Sensors **2009**, *9*, 3908-3941; doi:10.3390/s90503908

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Dynamic Hierarchical Sleep Scheduling for Wireless *Ad-Hoc* Sensor Networks

Chih-Yu Wen * **and Ying-Chih Chen**

Department of Electrical Engineering, Graduate Institute of Communication Engineering, National Chung Hsing University, Taichung 402, Taiwan; E-Mail: d9664106@mail.nchu.edu.tw

* Author to whom correspondence should be addressed; E-Mail: cwen@dragon.nchu.edu.tw

Received: 18 March 2009; in revised form: 29 April 2009/ Accepted: 20 May 2009/ Published: 25 May 2009

Abstract: This paper presents two scheduling management schemes for wireless sensor networks, which manage the sensors by utilizing the hierarchical network structure and allocate network resources efficiently. A local criterion is used to simultaneously establish the sensing coverage and connectivity such that dynamic cluster-based sleep scheduling can be achieved. The proposed schemes are simulated and analyzed to abstract the network behaviors in a number of settings. The experimental results show that the proposed algorithms provide efficient network power control and can achieve high scalability in wireless sensor networks.

Keywords: wireless sensor networks; scheduling management; sensing coverage

1. Introduction

Recent advances in microelectro-mechanical systems are driving the developments of low-cost and and low-power wireless sensors, with diverse applications in the physical world in areas such as environmental monitoring, disaster recovery, industrial process control, and smart environments. With sensors placed close to an event, wireless sensor networks can observe the phenomenon and receive data. However, having too few active sensors or excessive ones may result in reduced sensing coverage or severe interference, which will have a great influence on network performance features such as energy and bandwidth efficiency, and sensing quality. Therefore, sensing scheduling schemes may be implemented to tackle basic problems of sensor networks (e.g. energy constraints and communication interference) in order to reduce energy consumption and prolong network lifetime.

Sensor scheduling aims to maintain a balance of network resources. Recent research has found that significant energy savings can be achieved by dynamic power management in sensor networks [1-7]. To achieve this sensing process, sensors are scheduled to execute the sensing task. Hence, reducing the sensing redundancy and maintaining sufficient sensing coverage and network connectivity are critical requirements in sensor networks. In addition, the two issues of energy constraint and communication interference have to be considered together with both the network connectivity and data gathering strategy. In this work, two sensor scheduling protocols, *Centralized Adaptive Scheduling Algorithm* (CASA) and *Distributed Adaptive Scheduling Algorithm* (DASA), are proposed to address the application scenario of typical surveillance systems in a cluster-based network topology, where both connectivity and coverage constraint are taken into consideration to achieve performance balance.

For the CASA scheme, given the local information such as neighboring connectivity, the round determination problem may be solved centrally by the clusterheads. For the DASA scheme, as the clusterhead broadcasts a message to start the scheduling assignment, each sensor initializes a random waiting timer with a value which is related to the cluster topology and the neighbor information. If the random waiting timer expires, then the sensor broadcasts a message proclaiming that it is a good candidate to be a group member, which also serves to notify its neighbors that it has a higher priority for the sensing task. Based on the received messages from its neighboring cluster members, each cluster member may use the data gathering strategy (detailed in Section 3.3) to schedule itself to a specific round.

In order to facilitate performance evaluation of a sensor scheduling protocol design, two analytical models, a neural network model and a probabilistic model, are proposed. For the CASA approach, a neural network model is built up to approximate the desire performance. For the DASA approach, a probabilistic model using the concept of geometry is applied to abstract the properties of the algorithm. Moreover, based on the analysis, the sensor lifetime and cluster lifetime is further explored to show how the operations of the proposed schemes may prolong the network lifetime.

The organization of this paper is as follows: Section 2 reviews the current literature on the sensor scheduling management. Section 3 describes the system model and algorithm for sensor scheduling in a cluster-based network topology. In Section 4, a neural network model and a probabilistic model are built up to approximate the desire performance and estimate the sensing rounds of the proposed schemes. Section 5 summarizes the performance of the proposed scheduling methodology. Finally, Section 6 draws conclusions and shows future research directions.

2. Literature Review

A large number of sensor scheduling and coverage maintenance protocols have been proposed [8-35]. However, due to the sensing objectives, these management protocols can be different. Yan *et al*. [1] presented an energy-efficient sensing protocol to achieve the desired sensing coverage. Nodes decide their active periods by exchanging reference points among neighbors. In [2], the authors investigated coverage intensity of the proposed sleep scheduling protocols. Ren *et al*. [3] provided a generic analytical framework that can be widely used for sensing scheduling protocol design with detection quality requirements. Turau *et al*. [4] tried to route packet with the minimum time and energy and aimed to distribute the transmission time slots dynamically among sensor nodes such that the network congestion can be avoided.

Hohlt *et al.* [5] proposed a scheduling scheme for considering energy savings in a data collection process. Schrage *et al.* [6] applied an ant colony optimization method for scheduling the visiting order of targeted areas in the sensing field such that their energy consumptions are minimized. Decker *et al.* [7] developed a scheduler to manage the competition for resources among different sensing tasks at a single sensor node. Chamberland *et al.* [8] investigated the relationship between sleeping duration, detection delay and energy consumption in a stationary sensing field. References [9, 10, 11] are clustering-based protocols that attempt to minimize the energy dissipation in sensor networks.

Cheng *et al.* [12] proposed a bio-inspired scheduling scheme which is a kind of adaptive scheduling scheme which uses only local information for making scheduling decisions. Premkumar *et al.* [13] considered the problem of quickest detection of an intrusion using a sensor network, keeping only a minimal number of sensors active. In [14] and [15], randomized scheduling algorithms are proposed for monitoring a field to detect intrusion objects. The authors study the performance of the randomized scheduling algorithm and explore the impact of the size of intrusion object on the sensor network's configuration.

Since energy efficiency and reasonable sensing coverage can be achieved by exploiting the sensing spatial redundancy, redundant sensors may be turned off to save energy [16, 17, 18, 19]. However, the network connectivity is not considered in those schemes. In order to further reduce energy and computational overhead, some scheduling schemes [2, 16, 19, 20] operate without the location information or time synchronization. Although the joint problem of coverage and connectivity is considered in [21, 22, 23, 24, 25], the optimization of the sensing spatial redundancy is not taken into account. A survey of energy-efficient scheduling mechanisms in sensor networks is detailed in [26].

In contrast, the approaches of this paper consider coverage, connectivity, and sensing spatial redundancy simultaneously in order to improve energy efficiency in a hierarchical network structure. For the CASA approach, the clusterhead collects local topology information to manage the sensing schedule centrally. By approximating the network behavior throughout the neural network learning process, the clusterhead may be able to roughly predict the performance of the scheduling management. For the DASA approach, the setting of the random waiting timer allows each sensor to exploit the information about coverage, connectivity, and sensing spatial redundancy such that a balance of network resources can be maintained. Due to the randomized property of the waiting timer, the probabilistic model is proposed to abstract global network behavior. The comparison of the proposed approaches and the other cluster-based schemes [10][11] is further discussed in Section 5.

3. Dynamic Sensor Scheduling Algorithms

This section describes two scheduling management schemes for organizing the sensing tasks, the Centralized Adaptive Scheduling Algorithm (CASA) and the Distributed Adaptive Scheduling Algorithm (DASA). The main assumptions of the network are: (1) All sensors are homogeneous with the same transmission range; (2) The sensors are fixed without location information; (3) Symmetric communication channel: all links between sensors are bidirectional; (4) All sensors perform the sensing task periodically. Note that there are no base stations to coordinate or supervise activities among sensors.

3.1. Cluster Formation for Scheduling Management

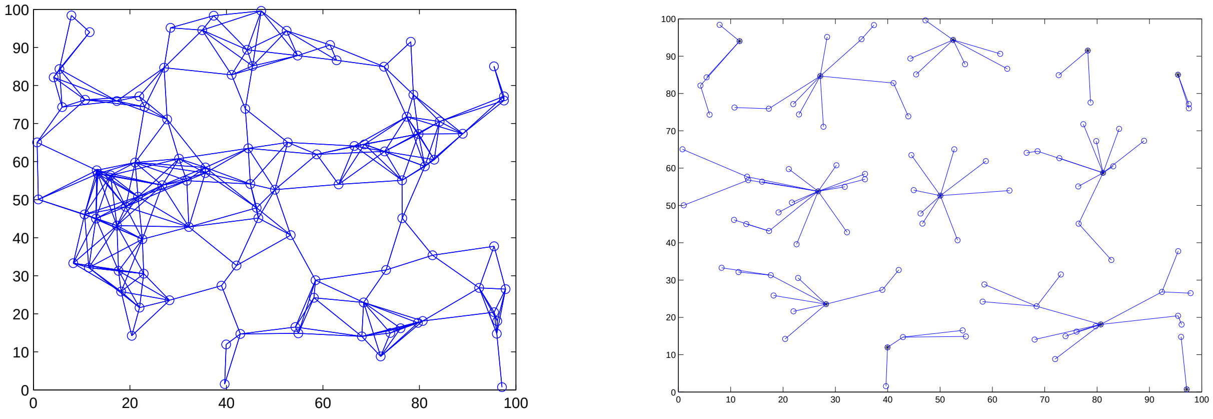
When sensors of a network are first deployed, they may apply the Clustering Algorithm via Waiting Timer (CAWT) from [27] to partition the sensors into clusters. Each sensor sets a random waiting timer, broadcasts its presence via a '*Hello*' signal, and listens for its neighbor's '*Hello*.' The sensors that hear many neighbors are good candidates for initiating new clusters; those with few neighbors should choose to wait. By adjusting randomized waiting timers, the sensors can coordinate themselves into sensible clusters, which can then be used as a basis for further communication and data processing.

Sensors update their neighbor information (i.e. a counter specifying how many neighbors it has detected) and decrease the random waiting time based on each 'new' *Hello* message received. This encourages those sensors with many neighbors to become clusterheads. The updating formula for the random waiting time of sensor *i* is:

$$WT_i^{(k+1)} = \gamma \cdot WT_i^{(k)}, \tag{1}$$

where $WT_i^{(k)}$ is the waiting time of sensor *i* at time step *k* and $0 < \gamma < 1$ is inversely proportional to the number of neighbors. Therefore, if the timer expires, then the sensor declares itself to be a clusterhead, a focal point of a new cluster. However, events may intervene that cause a sensor to shorten or cancel its timer. For example, whenever the sensor detects a new neighbor, it shortens the timer. On the other hand, if a neighbor declares itself to be a clusterhead, the sensor cancels its own timer and joins the neighbor's new cluster.

