a set of targets with known locations are necessarily covered by each set cover. Therefore, only do the active nodes in the set cover send and receive data. In Tian and Georganas (2002), a distributed scheduling algorithm has been proposed where each node turns itself off using local neighbour information.

In addition to aforementioned studies, there exist attempts to use clusters for energy-efficient communications (Heinzelman et al., 2002; Younis and Fahmy, 2004). In Heinzelman et al. (2002), a distributed clustering protocol, HEED, has been presented where cluster heads coordinate the communication among the nodes within their clusters and communicate with other cluster heads. In HEED, cluster heads also aggregate the received information. However, this approach attempts to reduce energy dissipation due to communications, not for scheduling node behaviours. Moreover, the two-tiered approach is used by Pan et al. (2003) for topology control by locating base stations on the optimal positions to maximise the lifetime of the WSN.

We compare our work via simulation, with the state-of-the-art connectivity and coverage preserving schemes (Gupta et al., 2003; Wang et al., 2003). In Gupta et al. (2003), a greedy algorithm is proposed to construct a logical topology in response to a query that maintains both coverage and connectivity. The area which is necessarily monitored can be a subregion where a query will be executed. Hence, the objective of this algorithm is to find a set of connected sensors in the query region. The relation between coverage and connectivity is also analysed in Wang et al. (2003). In this work, a coverage configuration protocol which combines with SPAN (Chen et al., 2001) has been proposed. Differing from (Gupta et al., 2003), it finds a connected sensor cover for the entire field. Similarly, sensors are *active* if they satisfy the eligibility rule of neither SPAN (Chen et al., 2001) nor coverage; whereas others are *inactive*. Also, the required ratio of transmission range to sensing range is proved for coverage to imply connectivity in Wang et al. (2003).

Besides existing works, two-tiered scheduling approach decomposes the two functionalities, *coverage* and *connectivity*, such that connected dominating backbone can be built among sensors providing the coverage. Such a decomposition allows us to schedule more nodes to be in power-savings mode, thus conserving more energy.

3 Problem formulation

Let $\mathbf{S} = \{s_1, s_2, s_3, \ldots, s_N\}$ be the finite set of sensors, distributed randomly in a two-dimensional area $\mathbf{A}$, where there are sufficient sensors to monitor the field. Each sensor s_i has a unique *identifier* (such as MAC address). We also assume that each node is equipped to learn its location information via any lightweight localisation technique for wireless networks (Cheng et al., 2004). Therefore, all sensor nodes and the sink know their location coordinates (x_i, y_i), *sensing range* r_i^s, and *transmission range* r_i^t. Transmission range is assumed to be at least as twice as sensing range which is the case for many sensor nodes (Wang et al., 2003). All nodes have similar processing and communication capabilities; messages are sent in a multihop fashion.

In this context, sensor network can be represented as an undirected graph $G(\mathbf{S}, E)$, where $\mathbf{S}$ is the set of sensors, and E is the set of edges. When sensor s_j is within the transmission range of sensor s_i, then edge (s_i, s_j) is in E.

3.1 Coverage and connected dominating sets

The *sensing region* R_i of a node s_i is the circular area with its center at (x_i, y_i) and radius of r_i^s. A subset of sensors, $\mathbf{C} \subseteq \mathbf{S}$ is called a *coverage set* if the union of the sensing regions of the $s_i \in \mathbf{C}$ covers the entire field $\mathbf{A}$, that is $\mathbf{A} \subseteq \bigcup_{s_i \in \mathbf{C}} R_i$. We consider a sensor node to be an *essential* (E) node in $\mathbf{C}$ if $s_i \in \mathbf{C}$. This E-node is referred to as $s^{(E)}$. Otherwise, it is a N node, $s^{(N)}$.

Given the sensor network $G(\mathbf{S}, E)$ with the set of sensors $\mathbf{S}$ and the set of edges E, a Connected Dominating Set (CDS), denoted by $\mathbf{D}$, is a connected set of E-nodes ($\mathbf{D} \subseteq \mathbf{C}$), where each E-node $s_i^{(E)} \in (\mathbf{C}/\mathbf{D})$ can directly communicate with one of the sensors in $\mathbf{D}$. Our goal is to construct a connected dominating set having minimum number of dominating nodes $\in \mathbf{D}$. We consider a sensor node to be an *Essential Dominating* (ED) node in $\mathbf{D}$ if $s_i \in \mathbf{D}$. This ED-node is denoted by $s^{(\mathrm{ED})}$.

In this paper, time is divided into rounds, denoted by T_R. Each round is composed of *classification update interval* T_CU and *network operation interval* T_NO. The coverage and the dominating sets are updated periodically every round in T_CU. We should ensure that T_CU is much smaller compared to T_NO because short T_CU implies less overhead and better performance of the network. However, a long T_NO may cause high variance in sensors residual energy as E-nodes and ED-nodes may drain out their battery much faster, thus partitioning the network. The effects T_NO on network lifetime and energy consumption is discussed in Section 6.

Next, we will explain the energy model and network lifetime.

3.2 Energy model and network lifetime

A wireless sensor can operate in one of transmitting, receiving, idle or sleep modes in a network. Recent works have shown that energy consumption of being idle is dramatically higher compared to energy drain in sleep mode. For example, typical power consumption of the Mica Mote is in Table 1. By scheduling of nodes, we may significantly reduce the energy consumption of being idle.

The total power consumption of the radio of a sensor node is a function of reception, $p_r(t)$, transmission, $p_t(t)$, being idle, $p_\mathrm{idle}(t)$ or being in sleep mode, $p_\mathrm{sleep}(t)$. The average reception and transmission power of a sensor node can be written as:

$$p_t(t) = r_t(t)E_b^t, \quad \text{and} \quad p_r(t) = r_r(t)E_b^r \tag{1}$$

where $r_t(t)$ is the average transmission rate at which the sensor transmits; $r_r(t)$ is the average reception rate of

data; E_b^t and E_b^r are transmission and receiving energy per bit, respectively, depending on modulation and coding schemes.

Table 1 Typical power consumption of the Mica Mote (Hill and Culler, 2002)

Tx	*Rx*	*Idle*	*Sleep*
24 mW	13 mW	13 mW	0.01 mW

Then, the residual energy of a sensor s_i in the beginning of the kth round, when it is an N-node, E-node and ED-node, is as follows, respectively:

$$e_i^{(\mathrm{N})}(kT_R) = e_i((k-1)T_R) - \int_{T_{\mathrm{CU}}}^{T_R^-} p_{\mathrm{sleep}}(t)\mathrm{d}t$$

$$e_i^{(\mathrm{E})}(kT_R) = e_i((k-1)T_R) - \int_{T_{\mathrm{CU}}}^{T_R^-} \{\alpha_1 p_{\mathrm{sleep}}(t) + \beta_1 p_t(t) + \eta_1 p_r(t)\}\mathrm{d}t,$$

$$e_i^{(\mathrm{ED})}(kT_R) = e_i((k-1)T_R) - \int_{T_{\mathrm{CU}}}^{T_R^-} \{\alpha_2 p_{\mathrm{idle}}(t) + \beta_2 p_t(t) + \eta_2 p_r(t)\}\mathrm{d}t$$

where $e_i((k-1)T_{\mathrm{R}})$ is the residual energy of the sensor in the beginning of round $k-1$; α_1 and α_2 are being in sleep and idle mode ratios during T_{NO}; β_1, β_2 are being in transmitting mode ratios of E-nodes and ED-nodes, and η_1, η_2 are being in receiving mode ratios of E-nodes and ED-nodes, respectively.

The process of selecting E-nodes and ED-nodes are triggered periodically. In each round a new coverage set is formed consisting of nodes having higher residual energy. We consider a WSN as alive when the sensing field is fully covered. In other words, a network is alive when every point in **A** is covered by at least one sensor. Then, network lifetime is defined to be the time from sensors are deployed until when the $(N - ||\mathbf{C}_{\min}|| + 1)$th node fails, where $\mathbf{C}_{\min}$ is the size of the minimum coverage set during the lifetime:

$$L = \max\{l_i | s_i \in \{N - ||\mathbf{C}_{\min}|| + 1\}\} \quad (2)$$

where l_i is lifetime of sensor s_i. Therefore, the objective of two-tiered scheduling is to reduce the number of active or on-duty nodes, and eventually prolong the network lifetime.

3.3 Two-tiered scheduling problem

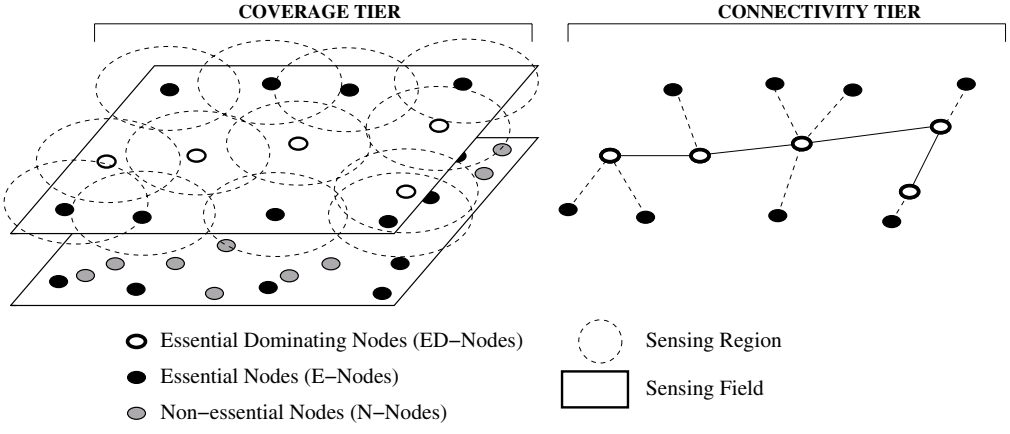
The main idea of two-tiered scheduling problem is to decompose the main functionalities of the WSN into *coverage-tier* and *connectivity-tier* as shown in Figure 1. Such a decomposition allows us to schedule more nodes to be in power-savings mode, thus conserving more energy. If the coverage-tier does not exist, the proposed mechanism works like an energy-efficient topology control. On the other hand, if the connectivity-tier does not exist, it becomes a coverage preserving node scheduling scheme.