Figure 1. The connectivity of the network (left); clusters are formed in a random network of 100 sensors with $R/l = 0.17$ (right).



After applying the CAWT, there are three different kinds of sensors: (1) the clusterheads (2) sensors with an assigned cluster ID (3) sensors without an assigned cluster ID, which will join any nearby cluster and become 2-hop sensors. In this phase, each sensor initiates two rounds of local flooding to its 1-hop neighboring sensors, one for broadcasting sensor ID and the other for broadcasting cluster ID, to select clusterheads and form 2-hop clusters. Figure 1 shows the network connectivity and cluster formation of a random network of 100 sensors with $R/l = 0.17$, where R/l is the ratio of transmitting range *R* to the side length *l* of the square. Thus, the topology of the ad-hoc network is now represented by a hierarchical collection of clusters.

Assume that a cluster of sensor nodes share a common view of a local clock time [28], so that all these nodes can coordinate in the sensor scheduling operation. Given the cluster-based network topology, each cluster member is assumed to sense only once during a sensing cycle T_{cycle} in a cluster. That is, $T_{cycle} = \sum_{\ell=1}^{N_{RG}} T_{RG_\ell}$, where N_{RG} is the number of sensing rounds in a cluster and T_{RG_ℓ} is the sensing duration of round ℓ. For sensor scheduling management, there are three kinds of sensors: (1) sensing nodes: executing the sensing task; (2) relay nodes: maintaining the cluster connectivity for intra-cluster communication; (3) gateway nodes: maintaining the network connectivity for inter-cluster communication. Note that the relay nodes may execute the sensing task in the later round and the gateway nodes may perform the sensing task and the relay transmission during the scheduling operation.

3.2. Centralized Adaptive Scheduling Algorithm (CASA)

There are many possible data gathering strategies to accomplish the sensing tasks. In many applications, the monitored area may be used to determine the group members in a specific round. For instance, since the core sensing field is covered by the sensing area of 1-hop cluster members in the cluster-based topology, the round determination problem can be expressed by the coverage subject to the number of 1-hop cluster members covered in the round sensing area,

$$\min_{i \in RG_\ell} \quad \bigcup O_i \tag{2}$$

subject to: $i, j \in RG_\ell, j \notin S_b^{(i)}, ((\cup H_1(i) + RG_\ell) \cap H_1) = H_1$, where O_i represents the coverage overlap of sensor *i* with other round group members, $H_1(i)$ is the set of neighboring 1-hop cluster members of sensor *i*, RG_ℓ is the set of round group members of round ℓ, sensors *i* and *j* belong to the set of round group members RG_ℓ, H_1 is the set of 1-hop cluster members, and $S_b^{(i)}$ is the set of neighboring sensors of sensor *i*. The rationale for using the constraint in (2) is to avoid heavy overlap between sensors in the same round. However, without location information, it may be not easy to cover the desired area and minimize the coverage overlap in each round. Hence, the optimization of the coverage may be modified to satisfy the problem constraint by the following scheme.

The Scheduling Scheme

When developing the sensing schedule, two rounds of local flooding are initiated in order to gather topology information for the clusterhead in the 2-hop cluster structure. Hence, given the local information such as neighboring connectivity, a clusterhead may choose a 2-hop cluster member and a 1-hop relay node to initialize the proposed scheduling algorithm. After that, the clusterhead may randomly pick 1-hop cluster members, which have no communication links with the chosen group members, as the new round group members. The purpose for this selection policy is to reduce the overlap between group members in the same round. Note that the relay node can be selected as the group member in the following round since it is not responsible for sensing at this round. If all the 2-hop cluster members have been selected for initializing the sensing rounds, the clusterhead will select a 1-hop cluster member for starting the new round. This procedure is repeated until all of the cluster members have been assigned. Then, with a common local clock time in the cluster, the clusterhead triggers two rounds of 1-hop flooding for broadcasting the sensor scheduling information throughout the 2-hop cluster topology.

Observe that the overlap is only approximately minimized; in our experiments we have noticed that the answers tend to be close to the optimal. The pseudo-code of the proposed algorithm is presented in Table 1, where $H_1(m)$ is the set of neighboring 1-hop cluster members of sensor *m* in a cluster, $H_2(n)$ is the set of neighboring 2-hop cluster members of sensor *n* in a cluster, RG_ℓ is the set of round group members of round ℓ, *U* is the set of cluster members, H_1 and $H_1^{'}$ are the sets of 1-hop nodes, H_2 is the set of 2-hop nodes, and N_{RG} is the number of sensing rounds in a cluster.

Table 1. The CASA Scheduling Scheme

```
Assign $N_{RG} = 0$, $\ell = 1$;
**while** ($U \neq \phi$) **do**
{
   $H_1^{'} = H_1$;
   $RG_\ell = \phi$;

  \* Selecting 2-hop round members *\
  **if** ($H_2 \neq \phi$)
  {
       $i = \text{argmax}_k |H_1(k)|, \forall\ k \in H_2$;
       $RG_\ell = \{i\}$;
       $H_1^{'} = H_1^{'} - H_1(i)$;
       $H_2 = H_2 - i$;
  }

  \* Selecting 1-hop round members *\
  **while** ($H_1^{'} \neq \phi$) **do**
  {
       Pick sensor $m$, $m \in H_1^{'}$;
       $RG_\ell = RG_\ell \cup m$;
       $H_1^{'} = H_1^{'} - H_1(m)$;
       $H_1 = H_1 - m$;
  }

  $U = U - RG_\ell$;
  $\ell = \ell + 1$;
  $N_{RG} = N_{RG} + 1$;
}
```

Maintenance of Network Connectivity

After establishing the sensing schedule in each cluster, network connectivity may be maintained with two phases of operation (observation and confirmation phases). The period of the observation phase may last several sensing cycles (nT_{cycle}, $n > 1$), which allow the sensors to learn about the scheduling operation of their neighboring sensors in nearby clusters. During the observation phase, each sensing node and relay node executes a 1-hop broadcast at the beginning of its active period in the sensing schedule such that the sensing nodes can assign the gateway sensors for inter-cluster communication and data dissemination. The broadcast message includes the sensor node ID and the sensing cycle time T_{cycle} for the gateway nodes to initialize the next relay transmission.

There are four possible scenarios when determining the gateway nodes: (1) When the sensing node receives only one broadcast message from an active node in the nearby cluster during its sensing period, these two nodes form a pair of distributed gateways. Hence, a sensing node or a relay node in the nearby cluster may be a gateway under this condition; (2) If the sensing node receives multiple broadcast messages from the same nearby cluster, the nearest active node in this specific cluster might be chosen as a gateway node based on distance information, which could be estimated by the received signal strength. Similar to Scenario 1, a sensing node or a relay node may be a gateway in this case; (3) When no broadcast message is received during the sensing period, the sensing node may choose the nearest node in an adjacent cluster as a gateway node; (4) If the clusters are too far apart (outside the range of communication R), no gateway sensors will be assigned.

Built upon the learning process in the observation phase, the sensing node and the candidate of the gateway node acknowledge the role assignment in the confirmation phase. Thus, each pair of distributed gateways send 1-hop broadcast messages to confirm the gateway selection with each other. Accordingly, the result of gateway selection is that each round group member assigns a single member of each nearby clusters such that network connectivity during the sensor scheduling operation may be assured. Therefore, the CASA approach provides a virtual backbone for sensing coverage and network connectivity maintenance. The procedures of gateway selection is depicted in Figure 2. Figure 2 (a) and Figure 2 (b) describe the operation period of a pair of distributed gateways in Scenarios 1 and 3, respectively. Given the local common clock, the time stamp of the received message, and the duration of the sensing cycle T_{cycle}, the sensing nodes A and B may work cooperatively as a pair of distributed gateways to adjust their active periods for data dissemination. An example which highlights network coverage and connectivity analysis is further illustrated in Section 5.

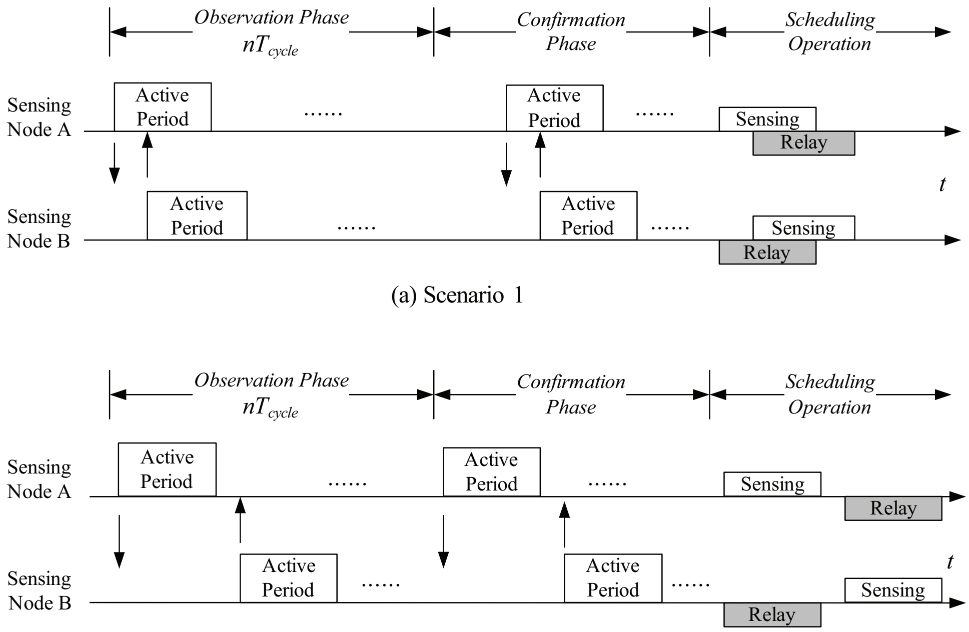
3.3. Distributed Adaptive Scheduling Algorithm (DASA)

The Setting of Waiting Timer

The distributed method operates much like the CAWT [27] in utilizing a random timer. As the clusterhead broadcasts a message to start the scheduling assignment, sensor *i* initializes a random waiting timer with a value $WT_i^{(0)}$:

$$WT_i^{(0)} = \frac{1}{N_{hop}} \cdot T_i \cdot \beta^{N_b^{(i)}}, \tag{3}$$

Figure 2. The procedures of selecting a pair of distributed gateways in Scenario 1 (a) and Scenario 3 (b).



(a) Scenario 1

(b) Scenario 3

which is related to the cluster topology and the neighbor information. Note that T_i is a sample from the distribution $C + \lambda \cdot U(0, 1)$, where C and λ are positive numbers, which are used to specify the sampling range of the waiting time, and $U(0, 1)$ is a uniform distribution. N_{hop} is the number of hops from the clusterhead to the cluster member, $N_b^{(i)}$ is the number of neighboring cluster members of sensor *i*, β is a positive number with $1 < \beta$. The rationale for the settings in equation (3) is that, due to the overlap of sensing area in a cluster, the coverage overlap of a 1-hop cluster member is usually larger than that of a 2-hop cluster member. This suggests that a 2-hop cluster member may be a good candidate to initialize a round group. On the other hand, a 1-hop cluster member may choose to wait and join the round group later. Furthermore, a cluster member with more neighbors may have a lower priority to execute the sensing task since its sensing area may be covered by the nearby cluster members.

If the random waiting timer expires (i.e. $WT_i = 0$), then sensor *i* broadcasts a message proclaiming that it is a good candidate to be a group member, which also serves to notify its neighbors that it has a higher priority for the sensing task. For its neighboring sensor *j*, the update formula of the random waiting timer may be given by:

$$WT_j^{(k+1)} = (1 + \alpha) \cdot WT_j^{(k)}, \tag{4}$$

where $WT_j^{(k)}$ is the waiting time of sensor *j* at time step *k* and $0 < \alpha$. The setting of α can be attributed to the fact that the neighboring nodes receiving the broadcasting message increase their waiting timers such that they may work in different rounds and the sensing redundancy may be suppressed in each sensing round.

The Scheduling Scheme

The message for communication among the cluster members consists of: (1) the ID of the sending sensor, (2) the round ID of the sending sensor, and (3) the relay round ID of the selected relay sensor. At the beginning, the round ID and the relay round ID of each sensor is one and zero, respectively. Based on equation (3), 1-hop cluster members set longer waiting times compared with 2-hop cluster members. When a 2-hop timer expires, the 2-hop cluster member broadcasts a message with the initial round ID 1 and selects its 1-hop parent cluster member as a relay node from the cluster topology. Thus, the selected relay node records its relay round ID and will execute the data dissemination in that sensing round. Since a 1-hop cluster member can report the collected information to the clusterhead directly, relay nodes are not necessary in this case. In order to reduce the overlap of the coverage in each sensing round, the neighboring nodes receiving the broadcasting message update their waiting timers and increase their round IDs by 1 such that they may work in different rounds.

In order to maintain the correct round ID information when receiving multiple messages among neighboring nodes, the ID updating strategy may be described as follows. Given a cluster member with round ID *u* and a message sent by a cluster member with round ID *v*, the cluster member with round ID *u* may update its round ID by $u = v + 1$ if the round ID $u \leq v$. Otherwise, the received message is ignored. Accordingly, an update criterion for the sensing node can be derived:

$$\text{Round ID } u = \begin{cases} v+1, & \text{if } u \leq v \\ u, & \text{otherwise.} \end{cases} \tag{5}$$

By following the above procedures, the round IDs and relay IDs can be determined for each cluster member. Based on the received broadcast messages for updating round ID information from 1-hop cluster members, the clusterhead can obtain the number of sensing rounds N_{RG} in a sensing cycle. This is because the number of sensing rounds N_{RG} is equal to the largest round ID of the 1-hop cluster members. Therefore, given a common local clock time in the cluster, the clusterhead may generate two rounds of local flooding for broadcasting the sensor scheduling information throughout the cluster. The procedures of sensor scheduling is outlined in the DASA of Figure 3.

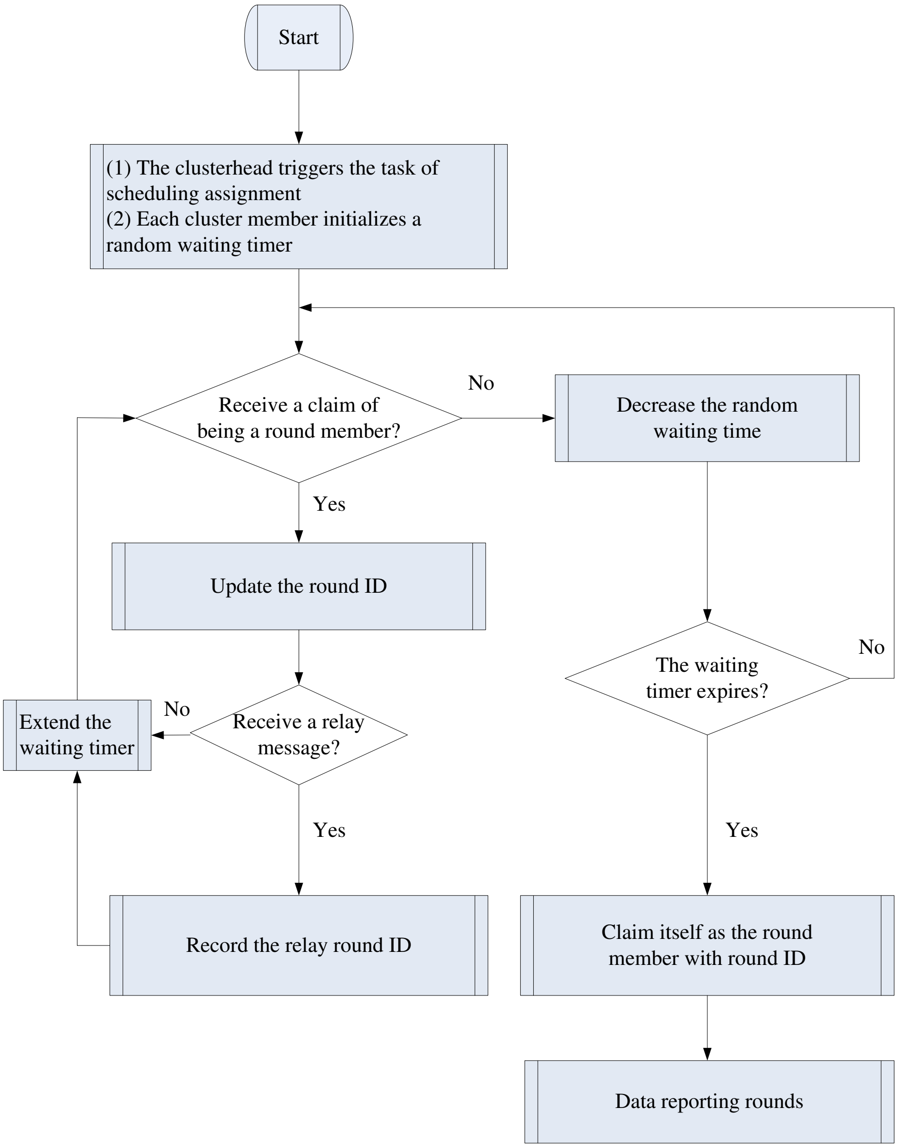
Figure 4 illustrates the updating process of round ID among the cluster members for determining the round group members for the first round. At the beginning, sensor 73 broadcasts a message with the initial round ID 1 and selects its 1-hop parent cluster member, sensor 11, as a relay node. Then, sensor 11 records its relay round ID and updates its round ID.

Under the operation of the DASA scheme, pairs of distributed gateways for inter-cluster communication can be decided by applying the same approach as described in Section 3.2. The sensing coverage and connectivity performance will be further explored in Section 5.

4. Analysis

Two analytical tools are provided to estimate the number of sensing rounds of the proposed schemes. For the CASA approach, a neural network model is built up to approximate the desire performance. For the DASA approach, a probabilistic model using the concept of geometry and the Lindeberg Theorem [29] are applied to abstract the properties of the algorithm. Moreover, based on the analysis, the sensor

Figure 3. Virtual sensor scheduling flowchart for the DASA algorithm.


Start
(1) The clusterhead triggers the task of scheduling assignment
(2) Each cluster member initializes a random waiting timer
Receive a claim of being a round member?
No
Decrease the random waiting time
Yes
Update the round ID
The waiting timer expires?
No
Extend the waiting timer
No
Receive a relay message?
Yes
Yes
Record the relay round ID
Claim itself as the round member with round ID
Data reporting rounds

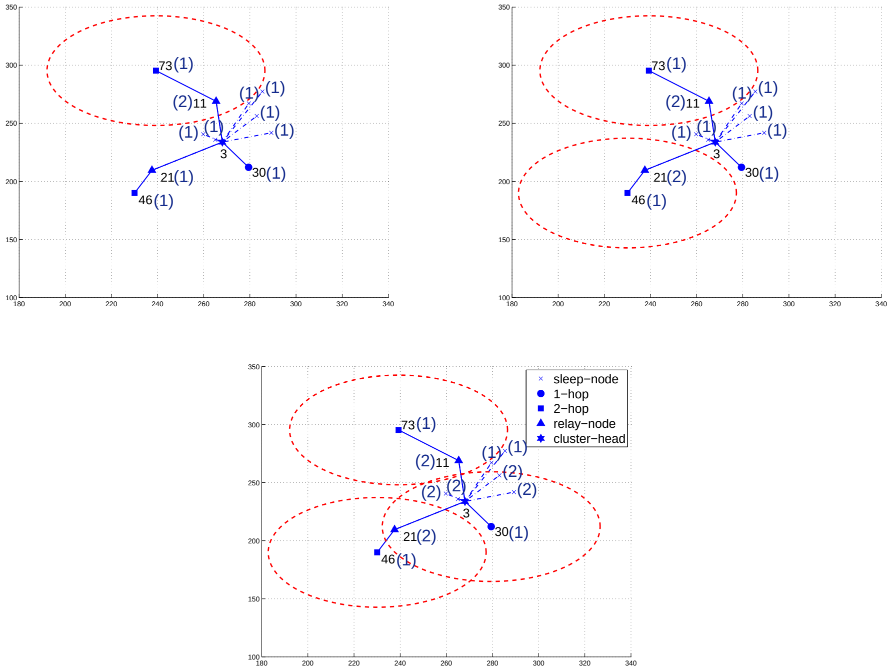
lifetime and cluster lifetime is further explored to show how the operations of the proposed schemes may prolong the network lifetime.