Particularly, in our two-tiered scheduling architecture, sensors are classified into three groups with different sleeping behaviours. The first group of nodes, called E-nodes, are selected to maintain the coverage thus, they should be active to send/receive to/from the sink for some periods, and then may go to sleep. Sleeping behaviour of E-nodes are called *semi-sleep* because they can be in active/sleep mode during a round as shown in Figure 2. The second group nodes are N-nodes which are scheduled to sleep until the next round without serving on the coverage tier. The third group of nodes, ED-nodes, which forward the data to sink are active because they are selected from the coverage set and serve in the connectivity-tier. Figure 2 summarises the sleeping behaviour of these different group of sensors.

Before describing the details of the proposed algorithms, some important aspects of the two-tiered scheduling scheme are explained as follows:

- Coverage is provided by E-nodes, which periodically wake up to send their measurement to the sink and receive querying from the sink. E-nodes also form a connected network since transmission range is assumed to be at least as twice as sensing range which is the sufficient condition of coverage that implies connectivity (Wang et al., 2003).
- However, all E-nodes are not necessarily be active all the time. Some E-nodes may be semi-sleep such that they may wake-up for collecting event data from time to time. Therefore, only a small number of them can be active as a backbone to forward the data traffic and delivery tasks sent by the sink. To achieve this, we establish a CDS among coverage set where an E-node is either a dominating node or a direct (one-hop) neighbour of a dominating node. The dominating nodes

Figure 1 Logical view of two-tiered scheduling

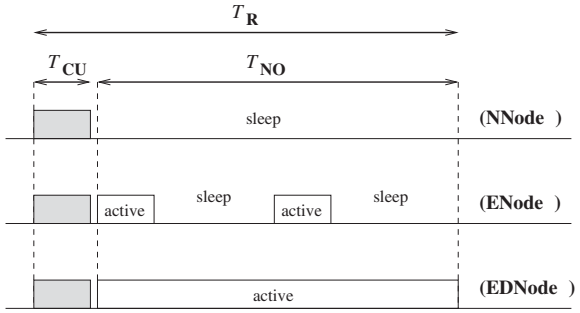


always stay active to preserve the connectivity of the network and forward the data traffic to/from the sink. E-nodes can communicate at least with one ED-node and send/receive their measurement/query via their neighbouring ED-nodes.

- Based on the sensing ranges of nodes, to provide full coverage, sensors should be densely deployed compared to the schemes which consider the connectivity of the network for topology control. Our approach is designed for fully covered networks.
- The active/sleep period of E-nodes are predetermined based on the WSN application. If an event can be detected frequently, then the sleep/wakeup period of an E-node should be shorter. Therefore, detected event can be reported immediately via ED-nodes.

Next, we give the details of the algorithms to perform this two-tiered scheduling mechanism.

Figure 2 Sleep schedules of sensors: N-nodes are always in sleep state, ED-Nodes are always active and E-nodes are semi-sleep



4 Two-tiered scheduling mechanism

In this section, we give the details of the algorithms to perform two-tiered scheduling under the centralised control of the sink. Firstly, we explain how the coverage-tier is established. Secondly, we describe the algorithm for connectivity-tier establishment. Finally, we discuss the update process of coverage and connectivity-tiers, followed by a walk-through example.

4.1 Establishment of the coverage-tier

We first present a weighted greedy algorithm used to establish the *coverage set* which will be sufficient to detect all events of interest in the entire sensing field. In order to choose the coverage set, an ideal solution would be to find the *minimum* number of sensors that cover the entire field. However, it is an NP-hard problem similar to the well-known set cover problem. The goal in set cover problem is to find a set with the smallest possible number of subsets given a ground set of elements (Slavik, 1996). Due to this reason, we use a greedy approach to find an *near optimal coverage set* running in polynomial time.

For different purposes, previous studies focused on the problem of finding near-optimal coverage in WSNs (Gupta et al., 2003; Wang et al., 2003). However, we choose the coverage set of sensors to maximise the *benefit* in terms of coverage and the residual energy, that is, the largest uncovered sensing region is covered with the least sensors. As a result, our approach is to cover the entire field with minimum number of sensors having maximum residual energy.

We propose a weighted-greedy algorithm based on residual energy to find a near optimal coverage set, given in *Algorithm 1*. In each step, *Algorithm 1* selects one node from the unselected sensors which covers the largest area with highest residual energy level. For this purpose, *weight* is defined to represent the weight of a sensing region of a sensor based on its residual energy. For a given region, the weight based on the residual energy level of a sensor is:

$$w(i,\ R_i) = e_i[|R_i|] \tag{3}$$

where e_i is the energy level given in Table 2 and $[|R_i|]$ is the area of sensing region R_i.

Table 2 Notations

Symbol	*Description*
S	The set of sensors in the WSN
C	Coverage set
D	Connected dominating set
A	Sensing field of the entire WSN
r_i^s	Sensing range of node s_i
R_i	Sensing region of node s_i
r_i^t	Transmission range of node s_i
$e_i(t)$	Residual energy at time t
$e_i(0)$	Initial energy of node s_i
l_i	Lifetime of node s_i
L	Network lifetime of a sensor network

Then, we calculate the *coverage-benefit* in selecting sensors to the coverage set using the weight function. To do this, we first find the size of the area that can be covered by sensor s_i and has not been covered yet. Consider the sensor s_i with sensing region R_i. Let $R_{\mathbf{C}}$ be the area that sensors of **C** covered so far, that is, $\bigcup_{s_j \in \mathbf{C}} R_j$. *Beneficial area* of s_i is defined to be the region inside the sensing field which has not been covered, that is, $R_{\mathbf{C}} = R_i \cap \mathbb{A})/R_{\mathbf{C}}$. Hence, *coverage-benefit* of sensor s_i is the total weight of its beneficial area, which is given as:

$$\text{Benefit}^{(C)}(s_i) = w\left(i, \frac{(R_i \cap \mathbb{A})}{R_{\mathbf{C}}}\right) \tag{4}$$

where R_i is the sensing region of sensor s_i and $R_{\mathbf{C}}$ is the total region covered by the sensors in **C**.

In the initial step of *Algorithm 1*, all nodes are candidates and the coverage set **C** is empty. Then, in each iteration (lines 2–8), the algorithm chooses an unselected node that has the maximum benefit. After selecting a node, coverage-benefits of remaining sensors are recalculated for the next iteration (lines 3–5) because by adding a new node to **C**, uncovered area in **A** shrinks gradually. This operation continues until **A** is fully covered.

Algorithm 1 Selecting E-nodes for the coverage-tier.

Input: **S**
Output: **C**

1 $\mathbf{C} \leftarrow \emptyset$
2 **while** $(\mathbf{A} \supset \bigcup_{s_i \in \mathbf{C}} R_i)$ **do**
3 **for all** $s_i \in \mathbf{S}/\mathbf{C}$ **do**
4 $benefit^{(C)}(s_i) \leftarrow w(i,\ (R_i \cap \mathbb{A})/R_{\mathbf{C}})$
5 **end for all**
6 *select* s_i *from* $\mathbf{S}/\mathbf{C}$ *having* $\max\{benefit\}$
7 $\mathbf{C} \leftarrow s_i \cup \mathbf{C}$
8 **end**
9 *return* **C**

Since *Algorithm 1* is to find a near-optimal coverage set, let us take a look how the proposed algorithm approximates an optimal coverage set.

Lemma 1: *Algorithm 1 gives a coverage set where the total weight of the entire field is $O(ln(N))$-factor of the optimal solution, where N is the number of sensor nodes.*

Proof: Let τ be the unit area and $\mathcal{A}$ be the size of the sensing field in terms of unit τ. *Algorithm 1* terminates when the sensing area of size $\mathcal{A}$ is fully covered. Consider the worst case where all N nodes have the minimum overlapping sensing regions covering the field, then all nodes will be selected as E-nodes.

Let each unit area have a *price* defined as follows:

$$\text{price}(\tau) = \{e_i \mid \tau \in R_i,\ s_i \in \mathbf{C}\}.$$

Algorithm 1 attempts to cover the entire field by maximising the total weight, which is also equal to the summation of the price of each unit area in the sensing field, that is, $\sum_{j=1}^{\mathcal{A}} \text{price}(\tau_j)$. At the jth iteration, the remaining uncovered area can be covered by a total weight of at most $\text{OPT}/\mathcal{A} - j + 1$, where OPT is the total weight of the optimal solution. Then we can write:

$$\sum_{j=1}^{\mathcal{A}} \text{price}(\tau_j) \leq \sum_{j=1}^{\mathcal{A}} \frac{\text{OPT}}{\mathcal{A}-j+1} = \text{OPT} H_{\mathcal{A}}$$

where $H_{\mathcal{A}}$ is harmonic number. Therefore, *Algorithm 1* finds an E-node set that covers the entire field at the cost of $O(ln\mathcal{A})$-$factor$ of the optimal solution.