4.1. Neural Networking for the Centralized Approach

Backpropagation Learning Algorithm

This subsection reviews the neural network algorithm [30] for analyzing the performance of the centralized method. Assume that the network under consideration has a general architecture with three layers of neurons. In our case, input and output layer neurons are linear, whereas neurons in the hidden

Figure 4. The round ID updating process of the DASA algorithm; the $(\cdot)$ represents the round ID.



layer are log-sigmoidal. Let the vector pairs in $\mathcal{T}$ be sample representation of the unknown function $f : \mathcal{R}^n \to \mathcal{R}^p$:

$$T = \{(X_k, D_k)\}_{k=1}^{Q}, \tag{6}$$

where *n* is the neuron index range in the input layer, *p* is the neuron index range in the output layer, $X_k \in \mathcal{R}^n$, $D_k \in \mathcal{R}^p$, *Q* is the number of training vector pairs, and *k* is the iteration index. Note that D_k is the desired vector response for the network input X_k. Thus, the mean square error of the entire training set is:

$$\mathcal{E} = \frac{1}{Q}\sum_{k=1}^{Q}\mathcal{E}_k \tag{7}$$

where $\mathcal{E}_k = \frac{1}{2}E_k^T E_k$, and E_k is the instantaneous error of the training pair (X_k, D_k). Based on the above description, the update of neuron activations can be formulated as follows.

For the hidden layer:

$$z_h^k = \sum_{i=0}^{n} w_{ih}^k x_i^k, \;\; h = 1, \ldots, q \tag{8}$$

$$\mathcal{S}(z_h^k) = \frac{1}{1 + e^{-z_h^k}}, \;\; h = 1, \ldots, q \tag{9}$$

$$w_{ih}^{k+1} = w_{ih}^k + \eta\left(-\frac{\partial \mathcal{E}_k}{\partial w_{ih}^k}\right). \tag{10}$$

For the output layer:

$$y_j^k = \sum_{h=0}^{q} \frac{w_{hj}^k}{1 + e^{-z_h^k}}, \;\; j = 1, \ldots, p \tag{11}$$

$$\mathcal{S}(y_j^k) = y_j^k,\ j = 1, \ldots, p \tag{12}$$

$$w_{hj}^{k+1} = w_{hj}^k + \eta\left(-\frac{\partial \mathcal{E}_k}{\partial w_{hj}^k}\right). \tag{13}$$

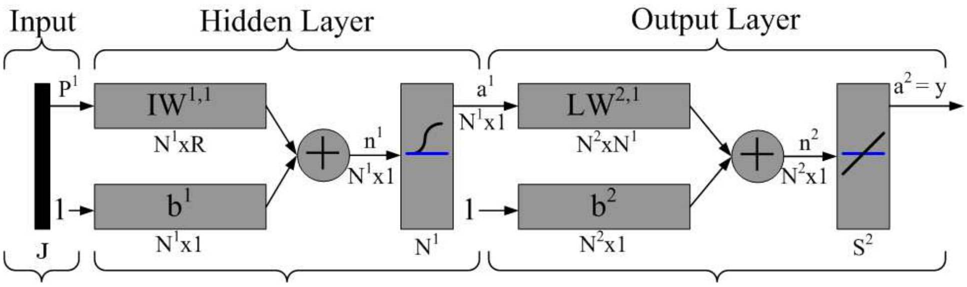
Note that *q* is the neuron index range in the hidden layer, x_i^k and y_j^k are the *i*th and *j*th component of the input vector X_k and the output vector Y_k, respectively, w_{ih}^k and w_{hj}^k are the biases of the hidden and output neurons, respectively, $\mathcal{S}(\cdot)$ is the signal function, and η is the learning rate in the back-propagation algorithm.

Estimation of the Number of Sensing Rounds

In order to estimate the number of schedule rounds in a given topology when applying the CASA scheme, the three-layer perceptron neural network is presented. For selecting the network parameters (weights and biases) that best approximate a given function, the backpropagation learning algorithm is considered to minimize the mean square error performance as described in (7).

Figure 5 illustrates the perceptron network architecture. Note that *J* represents the number of input neurons, which may denote the number of 2-hop cluster members, the number of relay nodes, and the number of 1-hop cluster members. N^1 denotes the number of hidden neurons. In the output layer, N^2 represents the number of neurons, which may denote the network approximation results. Moreover, let IW and LW be the input weight matrix and layer weight matrix for the hidden layer and the output layer, respectively. Let b^1 and b^2 be bias vectors for the hidden layer and the output layer, respectively. Established upon the developed neuron network model, the behavior of the CASA scheme may be abstracted with sensible settings, which is further discussed in Section 5.

Figure 5. The three-layer perceptron neural network architecture for analyzing the CASA scheme.



4.2. *Probabilistic Model (PM) for the Distributed Approach*

Overlap of Geometrical Figures

This subsection introduces a particular problem considering the mean and variance of the overlap of geometrical figures [31]. Given a number of circles placed at random on a plane so that each one may have some or all of its area inside a target area, with reference to the bombing studies, [32] uses the concept of geometry and of probability to explore the fundamentals of this type of problem. The result is represented as the following theorem.

Theorem 1 *Let X be a random Lebesgue measurable subset of* n*-dimensional Euclidean space E_n, with measure $\mu(X)$. For any point x of E_n, let $p(x) = Pr(x \in X)$. Assuming that the function $g(x, X)$ is a measurable function of the pair (x, X) with $g(x, X) = 1$ for $x \in X$ and zero elsewhere, the expected value of the measure* X *is the Lebesgue integral of the function $p(x)$ over E_n.*

Suppose that *A* and *C* are the interior of the closed curves. Let the subset *X* be the part of a region *A* in E_2 which is covered by *z C*'s dropped independently and randomly. Denote a reference point *Q* as the centra of the area *C* and assume that there is a frequency distribution $\phi(x, y)$ of the position (x, y) of *Q*. Based on the above assumptions, now we consider the moments of the area $Y = A - X$ (i.e. the area of *A* not covered by the *z C*'s).

Referring to Theorem 1, the probability of a point (x_1, y_1) in *Y* not being covered by a *C* is:

$$q(x_1, y_1) = \int_{T-\overline{C}(x_1,y_1)} \int \phi(x, y) dx dy, \tag{14}$$

where $T - \overline{C}(x_1, y_1)$ is the part of *T* exterior to the area *C*. Therefore, the first moment of *Y*, in the case of *z C*'s, yields:

$$\mu_Y^{(1)} = \int_A \int q^z(x_1, y_1) dx_1 dy_1. \tag{15}$$

similarly, for the *m*th moment, the probability that the points (x_1, y_1), (x_2, y_2), $\ldots$, (x_m, y_m) are not covered by a *C* is:

$$q(x_1, y_1, x_2, y_2, \ldots, x_m, y_m) = \int_{T-\overline{C}_1-\overline{C}_2-\ldots-\overline{C}_m} \int \phi(x, y) dx dy, \tag{16}$$

where $T - \overline{C}_1 - \overline{C}_2 - \ldots - \overline{C}_m$ is the area of *T* outside *C*'s centered at (x_1, y_1), (x_2, y_2), $\ldots$, (x_m, y_m). Thus, the *m*th moment in the case of *z C*'s is given by:

$$\mu_Y^{(m)} = \int_A \int \cdots \int_A \int q^z(x_1, y_1, x_2, y_2, \ldots, x_m, y_m) dx_1 dy_1 dx_2 dy_2 \ldots dx_m dy_m. \tag{17}$$

Accordingly, in our case we may interpret *T* as the sensing field in a cluster, let *A* be the sensing field of a given sensor, let *X* be the area covered by its neighboring cluster members, and *Y* will represent the sensing area of a given sensor not covered by its neighboring cluster members. Denote the parameters *z* and *m* as the number of unconnected cluster members and the neighboring cluster members, respectively. Therefore, the *m*th moment $\mu_Y^{(m)}$ may describe the fraction of *A* of a given sensor not covered by its *m* neighboring cluster members.

Lindeberg Theorem

This subsection reviews the probability that is used when analyzing the performance of the model. Readers may refer [29] for a complete discussion and proof of the theorem.

Suppose for each *n*

$$(X_{11}, X_{12}, \ldots, X_{1r_1})$$

$$\begin{gathered}(X_{21}, X_{22}, \ldots, X_{2r_2}) \\ \vdots \\ (X_{n1}, X_{n2}, \ldots, X_{nr_n})\end{gathered} \tag{18}$$

are independent random vectors. The probability space may change with *n* and (18) is called a *Triangular Array* of random variables. Put $S_n = X_{n1} + \cdots + X_{nr_n}$. In the network application, let X_{ni} be X_i and let X_i take the values 1 and 0 with probability p_i and $q_i = 1 - p_i$. We may interpret X_i as an indicator that sensor *i* is chosen to be a round member with probability p_i and S_n is the number of members in a round.

Denote $Y_i = X_i - p_i$. Hence:

$$S_n^Y \equiv \sum_{i=1}^{n} Y_i = \sum_{i=1}^{n} X_i - \sum_{i=1}^{n} p_i = S_n - \sum_{i=1}^{n} p_i, \tag{19}$$

$$E[Y_i] = E[X_i] - p_i = 0, \tag{20}$$

$$\sigma_{Y_i}^2 = \sigma_{X_i}^2 = p_i(1 - p_i), \tag{21}$$

$$\sigma_{s_n}^2 = \sum_{i=1}^{n} \sigma_{Y_i}^2 = \sum_{i=1}^{n} \sigma_{X_i}^2 = \sum_{i=1}^{n} p_i(1 - p_i). \tag{22}$$

For our case, the Lindeberg condition [29] reduces to:

$$\lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{s_n^2} \int_{|Y_i|\geq \epsilon s_n} Y_i^2 dP \leq \lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{s_n^2} \int_{|Y_i|\geq \epsilon s_n} dP = 0, \tag{23}$$

which holds because all the random variables are bounded by 1 and $[|Y_i| \geq \epsilon s_n] \to 0$ as $n \to \infty$.

Theorem 2 *Suppose that Y_i is an independent sequence of random variables and satisfies $E[Y_i] = 0$, $\sigma_{Y_i}^2 = E[Y_i^2]$, $S_n^Y = \sum_{i=1}^{n} Y_i$, and $s_n^2 = \sum_{i=1}^{n} \sigma_{Y_i}^2$. If the Lindeberg condition (23) holds, then $S_n^Y / s_n \to \mathcal{N}(0, 1)$.*

Observe that p_i may be described by the overlap fraction of sensor *i* since the sensors with less overlap between its neighboring sensors has a larger chance to be selected as a round group member in the round competition, which coincides with the operation of the DASA and the setting of the waiting timer.

Estimation of the Number of Sensing Rounds

Assume that each sensor will be grouped with probability $p_i^{(k)}$ at iteration *k*. Denote the collection of cluster members for selecting the round members at iteration *k* by V_k. Since the round members are selected and removed at each iteration, the collection of sensors at the next iteration, V_{k+1}, is simply a new and smaller cluster. Thus, by Theorem 2, the distribution of the number of round members at iteration *k* can be approximated by $\mathcal{N}(\mu_k, \sigma_k^2)$ with $\mu_k = \sum_{i=1}^{m_k} p_i^{(k)}$ and $\sigma_k^2 = \sum_{i=1}^{m_k} p_i^{(k)}(1 - p_i^{(k)})$. Once the procedure terminates, the number of iterations is an estimate of the number of rounds formed in the cluster. A statement of procedures for analyzing the DASA is given in Table 2.

Table 2. Procedures of the PM model for analyzing the DASA.

```
a) Let $n$ be the number of cluster members.
b) $r_k$ is the number of sensors to be removed and $m_k$ is the number of sensors
   remaining at iteration k
c) Assign the probability $p_i^{(k)}$ to sensor i, proportional to the fraction not covered by its
   neighboring cluster members. That is, $p_i^{(k)} \propto Y/A$, as described in Theorem 1.
d) Assign $k = 0$, $m_0 = n$, $r_0 = 0$.
   while $(m_k - r_k) > 0$
        $r_k = \lceil \sum_{i=1}^{m_k} p_i^{(k)} \rceil$ as detailed in Theorem 2,
        $m_{k+1} = m_k - r_k$,
        form members of this round,
        update $p_i^{(k)}$,
        $k = k + 1$.
   end
$^*\lceil \cdot \rceil$ is the ceiling function.
```

4.3. Sensor Lifetime and Cluster Lifetime

The main objective of the dynamic sleep scheduling approaches is to extend the lifetime of the clusters so that the network may remain functional longer. Say that the cluster lifetime ends when the first sensor in the cluster fails. Therefore, it is worthwhile to understand the lifetime of individual sensors.

Depending on the traffic model of the network, the expected sensor lifetime may be different. Suppose that the sensors measure periodically and transmit the data back to the clusterhead for further processing with a steady traffic. We also assume that the clusterhead collects the information from cluster members and communicates with the base station with a steady traffic flow [33]. Thus, the expected lifetime $E[\triangle T_i^{(j)}]$ of sensor *i* at round *j* in a sensing cycle is:

$$E[\triangle T_i^{(j)}] = p_i^{(j)} \cdot \left(\frac{E_i^{(j)} - E_i^{(j+1)}}{P_i} \right),$$

where $p_i^{(j)}$ is the probability of sensor *i* to be a round member at round *j*, P_i is the power dissipation of sensor *i*, and $E_i^{(j)} - E_i^{(j+1)}$ is the energy consumption at round *j*. Hence, for sensor *i*, the expected energy consumption in a sensing cycle is

$$E_i^{(sc)} = \sum_j P_i \cdot E[\triangle T_i^{(j)}] \tag{24}$$

and the expected sensor life time of sensor *i* for being a round group member is given by:

$$E_{RG}[T_i] = \frac{E_i^{(0)}}{E_i^{(sc)}} \cdot T_{cycle} \tag{25}$$

$$= \frac{E_i^{(0)}}{\sum_j P_i \cdot E[\triangle T_i^{(j)}]} \cdot T_{cycle}, \tag{26}$$

where $E_i^{(0)}$ is the initial energy of sensor *i* and $E[\triangle T_i^{(j)}]$ is the expected lifetime of sensor *i* at round *j* in a sensing cycle T_{cycle}.

Accordingly, the impact of the sleep scheduling approach on cluster lifetime is further examined. For a cluster without sleep scheduling strategy, the expected lifetime of sensor *i* is:

$$\widetilde{E}_{WS}[T_i] = \frac{E_i^{(0)}}{P_i}. \tag{27}$$

For a cluster with sleep scheduling strategy, the expected lifetime of sensor *i* is:

$$\begin{aligned} E_S[T_i] &= p_i^{(ch)} \cdot \frac{E_i^{(0)}}{P_i} + (1 - p_i^{(ch)}) \cdot E_{RG}[T_i] && (28) \\ &= p_i^{(ch)} \cdot \frac{E_i^{(0)}}{P_i} + (1 - p_i^{(ch)}) \cdot \frac{E_i^{(0)}}{\sum_j P_i \cdot E[\triangle T_i^{(j)}]} \cdot T_{cycle}, && (29) \end{aligned}$$

where $p_i^{(ch)}$ is the probability for sensor *i* to be a clusterhead, which may be related to the operation of the clustering algorithm.

Based upon the definition of the cluster lifetime, the cluster lifetime is equal to the minimum of the expected lifetime of sensors. That is, $\widetilde{L}_{ch} \equiv \min_i\{\widetilde{E}_{WS}[T_i]\}$ and $L_{ch} \equiv \min_i\{E_S[T_i]\}$. To quantitatively measure how well the cluster lifetime are extended, we introduce a parameter, cluster lifetime factor (CLF). The CLF is defined as the ratio of the cluster lifetime with sleep scheduling and to the cluster lifetime without sleep scheduling. Thus, the CLF is:

$$CLF \equiv \frac{L_{ch}}{\widetilde{L}_{ch}} = \frac{\min_i\{E_S[T_i]\}}{\min_i\{\widetilde{E}_{WS}[T_i]\}}. \tag{30}$$

Now we provide an example on how the cluster lifetime can be extended by applying the dynamic scheduling techniques. Assume that sensors of the network have identical initial energy levels and power dissipation. Therefore, the cluster lifetime factor (CLF) yields:

$$CLF = \min_i \left\{ p_i^{(ch)} + (1 - p_i^{(ch)}) \cdot \frac{T_{cycle}}{\sum_j E[\triangle T_i^{(j)}]} \right\}, \tag{31}$$

which shows that $CLF \geq 1$ since the sensing cycle $T_{cycle} \geq \sum_j E[\triangle T_i^{(j)}]$. That means the cluster can last longer by using sleep scheduling schemes, further extending the lifetime of the network.

4.4. Complexity Analysis

This subsection assesses the performance of the proposed schemes in terms of communication and time complexity for network operations.

CASA Scheme

When developing the sensing schedule, two rounds of local flooding are initiated in order to gather topology information for the clusterhead in the 2-hop cluster structure. Hence, the time complexity is $\mathcal{O}(2)$ rounds. Next, the clusterhead triggers two rounds of 1-hop flooding for broadcasting the sensor scheduling information throughout the cluster. At last, two rounds of 1-hop flooding is performed for determining the gateway nodes. Thus, the time complexity is $\mathcal{O}(6)$ rounds..

Consider a sensor, say sensor i, is a clusterhead. Suppose that the total power requirements include the power required to transmit messages E_T and the power required to receive E_R. Therefore, the total energy consumption, E_{sch}, for scheduling management in the cluster is $E_{sch} = N_T \cdot E_T + N_R \cdot E_R$ with:

$$N_T = 1 + 4 \cdot N_C + N_i, \tag{32}$$

$$N_R = N_i + 4 \cdot \sum_{j=1}^{N_C} N_j + \sum_{k \in S_b^{(i)}} N_k, \tag{33}$$

where N_C is the number of sensors in the cluster, N_i is the number of the neighboring sensors of sensor i, and $S_b^{(i)}$ is the index set of neighboring sensors of sensor i. Given the energy consumption analysis above, the communication complexity due to establishing the sensing schedule is $\mathcal{O}(N_C)$. Moreover, when operating the sensing task, the energy consumption is related to the number of active sensors in a round. Therefore, the communication complexity for collecting sensing data is $\mathcal{O}(M_{RG})$, where M_{RG} is the number of round members in the cluster.

Since the algorithm is mainly executed in the clusterhead, the computation cost analysis of a clusterhead is presented. Based on the procedures of the CASA scheduling scheme in Table 1, it includes the operations needed to check the status of the cluster members and to select round group members. Hence, the computation complexity for scheduling management in a clusterhead is $\mathcal{O}(N_C^2)$, where N_C is the number of sensors in the cluster.

In order to show the frequency of operations on the system resource and explore the impact of gathering all the information by the clusterhead, memory usage analysis is provided in terms of information processing perspective. Assume each node has a *L*-byte data packet to transmit. Based on the operation of the CASA scheme, the memory usage M_{CASA} is given by:

$$\begin{aligned} M_{CASA} &= M_T + M_S + M_G \qquad (34)\\ &= (\sum_{j=1}^{N_C} N_j + N_C + |H_1| + \sum_{k \in H_i^{(2)}} N_k + 2N_G) \cdot L, \qquad (35) \end{aligned}$$

where M_T is the memory usage for gathering topology information, M_S is the memory usage for broadcasting the scheduling information, and M_G is the memory usage for gateway selection, N_j is the number of the neighboring sensors of sensor j, N_C is the number of sensors in the cluster, H_1 is the set of 1-hop nodes, $H_i^{(2)}$ denotes the index set of 1-hop cluster members of cluster *i* with neighboring 2-hop cluster members, and N_G is the number of gateway sensors for inter-cluster communication in the cluster.

By using the CASA algorithm, the periodic on-off scheduling problem can be solved efficiently due to a sleeping schedule for each sensor node in a cluster. However, the drawback is using a centralized accumulator host to gather topology information of each sensor such that it can execute the scheduling management. The problem arises when some of the sensors can not transmit the required information to the accumulator host or the accumulator host malfunctions.

DASA Scheme

In the DASA approach, the clusterhead triggers two rounds of 1-hop flooding to initialize the sensor scheduling management process in the 2-hop network topology. Next, 1 round of local flooding is applied for updating the round ID of each node. Then, the clusterhead generates another two rounds of local flooding for broadcasting the sensor scheduling information. Finally, the gateway nodes are selected using two rounds of 1-hop flooding. Therefore, the time complexity is $\mathcal{O}(7)$ rounds.

Accordingly, the total energy consumption is $E_{sch} = N_T \cdot E_T + N_R \cdot E_R$, where:

$$N_T = 2 \cdot (1 + |H_i^{(2)}|) + 3 \cdot N_C, \tag{36}$$

$$N_R = 2 \cdot (N_i + \sum_{j \in H_i^{(2)}} N_j) + 3 \cdot \sum_{j=1}^{N_C} N_j, \tag{37}$$

where $H_i^{(2)}$ denotes the index set of 1-hop cluster members of cluster *i* with neighboring 2-hop cluster members. Hence, the communication complexity due to scheduling management is $\mathcal{O}(N_C)$, N_C is the number of sensors in the cluster. Similar to the CASA scheme, when operating the sensing task, the energy consumption is related to the number of active sensors. Therefore, the communication complexity for collecting sensing data is $\mathcal{O}(M_{RG})$, where M_{RG} is the number of members in each round in the cluster.