Consider a network with a total number of N sensors with sensing range r. When the sensors are placed such that overlapping sensing areas are minimum, the size of sensing field will be at most $\sqrt{27}N(r)^2/2$ under the assumption of fully coverage (Williams, 1979). Thus, the factor of the optimal total weight is obtained as $O(ln(N))$ for fixed sensing ranges. A loose bound of the running time of *Algorithm 1* is polynomial with upper bound $O(N^2)$.

In the worst case, **S** is the minimum coverage set, thus all nodes are selected as E-node. In this case, number of iterations (lines 2–8) in *Algorithm 1* will be N. Since the number of iterations (lines 2–8) is $O(N)$, the running time of *Algorithm 1* is polynomial with upper bound $O(N^2)$.

4.2 Establishment of the connectivity tier

In the second phase, we select a *connected dominating set* **D** from the coverage set **C**, where all other nodes in $\mathbf{C}/\mathbf{D}$ can directly communicate with a dominating node, that is, ED-node. The most effective approach to conserving energy is to establish the *Minimum Connected Dominating Set (MCDS)*, which is NP-hard as well as finding CDS (Garey and Johnson, 1990). Thus, similar to the coverage set, we use a weighted-greedy algorithm to find a near optimal CDS.

Since CDS may widely be used in many applications in wireless networks, there have been many research work that has proposed different solutions (Alzoubi et al., 2002; Butenko et al., 2004). We use a similar greedy heuristic method by Butenko et al. (2004). However, in our algorithm, we again consider the residual energy as *benefit* of sensors and aim to maximise the total benefit while conserving connectivity. In this phase, benefit function is based on the *residual energy* and the *degree of connectivity*. Therefore, nodes, having higher residual energy and degree of connectivity, may have a better chance of being ED-nodes.

After implementing the *Algorithm 1*, we have a coverage set which is also connected based upon the assumption that coverage implies connectivity when $r^t \geq r^s$ (Wang et al., 2003). Hence, the CDS set for coverage-tier is obtained in the first tier, which is composed of E-nodes. The objective of *Algorithm 2* is to reduce the number of nodes in the CDS and select dominating nodes, that is, ED-nodes for the connectivity-tier. For this purpose, we start selecting a node having minimum connectivity and decide either to remove it from the CDS or set it as an ED-node.

A node can be removed from the CDS if and only if the remaining set is still connected. For this reason, we start checking nodes with the minimum connectivities. Also, while removing a node, we have to ensure that at least one of its neighbours has already been assigned as a dominating node. Otherwise, we select one of its neighbours to be a dominating node. In this step, we use the *connectivity-benefit* to select a neighbour having maximum benefit to be set as an ED-node. This operation continues until all possible nodes in CDS are removed.

Let $\mathbf{N}_i$ be the set of one-hop neighbours of sensor s_i. We define the *connectivity-benefit* of sensor s_i as:

$$\text{Benefit}^{(D)}(s_i) = e_i||\mathbf{N}_i/\mathbf{D}|| \tag{5}$$

where $(\mathbf{N}_i/\mathbf{D})$ represents the neighbours of sensor s_i that are not currently included in **D**.

In the pseudocode of the *Algorithm 2*, **L** is used to denote the current CDS and set **D** denotes the CDS that will be returned at the end. In each iteration, (line 8), the sink checks if the current CDS is connected or not. That is, we examine whether there is a node in $\mathbf{L}/\mathbf{D}$ that can be removed such that $\mathbf{L} - s_i$ is still connected. When a sensor is added to **D**, it is assigned as an ED-node. Note that, the sink is a default member of CDS. *Algorithm 2* terminates when **D** and **L** are equivalent. After *Algorithm 2* terminates, nodes in **C**, are either dominating node in **D** or the neighbour of a dominating node.

In the implementation of the algorithm, to check whether the set is connected, we simply use *Depth-First-Search (DFS)*. We test whether all nodes are visited starting from a random dominating node. The running time of DFS is $O(||\mathbf{C}||+||E^{'}||)$, where $E^{'} \subseteq E$ is the set of edges belonging to the E-nodes, $E^{'} = \{(s_i, s_j)|s_i \in \mathbf{C}, s_j \in \mathbf{C}\}$. Consider the worst case in which any node subtraction from CDS

may cause the remaining set disconnected. In this case, all nodes should be added to **D** (line 16) before the algorithm terminates. In this case, the running time of *Algorithm 2* for establishing the CDS is polynomial time with an upper bound $O(||\mathbf{C}||^2)$.

Algorithm 2 Selecting ED-Nodes for Connectivity-Tier.

Input: **C**
Output: **D**

1 $\mathbf{D} \leftarrow \{sink\}$, $\mathbf{L} \leftarrow \mathbf{C} \cup \{sink\}$
2 **while** $(\mathbf{L}/\mathbf{D} \neq \emptyset)$
3 **for all** $s_i \in \mathbf{L}/\mathbf{D}$ **do**
4 $benefit^{(D)}(s_i) = e_i \cdot ||\mathbf{N}_i \cap \mathbf{L}||$
5 **end for all**
6 *select* s_i *from* $\mathbf{L}/\mathbf{D}$ *having* $\min\{||\mathbf{N}_i||\}$
7 **if** *(*$\mathbf{L} - s_i$ *is connected)*
8 $\mathbf{L} \leftarrow \mathbf{L} - s_i$
9 **if** *(*$\mathbf{N}_i \cap \mathbf{D} == \emptyset$*)*
10 *select* s_j *from* $\mathbf{N}_i$ *having* $\max\{benefit^{(D)}(s_i)\}$
11 $\mathbf{D} \leftarrow s_j \cup \mathbf{D}$
12 **end if**
13 **end if**
14 **else**
15 $\mathbf{D} \leftarrow s_i \cup \mathbf{D}$
16 **end**
17 *return* **D**

4.3 Updating coverage and connectivity Tiers

The energy consumption of ED-nodes may be higher than the E-nodes; and N-nodes may have the lowest energy consumption due to continuous sleep. Thus, to acquire a fair energy consumption among sensors, coverage and connectivity sets should be updated throughout the lifetime of the network.

In TTS, we make use of *global update* where all E-nodes and ED-nodes are re-selected independent from the current set. In particular, global update is the process of repeating classification algorithms with latest residual energy levels of sensors. By this way, sensors that have overlapping regions and were E-nodes in the previous round might be N-nodes in the next update because more energy has been consumed when they were E-node before. This is used to acquire fairly distributed energy consumption among sensors.

Sink can monitor up-to-date energy reserves of sensors using a *energy monitoring* scheme (Zhao et al., 2002). Based on this remaining energy of sensors, a new coverage and connectivity tiers are formed by running *Algorithm 1* and *2*. After each global update, the sink informs sensors of their type by using a control message. The effects of the length of rounds on network lifetime and energy consumption is discussed in Section 6.

4.4 Walk through the algorithms by an example

In this section, we present an example to explain the proposed two-tiered scheduling. Let $S = \{s_1, s_2, \ldots, s_{20}\}$ scattered through a sensing field randomly, having sensing range r^s and transmission range r^t as shown in Figure 3(a).

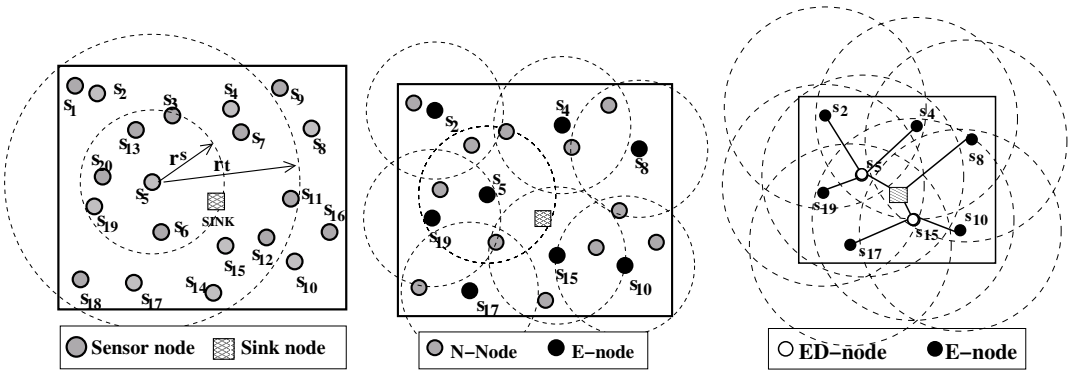
1 *Algorithm 1* is performed after receiving the neighbouring information and energy levels from sensors. Sink (gateway) node starts selecting the coverage set based on energy levels and location information as shown in Figure 3(b). In this example, the sink first selects s_5, then s_{10} followed by s_8 based on the coverage-benefit function given in (4). Sensing ranges of E-nodes are shown by dashed circles, where the total sensing area of selected nodes, can cover the entire field.

2 Next, the sink schedules N-nodes to sleep, then N-nodes turn their radio off until the next round. At the end of this step, the coverage-tier has been built. In this example, coverage set **C** is composed of sensors $\mathbf{C} = \{s_2, s_4, s_5, s_8, s_{10}, s_{15}, s_{17}, s_{19}\}$, which are represented by solid dots in Figure 3(b).

3 After constructing the coverage set, the neighbourhood information is determined using the coverage set selected in the previous step. In the first iteration, the sink has the information given in Table 3.

4 The sink starts selecting the node having minimum number of neighbours in the current CDS, denoted by **T** in the pseudo code in *Algorithm 2*, Initially **T** is equal

Figure 3 Walk through the centralised algorithms by an example: (a) initial state, (b) coverage-tier and (c) connectivity-tier



(a) (b) (c)

to **C**. For the example, sink starts with sensor s_2 having 3 neighbours. When we remove s_2 from **T**, $\mathbf{T} - s_2$ is still connected. Therefore, after setting its neighbour with maximum benefit as dominating node, we can remove s_2 from CDS. From s_4, s_5 and s_{19}, s_5 has the maximum benefit (assuming all sensors have equal residual energy). Then we add s_5 to **D**, and the *Algorithm 2* jumps to next iteration.

5 When, *Algorithm 2* terminates, we have the dominating set $\mathbf{D} = \{s_5, s_{15}\}$ as shown in Figure 3(c). Then sink immediately sends schedule nodes in **D** and $\mathbf{C} - \mathbf{D}$ to be active sleep and semi-active, respectively.

6 Sink re-performs this operation in every T_{CU} using up-to-date energy levels sent by the sensors.

Table 3 An example: e-node information in the sink

Sensor	*One-hop neighbours*	*Node degree*
2	4, 5, 19	3
4	0, 2, 5, 8, 15	5
5	0, 2, 4, 15, 17, 19	6
8	0, 4, 10, 15	4
10	0, 8, 15	3
15	0, 4, 5, 8, 10, 17, 19	7
17	0, 5, 15, 19	4
19	2, 5, 15, 17	4

5 Distributed two-tiered scheduling

We described a two-tiered scheduling mechanism with centralised control at the sink node in previous sections because it can provide algorithms for closer-to-optimal coverage set determination. Choosing the sink node as the target of data propagation is reasonable if we consider that the sink node has ample energy and computing power compared to individual sensor nodes. However, for some sensor network applications, sensor nodes may be deployed incrementally to highly dynamic and hostile environments thus, scheduling them without centralised control becomes particularly important. Therefore, we extend our work and discuss the distributed implementations of the *Algorithm 1* and *Algorithm 2* in this section.

The major challenges in distributed implementation are:

1 we need additional messages so that sensor nodes can exchange their location information and find their neighbours

2 sensors should calculate their coverage area and should determine whether their sensing regions are overlapping with neighbours

3 algorithms should be well modified such that, by use of local information, nodes can make decisions to become E-nodes or ED-nodes in an efficient way.

Next, we propose our solutions to these challenges.

5.1 Additional messages

In distributed implementation, additional messages are needed to before the coverage and connectivity sets are established. By this messaging phase, sensors in the network can achieve following goals:

- *Localisation:* first, each sensor determines its location, which can be done through one of lightweight localisation techniques designed for wireless sensor networks (Cheng et al., 2004).
- *Neighbour discovery:* next, each sensor must communicate with its neighbours using broadcast advertisements. In order to collect the one-hop neighbourhood information, all sensors broadcast their unique sensor IDs and their location coordinates. We assume stationary sensors having identical sensing ranges are located in a sensing field with no obstacles.
- *Overlapping coverage:* third, each sensor identifies its neighbours having overlapping coverage area based on the information received from neighbours. These neighbouring sensors are referred to as *coverage neighbours*. In other words, each sensor will select a set of neighbours with the common sensing area with itself. This is a necessary step in order to form a coverage-tier.

Consider that sensor s_i is located in coordinate (x_i, y_i) and receives an advertisement from sensor s_j which is located in (x_j, y_j). Let us denote the distance between s_i and s_j by $d(i, j)$, that is, $d(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. If $d(i, j) < 2r^s$, then sensor s_i records the sensor s_j as its *coverage neighbour*, which means it shares a sensing area in common. Let $\mathbf{N}_i^C$ be the coverage neighbours, then, $s_j \in \mathbf{N}_i^C$, where $\mathbf{N}_i^C \subseteq \mathbf{N}_i$.

This process ensures that every sensor node knows its its neighbours and their locations. Upon receiving location information of neighbouring nodes, a sensor can determine whether its sensing region can be fully covered by the sensors in their coverage neighbourhood. Therefore, it will decide to be an E-node or N-node for the coverage-tier. We first explain the approximate coverage calculation model, then move on to the distributed algorithms.

5.2 Approximate coverage calculation

One of the main challenges in distributed implementation is to identify the coverage of each sensor and determine whether the sensing region can be fully monitored by the neighbours. Note that, in centralised TTS, coverage-tier is determined by the sink, where the sink has the location and coverage information of all nodes in the network. Thus, before explaining the distributed algorithm, we describe how a sensor determines whether its coverage neighbours can fully monitor its sensing region. This is necessary to make a decision of being an essential or non-essential node.

To this end, we use the term of *coverage calculation* to calculate whether a specific sensing region of a sensor is covered. In fact, there is very limited work on the *distributed* coverage calculation in existing studies since it is a complicated geometric problem in finding the necessary *and* sufficient conditions. In this work we describe an

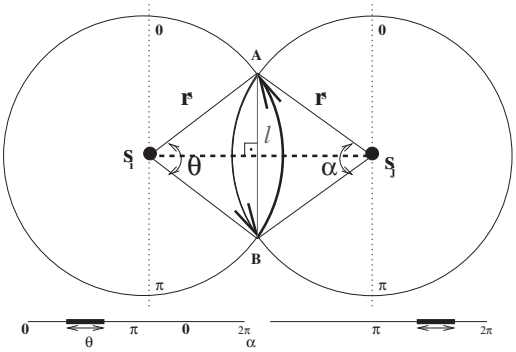
approximate coverage calculation which is composed of three conditions. The first condition, called *perimeter-test*, checks whether there are enough coverage neighbours such that all points in the perimeter should be within a sensing range of a coverage neighbour. This is a necessary condition based on the assumption of densely deployed nodes (Huang and Tseng, 2003). The second condition is called *center-test* in which it is examined whether the center of a sensor's coverage can be covered by at least one of its neighbours. The third condition is called *distance test*, those coverage neighbours must be close enough to the sensor, so that there may not be uncovered area inside the sensing region. Here, we explain these three conditions in detail as follows:

5.2.1 Perimeter-test

A sensor first determines whether the perimeter of its sensing region is covered. For example, two sensors s_i and s_j are shown in Figure 4 where their sensing ranges are illustrated as circular disks. In Figure 4, the distance between s_i and s_j is less than $2r^s$ (assume all sensing ranges are identical), thus having common sensing area. In this example, sensing range of s_i intersects with the range of s_j at points A and B. We observe that arc [AB] is inside of the sensing region of s_i whereas arc [BA] is inside of the sensing region of s_j. In this context, we simply say arc [AB] is covered by s_i and [BA] is covered by s_j. A sensor calculates the arc which is covered by a neighbour using the angles α or θ in Figure 4 which are:

$$\alpha = \theta = \arccos\left\{\frac{d(i,j)^2}{2(r^s)^2} - 1\right\} \tag{6}$$

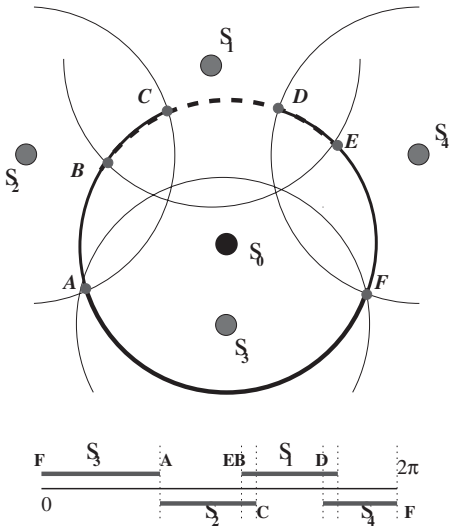
Figure 4 Sensor s_i and sensor s_j share an area in common, where $d(i, j) < 2r^s$



By examining each coverage neighbour, a sensor can determine if intersected arcs in total are sufficient to enclose its perimeter from 0 to 2π. If the perimeter is enclosed, we refer that "perimeter-test" is passed. To do this, we simply determine the intersection points of the arcs and scan the perimeter as illustrated in Figure 5. Note that, if we assume that no two sensors are located in the same location, then we need at least 3 coverage neighbours to satisfy the perimeter-test.

We show an example of the perimeter-test in Figure 5, where sensor s_0 has 4 coverage neighbours. Using their location coordinates, sensor s_0 can find the intersection points of the arcs (Arganbright, 1993). Let the line segment from 0 to 2π in 5 denote the perimeter of s_0. After we list the arc as shown in Figure 5, we scan the perimeter in Figure 5. We can see that the entire perimeter is enclosed by [AF] (node s_3), [FD] (node s_4), [EB] (node s_1) and [CA] (node s_2).

Figure 5 Sensor s_0 having 4 coverage neighbours s_1, s_2, s_3 and s_4. Its perimeter is fully enclosed



Although, perimeter-test ensures that a sensor has necessary number of coverage neighbours, it is not sufficient to satisfy the full coverage of the sensing region. There may have some uncovered area in the middle of the region. Thus, we propose a *center-test* and a *distance-test*, so that a sensor ensures that neighbours are close enough to the center and provide full coverage.