Due to the operation of the DASA scheme, the computation burden is distributed among the sensors. Thus, the computation cost analysis is considered with respect to the clusterhead and cluster members, respectively. For the clusterhead, it arranges the sensing schedule based on the largest received round ID. Hence, the computation complexity for updating the round ID in a clusterhead is $\mathcal{O}(N_C)$. For the cluster members, they need to initialize a waiting time, to check if a claim of being a round member is received, to update the round ID, to extend the waiting time, and to check if the waiting timer expires. Therefore, the computation complexity for scheduling management in a cluster member is $\mathcal{O}(N_b^{(j)})$, where $N_b^{(j)}$ is the number of the neighboring cluster members of sensor *j*.

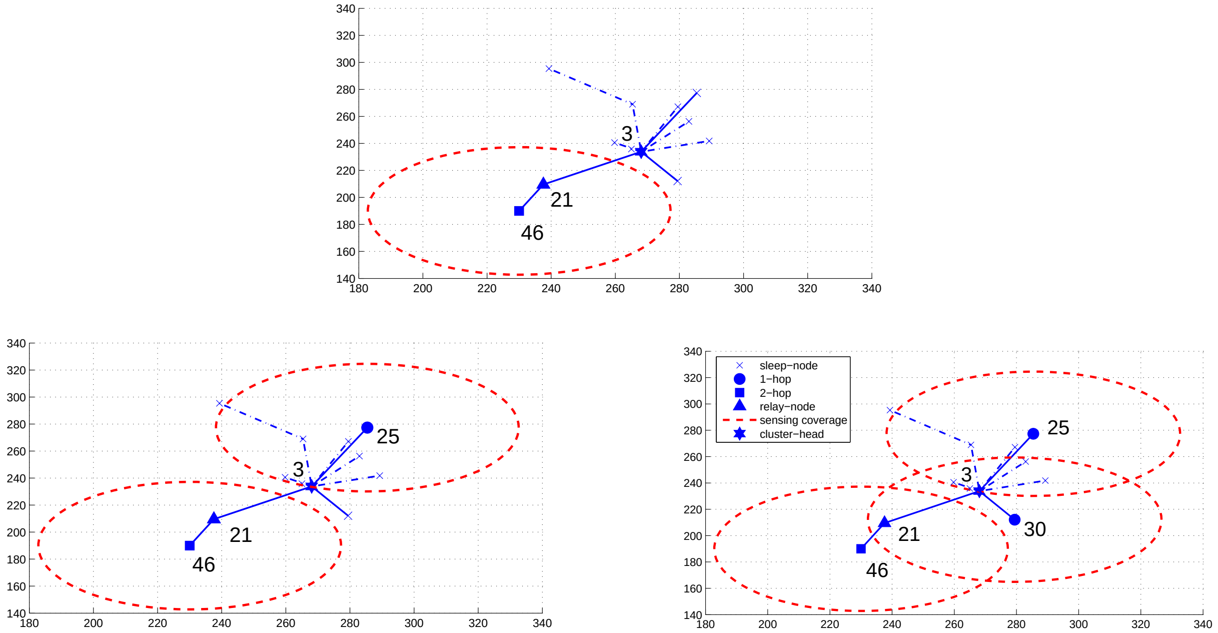
Suppose that each sensor node has a *L*-byte data packet to transmit. According to the operation of the DASA scheme, the memory usage M_{DASA} yields:

$$\begin{aligned} M_{DASA} &= M_I + M_R + M_S + M_G \qquad (38) \\ &= (2 \cdot \sum_{j=1}^{N_C} N_j + 2|H_1| + \sum_{k \in H_i^{(2)}} N_k + 2N_G) \cdot L, \qquad (39) \end{aligned}$$

where M_I is the memory usage for initializing the procedure of scheduling management, M_R is the memory usage for updating the round ID, M_S is the memory usage for broadcasting the scheduling information, and M_G is the memory usage for gateway selection, N_C is the number of sensors in the cluster, H_1 is the set of 1-hop nodes, and N_G is the number of gateway sensors for inter-cluster communication in the cluster.

Note that although the DASA scheme has a higher time complexity due to the round ID updating process, the DASA allows the cluster members to organize themselves into round groups and complete the scheduling assignment automatically with only local neighboring information.

Figure 6. An example of generating a sensing round with the CASA approach.



5. Experimental Results

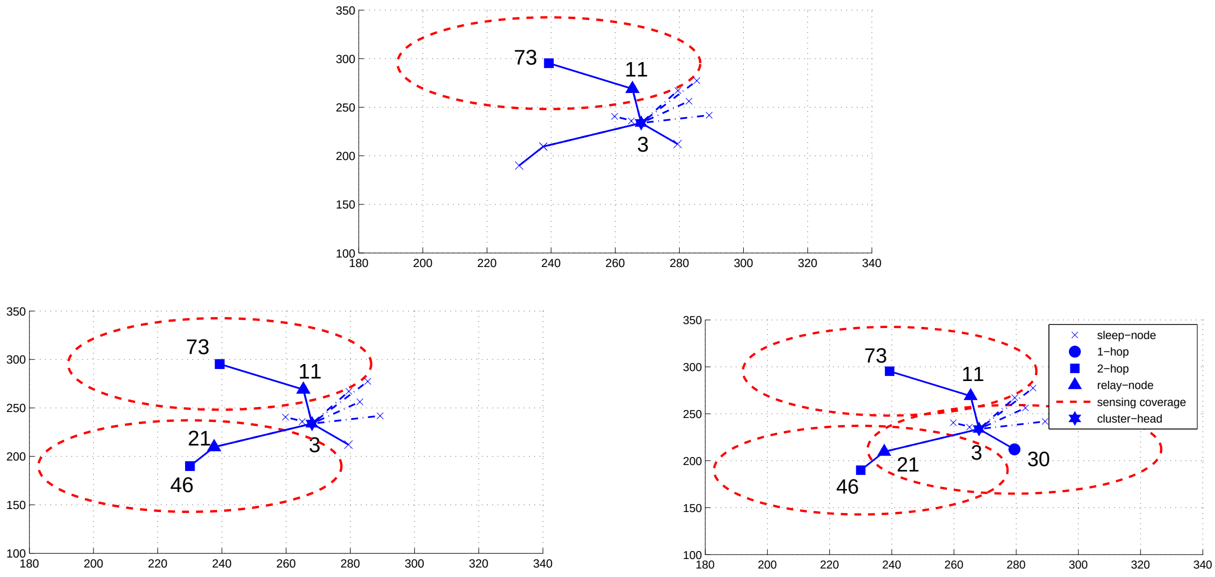
Assume that N_s sensors are uniformly distributed over a square region in two-dimensional space. Parameters for the random waiting timer, number of sensors, and ratio of transmitting range *R* to the side length *l* of the square, *R/l*, are investigated to provide a simulation-based study of the proposed schemes. for the DASA scheme, the parameters (detailed in Section 3.3) for the experiments are given by $C = 100$, $\lambda = 10$, $\alpha = 0.5$, and $\beta = 1.5$.

The first set of experiments illustrates two examples of generating a sensing round with the CASA approach and the DASA approach, respectively. According to the procedures of the CASA approach, as shown in Figure 6, the clusterhead selects a 2-hop cluster member and a relay node, say sensors 46 and 21, to initiate the round generation. For approximately minimizing the overlap of sensing coverage, sensors having no connectivity with sensor 46 may be selected as the round group member. By following the data gathering strategy in Section 3.2, sensors 46, 25, and 30 are chosen to form a sensing round.

In Figure 7, based on the settings of the DASA approach, a 2-hop cluster member with a shorter random waiting timer, say sensor 73, broadcasts a message to its neighbors and a 1-hop cluster member, say sensor 11, broadcasts a message to claim its being a relay node for sensor 73. When receiving the broadcasting messages, the neighboring sensors extend their waiting times to reduce the sensing area redundancy and further prepare for being the round group members in the following sensing rounds. Thus, with the data gathering strategy as described in Section 3.3, sensors 73, 46, and 30 form a sensing round. Observe that, as shown in Figures 6 and 7, the proposed approaches avoid heavy sensing redundancy and maintain sufficient sensing coverage.

Given a cluster topology, the second set of experiment studies the impact of parameter settings on

Figure 7. An example of generating a sensing round with the DASA approach.



network performance. With varying the number of sensors N_S, Figure 8 shows the number of sensing rounds N_{RG} versus α for $\beta = 1$ (left) and N_{RG} versus α for $\beta = 1.5$ (right). Given a value of the parameter β, the number of sensing rounds N_{RG} decreases with increasing value of the parameter α, which implies that the operation of updating the waiting time when receiving the round ID broadcast from the neighboring sensors may rearrange the sensing schedule and decrease N_{RG}.

Similarly, Figure 9 depicts the number of sensing rounds N_{RG} versus β with $\alpha = 0$ (left) and N_{RG} versus β with $\alpha = 0.5$ (right). Given $\alpha = 0$ (i.e. without applying the update formula), the number of sensing rounds N_{RG} increases slightly with increasing value of the parameter β. This is because the setting of the parameter β may allow nearby sensors to work on different rounds such that the sensing redundancy may be suppressed. On the other hand, by applying the update formula (e.g. $\alpha = 0.5$), the parameter α may be a dominant factor affecting the number of sensing rounds N_{RG} since this mechanism allows the sensors to observe the behaviors of their nearby sensors and make adjustments in their waiting times. Therefore, the update operation of the waiting time described in (4) may play a critical role in scheduling management.

Furthermore, in order to describe the interaction between the parameter settings and the network performance, Figures 10 and 11 illustrate the average coverage and average sensing overlap per round versus α with $\beta = 1.0$ (left) and with $\beta = 1.5$ (right), respectively. Observe that in Figure 10, given β the average coverage per round increases with increasing value of α. As shown in Figure 8, the number of sensing rounds N_{RG} decreases with increasing value of the parameter α, which means the number of active nodes in a round M_{RG} increases with increasing value of α and more sensing overlap may be introduced under this condition (Figure 11). Based on the above results, the parameters $\alpha = 0.5$ and $\beta = 1.5$ may be sensible settings for balancing the relationship between scheduling management and network performance. Therefore, depending on the requirement of the sensing task, these key parameters may be chosen to achieve desired performance.

Figure 8. The number of sensing rounds N_{RG} versus α with $\beta = 1$ (left) and N_{RG} versus α with $\beta = 1.5$ (right).