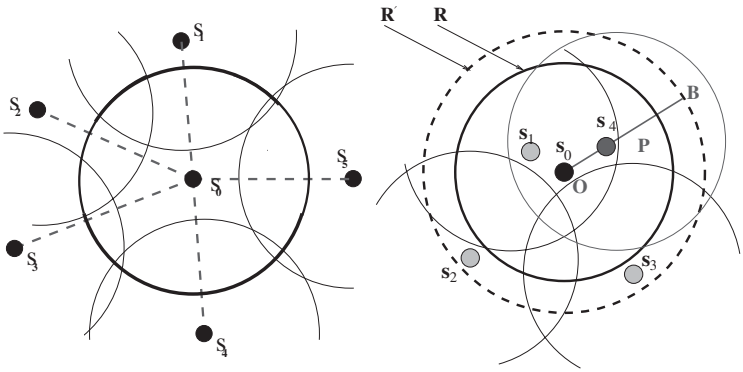
5.2.2 Center-test

When a sensor passes the perimeter-test, it has necessary number of coverage neighbours. However, this is not sufficient to claim that the sensing region is fully covered. For instance, there are two examples in Figure 6 (a) and (b), where perimeter-test has been passed. In Figure 6 (a), neighbours of s_0 are successfully covers the sensing region. However, in Figure 6 (a), even there are 5 coverage neighbours, there is an uncovered area in the middle of the region. Therefore, the motivation of the center-test is to ensure that the center of a sensing region can be covered by at least one of a node's neighbours.

In this step, sensor s_i chooses one of its coverage neighbour s_j as *primary* neighbour which satisfies $d(i, j) \leq r^s$, where $d(i, j)$ is the distance between s_i and s_j. One intuitive necessary condition is that there should be at least one primary neighbour to cover the center point of a sensing region. If there is no primary neighbour as in Figure 6(a), that is, distances between all coverage neighbours and the s_0 are greater than r^s, then neighbours are not sufficient to achieve full coverage. Therefore, center-test is a necessary condition in finding a coverage set.

In Figure 6(b), the *primary* neighbour of s_0 is s_4, where $d(0, 4) = |OP| \leq r^s$. In case there are multiple sensors satisfying $d(i, j) \leq r^s$, we select the one having the minimum distance as the primary neighbour.

Figure 6 Two examples where perimeter-test is passed for sensor s_0: (a) distance test: fail and (b) distance test: pass



(a) (b)

5.2.3 Distance-test

When a sensor passes the perimeter-test, and center-test, it is still possible that there are uncovered area of a sensor's coverage. The motivation of *distance-test* is to verify that coverage neighbours are close enough to the center and satisfy full coverage, which is based on the selection of primary neighbour in the center-test.

Let $s_p(i)$ be the primary neighbour of s_i. In this test, we check if for all $s_j \in \mathbf{N}_i^C$, the following condition satisfies:

$$d(i, j) = r^s + d(i, p) \tag{7}$$

In Figure 6 (b), to illustrate the distances between s_0 and neighbours clearly, we draw an *extended coverage* range of node s_0, where the center is s_0 and the radius is $R' = |OB| = r^s + |OP|$. We called the original sensing range of s_0 as R and the extended range as $R^{'}$ in the Figure 6 (b). The condition in (7) can be verified by two extreme cases: for $d(i, p) = 0$, that is, node s_p and node s_i are in the same location, then the radius of the extended coverage, $R' = r^s$, which is exactly the same as s_i, that is, there is no extension of coverage from node s_p; for $d(i, p) = r$, that is, node s_p is on the perimeter of node s_i, then $R' = r^s + r^s = 2r^s$, which means that the extended coverage is enlarged one time. Therefore, the extended coverage shows the maximum distance that the primary neighbour can reach. If all sensors in the coverage neighbourhood of s_i are closer than $r^s + d(i, p)$, then we say that full coverage is achieved and distance-test is passed.

Therefore, the first condition, *perimeter-test* is a necessary condition to cover the perimeter; and the second condition, *center-test* is also a necessary condition to cover the center. The third condition, *distance-test* is very effective for the full coverage after many tests, though it is an approximate condition for coverage calculation. Therefore, in our approach, sensors that have passed the perimeter-, center-, and and distance- tests, may have gone into sleep mode, which will be discussed in the next section.

5.3 Distributed algorithms

Using broadcast messages, each sensor in the network is aware of its location, neighbours, and the overlapping coverage. Our objective is to establish the coverage and connectivity tiers in a distributed fashion using this local information. Like the centralised approach, distributed algorithms for two-tiered scheduling include a sequence of steps. First step is to check whether the sensing region of a sensor is covered by coverage neighbours. If so, sensors calculate their benefits of being E-nodes and then start sending announcement messages to build the coverage set. After coverage set is established, connected dominating set is dynamically constructed starting from the sink. Here, we give the details of the steps that each sensor follows to build two-tiered scheduling architecture.

5.3.1 Mandatory E-node test

As we mentioned, the first step is to decide whether the sensing region of a sensor is fully covered by the neighbours using the approximate coverage calculation model. If sensor realises that the neighbours are not sufficient to provide fully coverage, then it is a *mandatory* member of the coverage set. We denote such E-nodes as *mandatory* E-nodes. In this case, a sensor broadcasts an I-AM-ESSENTIAL message and becomes an E-node. Otherwise, it follows the *benefit calculation* step. Nodes, which are not mandatory, may go into sleep mode based upon their benefits.

5.3.2 Benefit calculation

Any sensor that is not a mandatory E-node calculates its benefit similar to the coverage-benefit function given in (4). However, in distributed implementation, the *uncovered sensing area* represents the sensing region of the sensor s_i that is not covered by mandatory nodes. Hence, we replace the denominator of (4), that is, $(R_i \cup A)/R_C$ with $R_i/R_{\mathbf{N}_i^{\mathrm{CM}}}$, where $\mathbf{N}_i^{\mathrm{CM}}$ is the set of mandatory essential neighbours of sensor s_i. Set $\mathbf{N}_i^{CM}$ is formed upon receiving I-AM-ESSENTIAL messages. Then, each node calculates its benefit as follows:

$$\text{Benefit}^{(\mathrm{C})}(s_i) = w\left(i, \frac{R_i}{R_{\mathbf{N}_i^{\mathrm{CM}}}}\right), \tag{8}$$

where R_i is the sensing region of sensor s_i and $R_{\mathbf{N}_i^{CM}}$ is the total region covered by the mandatory essential neighbours of sensor s_i.

One challenging issue in constructing the coverage set is how to select E-nodes such that, nodes having higher benefits may have a better chance of being essential. Further, if there are enough E-node neighbours of sensor s_i, such that, they can cover its sensing region, then s_i should go into sleep mode. As a solution for these issues, we use a 'max-benefit announce first' strategy which will be explained next.

5.3.3 Max-benefit announce first

In this step, mandatory E-nodes have already announced and each node is aware of its benefit. Then, nodes start to broadcast I-AM-ESSENTIAL messages after a back-off time between $[0, CB_{\text{MAX}}]$, where CB_{MAX} denotes the maximum back-off while establishing coverage-tier and is determined based on average one-hop latency. Sensors determine their back-off time based on their benefit, that is, a node having the maximum benefit will have the shortest back-off time, thus announcing the I-AM-ESSENTIAL message earlier.

One important point is to keep the benefit updated whenever an I-AM-ESSENTIAL message is received from a coverage neighbour. New benefit should be calculated based upon the up-to-date *uncovered area*, because some of its region will be covered by newly announced neighbours. Thus, the benefit will decrease proportional to the uncovered area which also effects the back-off time. If a sensor has not sent an I-AM-ESSENTIAL message by CB_{MAX}, it may go to sleep, which means its sensing region is fully covered by its coverage neighbours. Coverage tier is established at the end of CB_{MAX}. When a coverage tier is set up, a set of sensors are selected as E-nodes which can monitor the entire field with maximum benefit. Next, we will discuss how the connectivity tier is established.

5.3.4 Dominating set construction

Following, we need to construct the connected dominating set among the nodes in the coverage set. We use a greedy approach similar to the centralised algorithm, that is, sensors are removed one by one as long as the remaining set is connected. However, when such a greedy approach is run by sensors, a distributed algorithm, for example, distributed breadth-first search, is necessary to ensure that the remaining network is connected in each iteration. In a large-scale sensor network, distributed breadth-first search may incur high overhead due to its computational complexity, that is, $O(Dlog^3N)$, where D is the diameter of the network (Butenko et al., 2004). Therefore, we propose a dynamic dominating set construction approach triggered by the sink.

During dominating set construction, sensors broadcast three types of messages: FIND-DOMINATOR, DOMINATOR-CANDIDATE and SELF-REMOVAL. Receiving a FIND-DOMINATOR message indicates that there is no dominator in the neighbourhood, so a sensor node may become a dominating node. In response, this sensor broadcasts a DOMINATOR-CANDIDATE message after receiving FIND-DOMINATOR. However, a sensor do not forward the FIND-DOMINATOR to all its neighbours immediately. Instead, it takes a back-off time to ensure that it should be a dominating node to preserve the connectivity of the network. After the back-off time, if the sensor forwards the FIND-DOMINATOR message to its neighbours, it becomes an ED-node. SELF-REMOVAL message is sent by sensors which are already connected to a dominating node and they decide not to be ED-nodes.

Our dominating set construction starts from the sink by sending a broadcast FIND-DOMINATOR message.The idea is that sensors which will forward the message, are included to the connectivity set as a dominator. In the first step, FIND-DOMINATOR message is received by the neighbours of the sink, which are called *candidate dominators*. A candidate dominator, again sets a back-off time $\in [0, DB_{\text{MAX}}]$ to forward the message, where DB_{MAX} denotes the maximum back-off while establishing connectivity-tier and is determined based on average one-hop latency, and the node density. Back-off time will be inversely proportional to the benefit that uses the degree of connectivity and residual energy given in (5). In distributed implementation, degree of connectivity is the number of neighbours which have not sent a FIND-DOMINATOR or SELF-REMOVAL message. For example, sensor s_i has 4 neighbours among which two of them have sent a SELF-REMOVAL message, whereas the other neighbour has sent a FIND-DOMINATOR message. This implies that two neighbours are connected to ED-nodes and one neighbour has already become an ED-node. In this case, degree of connectivity of s_i is 1 in calculating its benefit. Similar to the previous step, a node having greater benefit has shorter back-off time, thus forwarding FIND-DOMINATOR message earlier to be an ED-node.

When a node receives a FIND-DOMINATOR message, it

1 updates its benefit such that its degree of connectivity is decreased; because one of its neighbours becomes a dominator

2 sets/updates its back-off time based on newly calculated benefit and

3 broadcasts a DOMINATOR-CANDIDATE message.

By receiving DOMINATOR-CANDIDATE messages during the back-off time, candidate dominators can be noticed if their neighbours are also candidates. If all neighbours of a candidate node is either dominator or candidate dominators, it can safely give up of being dominator, since all its neighbours have already received a DOMINATOR-CANDIDATE message. In this case, a sensor node sends a SELF-REMOVAL message indicating that it will not be an element of CDS.

At the end of a back-off period, a candidate which has not been self-removed, forwards FIND-DOMINATOR message and becomes an ED-node. Following the forwarded FIND-DOMINATOR message, new candidates appear and send DOMINATOR-CANDIDATE messages. This process continues until all nodes have received at least one FIND-DOMINATOR message. Note that, a sensor updates its benefit after receiving FIND-DOMINATOR or SELF-REMOVAL messages.

Next, we show the execution of the algorithm by an example.

5.4 Walk through the alternative distributed TTS by an example

In this section, we use the same example in Section 4.4 to explain the distributed implementation of two-tiered scheduling. We have the following steps while establishing coverage and connectivity tiers:

- First step is finding the mandatory E-nodes with the help of perimeter-test, center-test, and distance-test given in Section 5.2. Each node determines if its sensing region can be fully covered by its neighbours. In the example in Figure 7 (a), nodes s_8 and s_{10} are mandatory nodes. All other nodes have passed perimeter, center, and distance tests, that means, any of them can go into sleep mode based on the benefit calculation. This is quite different from the TTS centralised implementation. In Figure 3 (a) for centralised TTS, the sink selects s_5, so the coverage of s_5 is shown only.

- Next, they calculate their coverage benefits. Assume that each sensor has equal residual energy, that is, the network is just deployed. In this case, for example sensor s_5 will have the maximum benefit, since it covers the largest uncovered area. On the other hand, sensor s_{11} has almost 0 benefit, because its sensing region inside the rectangle is already covered by mandatory E-nodes s_8 and s_{10} In this case, sensor s_5 has the shortest back-off time and sends I-AM-ESSENTIAL message first. After getting this massage, its coverage neighbours update their benefits. This process will continue by the end of CB_{MAX}, where each node sends either I-AM-ESSENTIAL message or is fully covered by its neighbours. At the end of the CB_{MAX}, coverage set is established.

- Establishment of dominating set starts from the sink. As illustrated in Figure 7 (b), the sink broadcasts a FIND-DOMINATOR message which is received by its neighbouring E-nodes s_4, s_5, s_8, s_{10}, s_{15}, s_{17} and s_{19}. Those E-node neighbours then broadcast DOMINATOR-CANDIDATE messages after receiving the sink's message and calculate their benefits. Among those neighbours, s_{15} has the highest degree of connectivity as given in Table 3, thus it has the shortest back-off time. Hence, s_{15} will forward FIND-DOMINATOR message and become an ED-node first. After receiving message from s_{15}, only sensor s_5 determines that its neighbour s_2 is neither a dominator nor a candidate dominator (not receiving any FIND-DOMINATOR message). Therefore, sensor s_{15} also forwards FIND-DOMINATOR message and becomes a dominator. Figure 7(c) shows the coverage and connectivity tiers at the end of the execution of distributed algorithms.

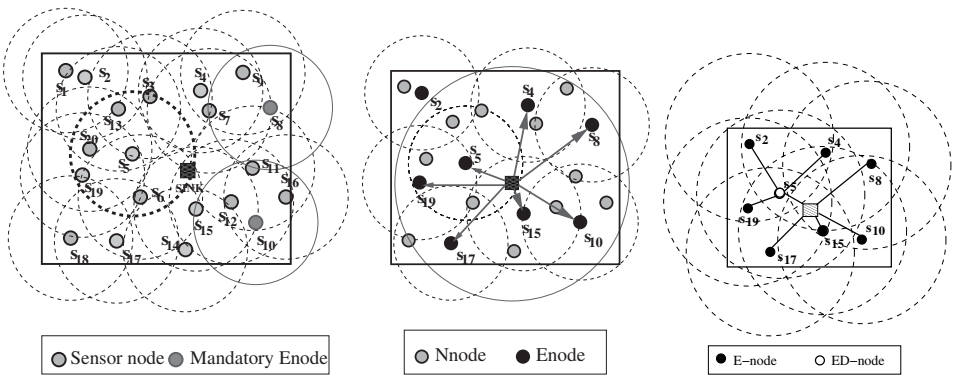
5.5 Comparison of alternative distributed TTS

In this section, we compare the distributed TTS, centralised TTS and CCP (Wang et al., 2003) in terms of on-duty node ratio in Figure 8 because it is an immediate measure to evaluate the effectiveness of scheduling mechanisms. The lower the on-duty node ratio, the more nodes can be put into sleep, thus increasing the network lifetime. Here, *on-duty nodes* refer to the nodes which are not scheduled to sleep. We show that the number of on-duty nodes in centralised TTS and distributed TTS are lower than the number of on-duty nodes in CCP, that is, scheduling scheme that integrates coverage and connectivity. In Figure 8, ratio of ED-nodes and E-nodes are illustrated on a single column where ED-node ratio is on the bottom with having different patterns.

In CCP, results for 1-degree coverage show that 20% of 100 nodes should be active which have been randomly deployed on a 50 m × 50 m area with sensing range of 10 m. However, in centralised TTS, on average 1% of nodes is always active and 23% is periodically active/sleep in the same dense network. Similarly, distributed TTS performs better than CCP. Eventhough, the ratio of E-nodes is higher than the ratio of on-duty nodes in CCP, E-nodes are not active all the time, instead having semi-sleep behaviour.

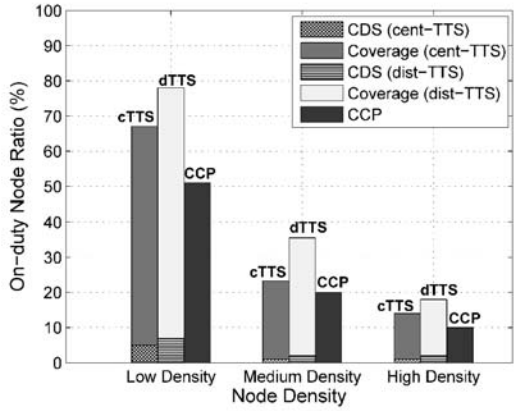
In Figure 8, we can observe that the ratio of E-nodes and ED-nodes decreases in both centralised TTS and distributed TTS as the node density increases, which shows that the proposed algorithms can establish a near optimal coverage set and CDS regardless of node density.

Figure 7 Walk through the distributed TTS by an example: (a) mandatory E-nodes, (b) coverage tier and (c) connectivity tier



(a) (b) (c)

Figure 8 Percentage of on-duty nodes under different node densities



On the other hand, when we compare the centralised TTS and distributed TTS, we observe that centralised TTS performs 10% better at most in terms of on-duty node ratio. However, the cost for deploying the centralised algorithm in sensor networks may be different from that for distributed approach, which may vary greatly between applications. In essence, we extend the two-tiered scheduling mechanism to distributed implementation.

6 Performance evaluation

We present the performance of the two-tiered scheduling by simulations with three sets of experiments. In the first set of experiments, we evaluate the number of *on-duty nodes* which are *E-nodes* and *ED-nodes* in two-tiered scheduling. We compare our results with the schemes that integrates coverage and connectivity, referred as CCP (Wang et al., 2003), and connected sensor cover (Gupta et al., 2003). Second, we measure energy consumption and residual energy distribution among nodes. We also compare the effect of two-tiered scheduling in prolonging the network lifetime. Finally, the impact of the proposed scheme is investigated on end-to-end delay and packet loss.