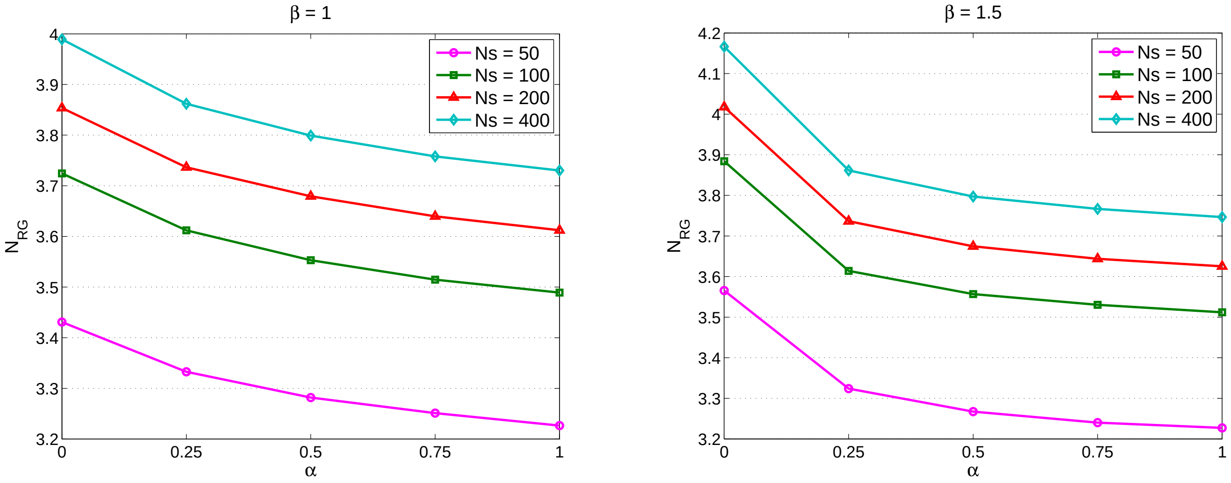
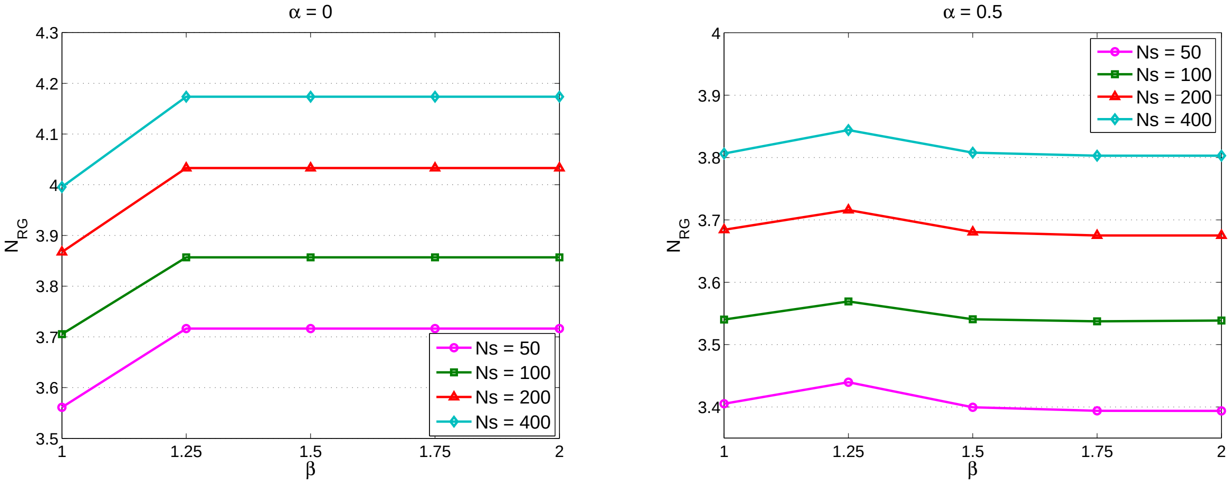


Figure 9. The number of sensing rounds N_{RG} versus β with $\alpha = 0$ (left) and N_{RG} versus β with $\alpha = 0.5$ (right).



The third set of experiments explores the performance of the neural network model. The efficiency of neural network training can be improved with certain preprocessing steps performing on the network inputs and targets [34]. Figure 12 illustrates the preprocessing results of the network inputs and targets, which transform inputs and targets into a better form and then reverse transformed outputs back to the characteristics of the original target data.

Figures 13 and 14 depict the learning and regression analysis of the network. Figure 13 shows that the network is learning since the mean squared error of the network is decreasing to a smaller value. The 6,514 input and target vectors are randomly divided into three sets. Four thousand vectors are used to train the network. Of these vectors 1,257 are used to validate how well the network generalized. Finally, the last 1,257 vectors provide an independent test of network generalization to data that the network has

Figure 10. Average coverage per round versus α with $\beta = 1.0$ (left) and with $\beta = 1.5$ (right).

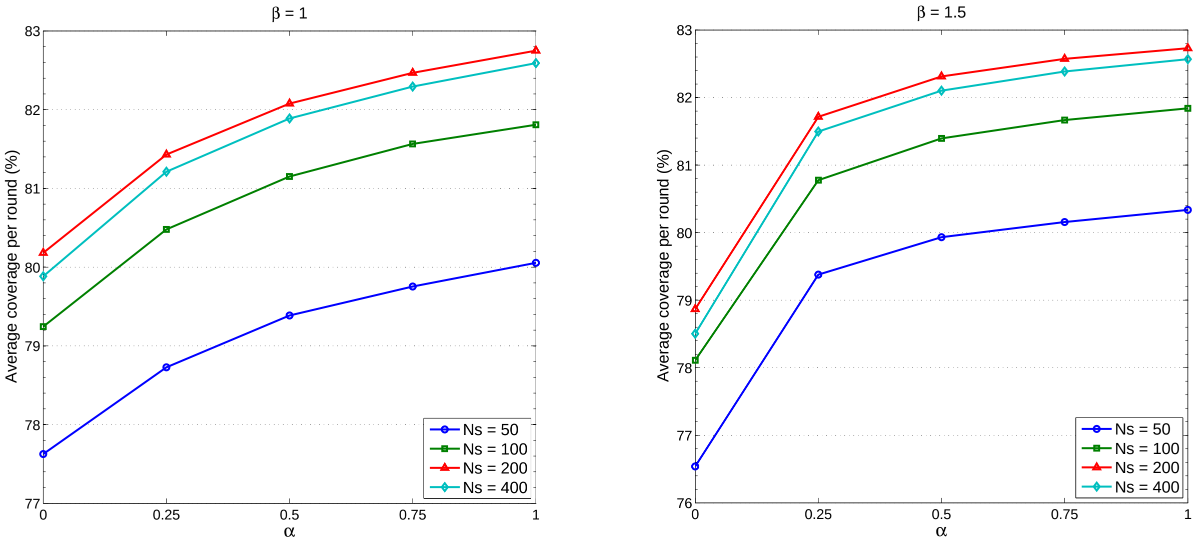
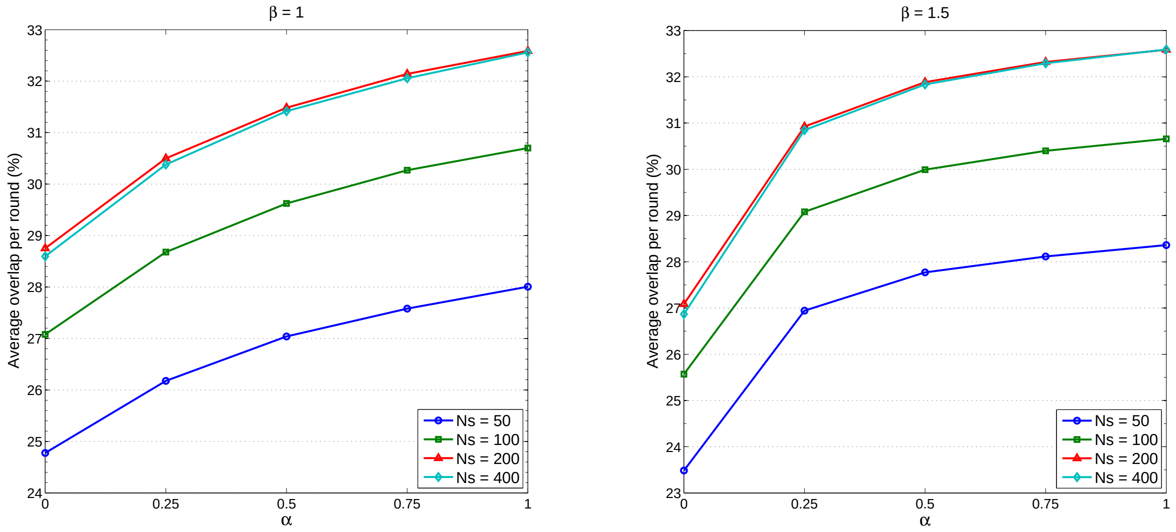


Figure 11. Average sensing overlap per round versus α with $\beta = 1.0$ (left) and with $\beta = 1.5$ (right).



never seen.

Moreover, regression analysis is employed as post-training analysis between the network response and the corresponding targets and three parameters are returned to evaluate the performance. The first two parameters, slope and y-intercept of the best linear regression relate targets to network outputs. If the outputs exactly equal to targets, the slope and y-intercept would be 1 and 0, respectively. For the 1-hop case, slope = 0.79 and y-intercept = $-6.1 \cdot 10^{-3}$. For the 2-hop case, slope = 1.0 and y-intercept = $2.7 \cdot 10^{-4}$. The third parameter is correlation coefficient between the outputs and targets. When the

Figure 12. The preprocessing results of the network inputs and targets.