6.1 Simulation environment

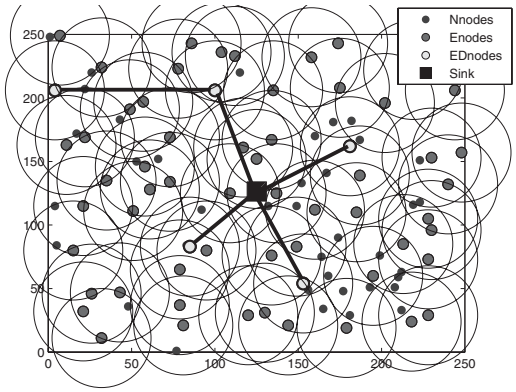
The performance of our tiered approach is evaluated using ns2 simulator (Ns-2, 2004). Simulations are performed in an 250 m × 250 m area consisting of different numbers of sensors distributed randomly. In the basic scenario, 100 fixed sensor nodes having transmission range of 100 m and sensing ranges of 25 m are used.

We use the energy model given in Section 3.2 and radio power consumption parameters in Table 1. The energy consumption of turning the radio on/off is negligible. The buffer size of sensor nodes is chosen as 50 and the packet length is 100 bytes. Our protocols run on the 802.11 MAC with power saving support.

In our experiments, we use a mobile tracking application in which the movements of a mobile node are reported to the sink in every sensing period. Movements of the mobile (phenomenon) node are generated with *random waypoint* model. *Event-driven* data delivery model is used from sensors to the sink, where sensors send an event report if the phenomenon is detected in their sensing region. Sink node is located at the center of the sensing field. Event reports are sent to the sink in every sensing period, which is 0.5 sec during the simulation. On the other hand, the sink sends periodic queries to the sensors in every 2 sec.

All experimental results are provided after averaging 10 random topologies. A random network topology of a simulation experiment with 100 nodes is shown in Figure 9. The sensing range of E-nodes is represented by circular disks which is 25 m. We see the coverage and connectivity tiers in Figure 9. There are 5 dominating nodes among 100 nodes which stay active all the time on their coverage set. As illustrated in the figure, E-nodes are sufficient to monitor the entire 250 m × 250 m sensing field.

Figure 9 Network topology of 100 nodes in a typical run



6.2 Percentage of on-duty nodes

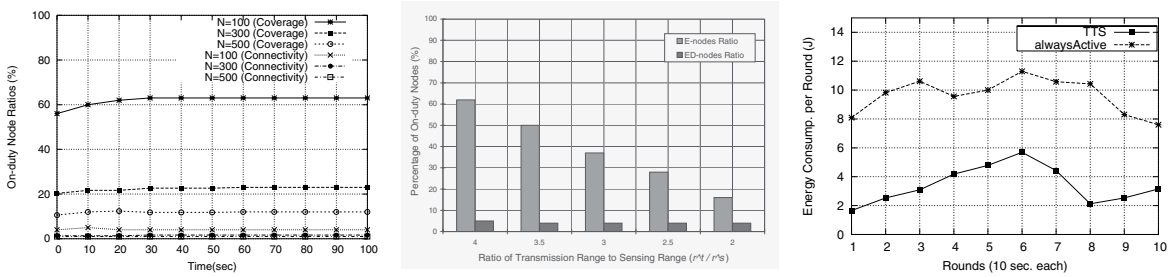
In this paper, one of our goals is to reduce the number of on-duty nodes, E-nodes and ED-nodes. For this purpose, we first measure the number of E-nodes and ED-nodes for each round. In the basic scenario, we take the ratio of transmission range over sensing range (r^t/r^s) as 4, that holds for most commercially available sensor nodes (Zhang and Hou, 2004). Figure 10 shows the percentage of on-duty nodes of three networks of different node densities. To cover a 250 m × 250 m area with sensing range of 25 m, we use random placement of 100, 300 and 500 nodes. According to (Slijepcevic et al., 2001; Williams, 1979), the minimum number of nodes needed for full coverage of an area **A** can be calculated by:

$$\frac{N(r^s)^2\pi}{\mathbf{A}} = \frac{2\pi}{\sqrt{27}} \tag{9}$$

Therefore, to cover a 250 m × 250 m area with sensing range of 25 m, we classify random placement of 100 nodes as *low* density (1600 sensors/km^2), 300 as *medium* (4800 sensors/km^2), and 500 as *high* density (8000 sensors/km^2).

From Figure 10(a), we can observe that among these three scenarios, the network having 500 nodes has the lowest ratio of E-nodes and ED-nodes, thus showing that the greedy algorithms perform even better in densely deployed network.

Figure 10 Performance of two-tiered scheduling mechanism: (a) percentage of on-duty nodes versus time, (b) percentage of on-duty nodes versus the ratio of r^t to r^s and (c) energy consumption per round



(a) (b) (c)

In low density scenario, we observe that the E-node ratio is at most 63%, whereas the ED-node ratio is around 5%. This indicates that only 5 nodes among 100 is active over time, which may dramatically effect the network lifetime. Also results indicate that the ratio of on-duty nodes remains stable in time. In the simulation, T_{NO} is 10 sec. Therefore, the sink re-generates the coverage-tier and connectivity-tier using new energy values every 10 sec to balance the energy consumption.

Moreover, we also evaluate the number of on-duty nodes of higher sensing ranges. When the ratio of (r^t/r^s) is getting smaller, the number of E-node decreases dramatically up to 18%, while ED-node ratio remains the same. Figure 10(b) shows the ratios of on-duty nodes in a low dense network which decrease monotonically as the ratio of r^t and r^s decreases. In other words, the number of on-duty nodes in a sensor network depends heavily on the ratio of transmission range and sensing range of each sensor.

6.3 Energy consumption and network lifetime

Here, we evaluate the energy consumption of the proposed two-tiered scheduling approach, comparing with *alwaysActive* scheme, where nodes are not scheduled to sleep. The reason is that there is not much performance evaluation in the prior works integrating coverage and connectivity (Gupta et al., 2003; Wang et al., 2003) in terms of energy conservation. Instead, they have evaluated the coverage percentage, achieved coverage degree and size of the coverage sets. In our experiments, we show that two-tiered scheduling provides a considerable energy consumption compared to *alwaysActive* scheme while fully monitoring the sensing field.

In Figure 10(c), we show the average residual energy dissipation per round. By two-tiered scheduling, labelled as *TTS*, the average energy dissipation is reduced around 40% compared to *alwaysActive* scenario. Energy dissipation values, depicted in this figure, are the average of the corresponding round. For example in the 3rd, the average energy consumption is around 3 J in two-tiered scheduling, while it is 6 J in *alwaysActive* scenario. The reason for this performance lies on the tiered approach in which on-duty nodes can be put into semi-sleep state for the coverage, which results in more energy savings.

Further, we show the effect of the two-tiered scheduling on the network lifetime. The key factors prolonging the network lifetime are

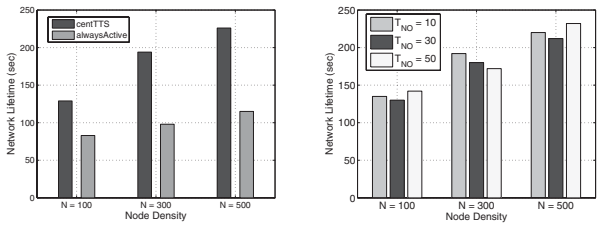
(i) the energy conservation, which can be achieved by putting more sensors into sleep mode and

(ii) balanced energy consumption among sensors.

A non-uniform residual energy consumption within the entire sensor network may lead to network partitioning and shorten the network lifetime. Severity of such non-uniform energy consumption is alleviated by updating the coverage and connectivity tiers.

First, we demonstrate the network lifetime in Figure 11(a) compared to *alwaysActive* scheme with the initial energy of 1 J. We consider a WSN as *alive* when the sensing field is fully covered. In other words, a network is alive when every point in **A** is covered by at least one sensor. According to this, we observe that network lifetime is prolonged significantly in two-tiered scheme compared to *alwaysActive*, especially in high density networks. Even in low density network with 100 nodes, network lifetime is prolonged up to 28% which shows the effective energy savings of proposed tiered approach. We estimate a similar effect on lifetime for distributed implementation of TTS, since we observe very close on-duty node ratio compared to the centralised TTS.

Figure 11 Network lifetime: (a) lifetime versus node density and (b) lifetime versus network operation interval T_{NO}



(a) (b)

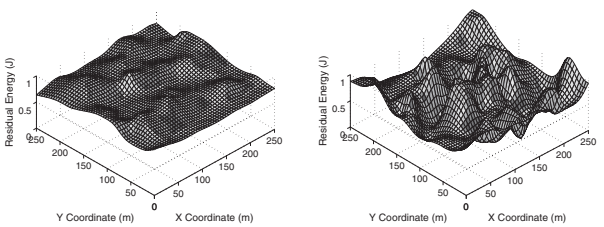
It is worthy of mention that in CCP, network lifetime is measured in terms of coverage percentage. CCP with connectivity feature keeps the coverage above 90% (lifetime threshold) until 470 sec with node density 200. However,

the sensors are deployed with a much higher initial energy selected randomly within the range from 200 J to 300 J. Also, the power consumption of Tx, Rx, Idle and Sleeping in CCP are for 802.11 network card, which is not common for a sensor node (Hill and Culler, 2002). Compared to CCP, network lifetime can be prolonged by TTS up to 132 sec in a similar dense network with the given Mica2 power consumptions and initial energy of 1 J. This signifies that, *two-tiered scheduling* approach with *semi-sleep* behaviour of coverage-tier nodes has great impact on prolonging the network lifetime.

Meanwhile, we notice that energy savings resulted from the proposed two-tiered scheduling is accompanied by the overhead of updating the coverage and connectivity sets. At the beginning of each round, the sink re-schedules the sensors. Without updating, although we conserve energy, nodes in active state (ED-nodes) will die faster, which causes shorter network lifetime due to the waste of unused energy of sleeping nodes. For this purpose, in Figure 11(b), we plot network lifetime against network operation interval, T_{NO}, for various dense networks. When T_{NO} is longer, the sink updates the coverage-tier and connectivity-tier less frequently, thus conserving energy. However, this does not result in prolonging the network lifetime. We observe that T_{NO} changes the network lifetime at most 5% as a result of energy consumption and residual energy distribution. This shows that even with the overhead of updating, the network lifetime can be expanded significantly by using two-tiered scheduling mechanism.

As a matter of fact, updating E-nodes, which provides balanced energy distribution among sensors, is also effective in prolonging lifetime. To show the energy distribution, we depict the residual energy of sensors in Figure 12 where x-y plane represents the sensing field. Nodes are positioned in their actual locations as in the simulation and z-axis represents their residual energy. In Figure 12(a), coverage-tier and connectivity-tier are updated in every round based on their new residual energy levels. However, in Figure 12(b) coverage-tier and connectivity-tier are established at the beginning without updating or rotation over time. Note that the surface in Figure 12(a) does not fluctuate dramatically as in Figure 12(b), indicating that the residual energy of sensors in Figure 12(a) is close to each other. Therefore, by rotating nodes, the balance or the fair energy consumption of sensors is achieved and the lifetime of a sensor network is extended.

Figure 12 Residual energy distribution: (a) two-tiered scheduling with updating and (b) two-tiered scheduling without updating



(a) (b)

6.4 End-to-end delay and average packet loss

Figures 13(a) and (b) show the performance of TTS with respect to average end-to-end delay and packet loss ratio. We have simulated two types of traffic load scenarios for TTS as:

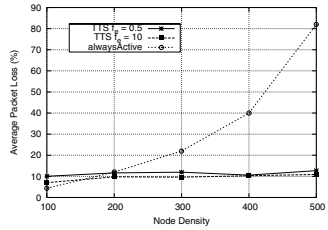
1 TTS $f_e = 0.5$, where sensors send event messages on every 0.5 sec if the event is detected and

2 TTS $f_e = 5$, where sensors send event messages on every 5 sec.

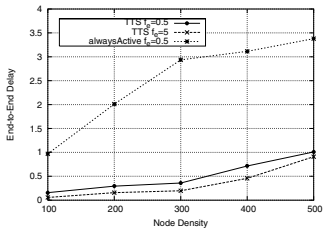
Figure 13(a) shows the packet loss ratios of two TTS traffic loads compared to *alwaysActive* scheme. When node density increases, number of packets generated in the network also increases. In a dense network, there will be more nodes detecting the same event and sending this event packet to the sink. For this reason, in *alwaysActive* scheme, packet loss ratio dramatically increases as the number of nodes in the network increases. However, in TTS, even for different traffic loads, packet loss ratio remains almost constant. The reason is that, using scheduling algorithms, only the on-duty nodes are active, whereas others are scheduled to sleep. As the node density increases, since the percentage of on-duty nodes decreases, density of the on-duty nodes are always same, thus we observe a constant packet loss ratio.

Further, in Figure 13(b), we observe that the end-to-end delay in TTS is significantly smaller than in *alwaysActive* scheme. This is again the effect of scheduling of large number of nodes. As a result, we show that two-tiered scheduling not only provides an efficient energy conservation but also is a scalable architecture for wireless senor networks.

Figure 13 Network performance: (a) Packet loss ratio and (b) End-to-end delay



(a)



(b)

7 Conclusion

In this paper, we presented a two-tiered approach addressing the efficient scheduling issue in wireless sensor networks. In order to prolong network lifetime, we schedule sensors

to be in power saving mode, while preserving coverage and connectivity. We decomposed the coverage and connectivity functionalities of a sensor network into two-tiers; thus, nodes having been used for connectivity or coverage have different sleeping behaviours. We first established the coverage-tier based on their sensing areas of sensors by a weighted greedy algorithm. Nodes in the coverage-tier can monitor the entire sensing field and periodically wake up to send and receive to/from the sink, whereas dominating nodes selected for the connectivity-tier stay active to forward the data traffic. Further, we discuss an alternative distributed implementation of the algorithms where sensors use local neighbouring information to establish the coverage and connectivity sets.

Simulation experiments have validated that a significant energy saving is achieved by the proposed scheduling algorithms while providing full coverage and connectivity. The centralised algorithm for finding the coverage set, within $ln(N)$ factor of optimal size, selected 30% of nodes on average in medium dense networks. Among the coverage set, on average 2% of nodes are always active as dominating nodes. Furthermore, we showed that energy consumption is balanced and the network lifetime is prolonged around 30% by rotation of coverage and connectivity tiers.

References

Alzoubi, K., Wan, P. and Frieder, O. (2002) 'Distributed heuristics for connected dominating set in wireless adhoc networks', *KICS Journal of Communications and Networks*, Vol. 4, No. 1.

Arganbright, D. (1993) *Practical Handbook of Spreadsheet Curves and Geometric Constructions/Book and Disk*, CRC Press.

Butenko, S., Cheng, X., Oliveira, C. and Pardalos, P. (2004) 'A new heuristic for the minimum connected dominating set problem on adhoc wireless networks', *Cooperative Control and Optimization, Kluwer Academic Publisher*, pp.61–73.

Cardei, M., Thai, M. and Wu, W. (2005) 'Energy-efficient target coverage in wireless sensor networks', *Proceedings of IEEE Infocom*, Miami, Forida, USA.

Cerpa, A. and Estrin, D. (2002) 'ASCENT: adaptive self-configuring sensor networks topologies', *Proceedings of IEEE Infocom*, New York: USA, Vol. 3, pp.1278–1287.

Chang, J-H. and Tassiulas, L. (2004) 'Maximum lifetime routing in wireless sensor networks', *IEEE/ACM Transactions on Networking*, August, Vol. 12, No. 4, pp.609–619.

Chen, B., Jamieson, K., Balakrishnan, H. and Morris, R. (2001) 'SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks', *Proceedings of ACM Mobicom*, pp.85–96, Italy: Rome.

Cheng, X., Thaeler, A., Xue, G. and Chen, D. (2004) 'TPS: a time-based positioning scheme for outdoor wireless sensor networks', *Proceedings of IEEE Infocom.*

Garey, M.R. and Johnson, D.S. (1990) *Computers and Intractability; A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., NY: New York, USA.

Gupta, H., Das, S.R. and Gu, Q. (2003) 'Connected sensor cover: self-organization of sensor networks for efficient query execution', *Proceedings of ACM Mobihoc*, Annapolis, Maryland, USA.

Heinzelman, W., Chandrakasan, A. and Balakrishnan, H. (2002) 'An application specific protocol architecture for wireless microsensor networks', *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, pp.660–670.

Hill, J. and Culler, D. (2002) 'Mica: a wireless platform for deeply embedded networks', *IEEE Micro 22*, Vol. 22, No. 6, pp.12–24.

Huang, M-F. and Tseng, Y-C. (2003) 'The coverage problem in a wireless sensor network', *Proceedings of ACM WSNA*, San Diego, CA: USA.

Ns-2: (2004) Available at: http://www.isi.edu/nsnam/ns.

Pan, J., Hou, T., Cai, L., Shi, Y. and Shen, S. (2003) 'Topology control for wireless sensor networks', *Proceedings of ACM Mobicom*, San Diego, California, USA.

Slavik, P. (1996) 'A tight analysis of the greedy algorithm for set cover', *ACM Symposium on Theory of Computing*.

Slijepcevic, S. and Potkonjak, M. (2001) 'Power efficient organization of wireless sensor networks', *Proceedings of IEEE International Conference on Communications*, pp.472–476, Helsinki.

Tian, D. and Georganas, N.D. (2002) 'A coverage-preserving node scheduling scheme for large wireless sensor networks', *Proceedings of First ACM Interactions Workshop on Wireless Sensor Networks and Applications*, Georgia, USA.

Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R. and Gill, C. 'Integrated coverage and connectivity configuration in wireless sensor networks', *Proceedings of SenSys*, pp.28–40, Los Angeles, CA, USA.

Williams, R. (1979) 'The geometrical foundation of natural structure: a source book of design', *Dover Pub. Inc*, pp.51–52, New York.

Xu, Y., Heidemann, J.S. and Estrin, D. (2001) 'Geography-informed energy conservation for ad hoc routing', *Proceedings of ACM Mobicom*, pp.70–84, Rome, Italy.

Ye, W., Heidemann, J. and Estrin, D. (2002) 'An energy-efficient MAC protocol for wireless sensor networks', *Proceedings of IEEE Infocom*, Vol. 3, pp.1567–1576, New York, USA.

Younis, O. and Fahmy, S. (2004) 'HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks', *IEEE Transactions on Mobile Computing*, Vol. 3, No. 4, pp.366–379.

Zhang, H. and Hou, J. (2004) 'Maintaining sensor coverage and connectivity in large sensor networks', *Proceedings of NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Adhoc Wireless, and Peer-to-Peer Networks*.

Zhao, Y., Govindan, R. and Estrin, D. (2002) 'Residual energy scan for monitoring sensor networks', *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'02)*.