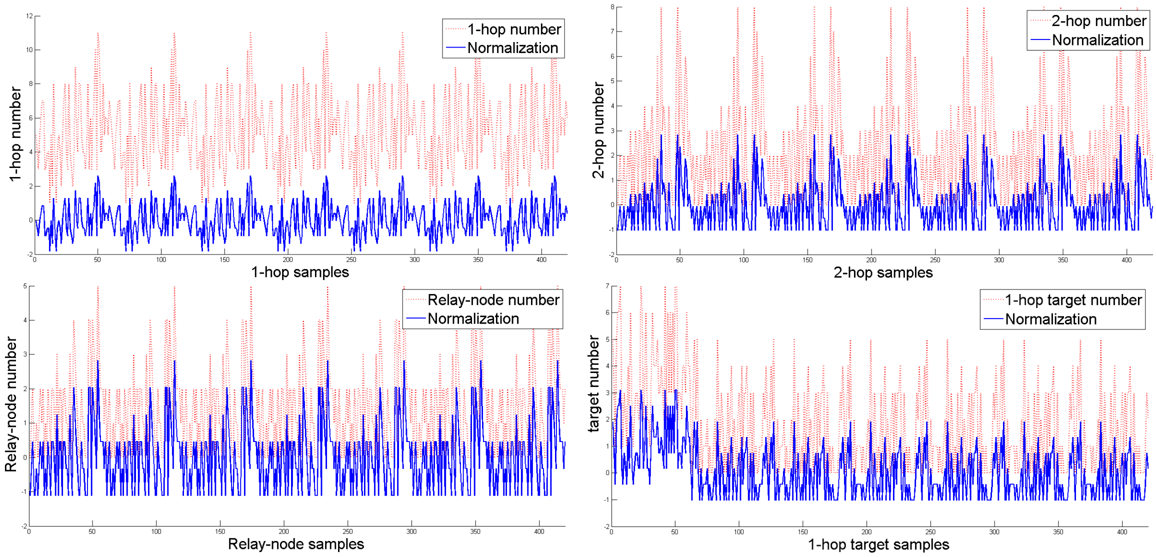
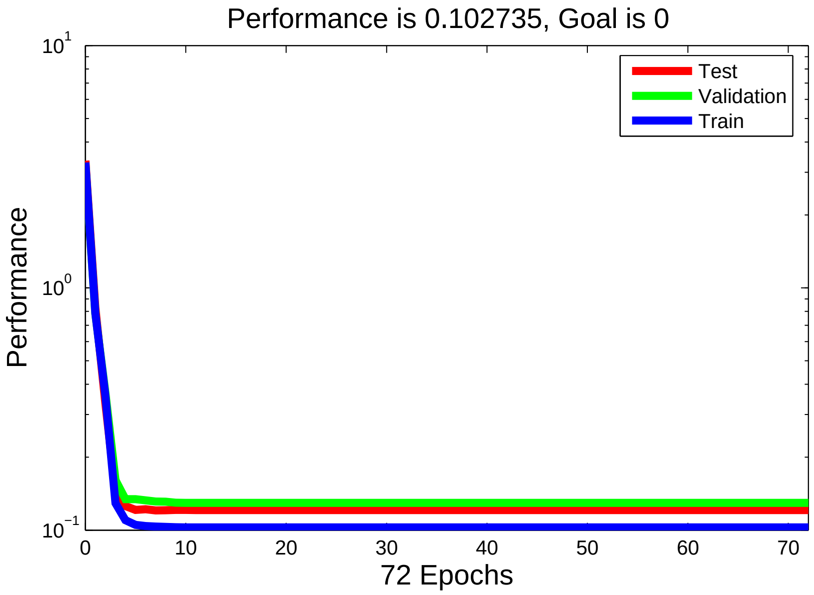


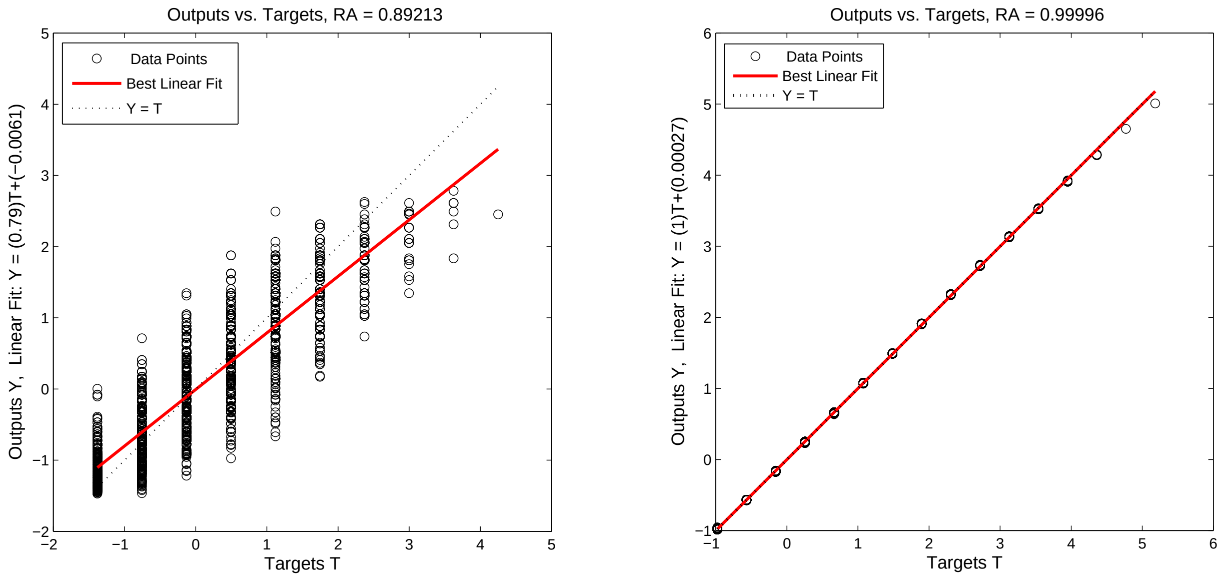
Figure 13. An independent test of network generalization.



correlation coefficient is equal to 1, then there is perfect correlation between targets and outputs. In this study, the correlation coefficients of 1-hop regression analysis and 2-hop regression analysis are $RA = 0.90$ and $RA = 1.00$ as shown in Figure 14 (left) and Figure 14 (right), respectively, which therefore illustrate a good fit.

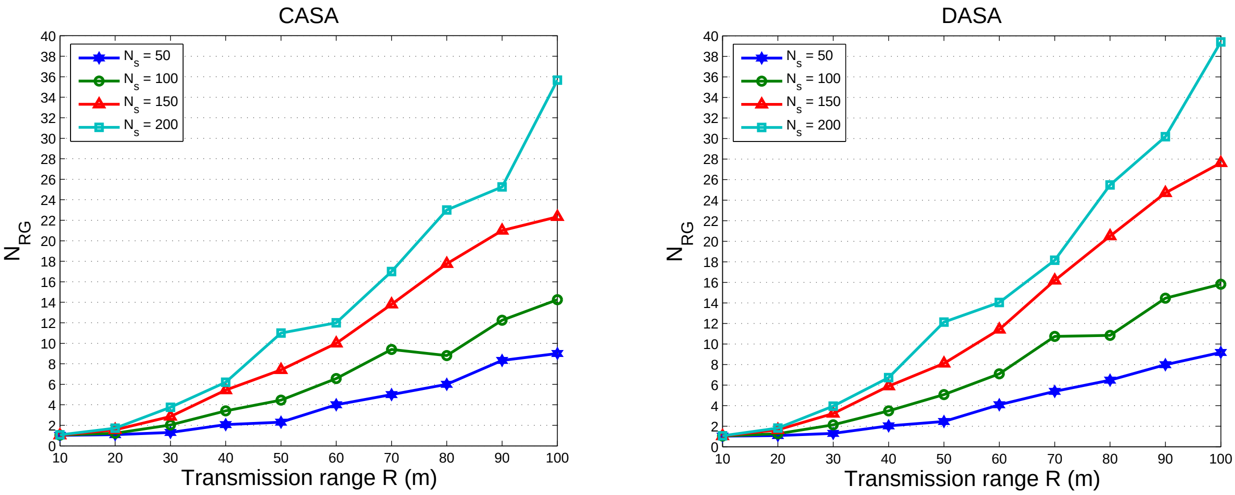
In order to simultaneously consider energy conservation, network connectivity, and the data gathering strategy, the fourth set of experiments investigates the impact of the transmission range *R* on the average number of round groups N_{RG} in the scheduling operation with $l = 300$ m. Figure 15 shows the relationship between the average number of round groups N_{RG} and the R/l ratio with varying the number of sensors. Figure 16 shows the comparison of the average number of round groups N_{RG} applying the

Figure 14. The regression analysis between the network response and the corresponding targets: 1-hop regression analysis (left) and 2-hop regression analysis (right).



proposed scheduling schemes and those using the analytical models. To further explore the sensing load in a round, Figure 17 depicts the number of active nodes M_{RG} in a round with varying the transmission range *R*. The result shows that the average number of group members in a round is between 1.5 and 2.5 for the proposed scheduling schemes.

Figure 15. The relationship between the number of sensing rounds N_{RG} in a cluster and transmission range *R* with varying the number of sensors.



Observe that, as shown in Figure 15, the average number of sensing rounds N_{RG} increases as the ratio R/l increases (i.e. the transmission power increases). Since larger transmission power allows

Figure 16. The comparison of the average number of round groups N_{RG} applying the proposed scheduling schemes and those using the analytical models.

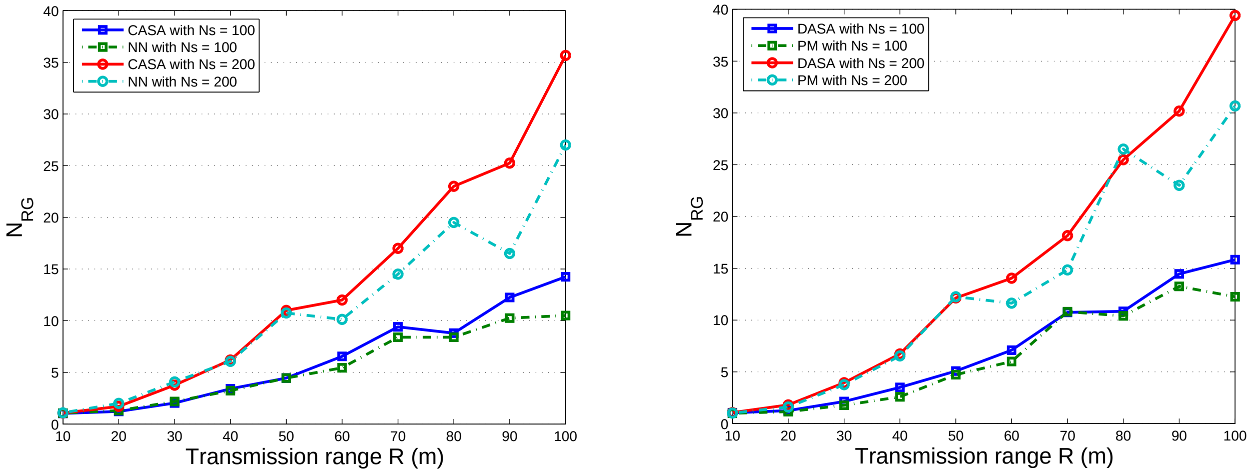
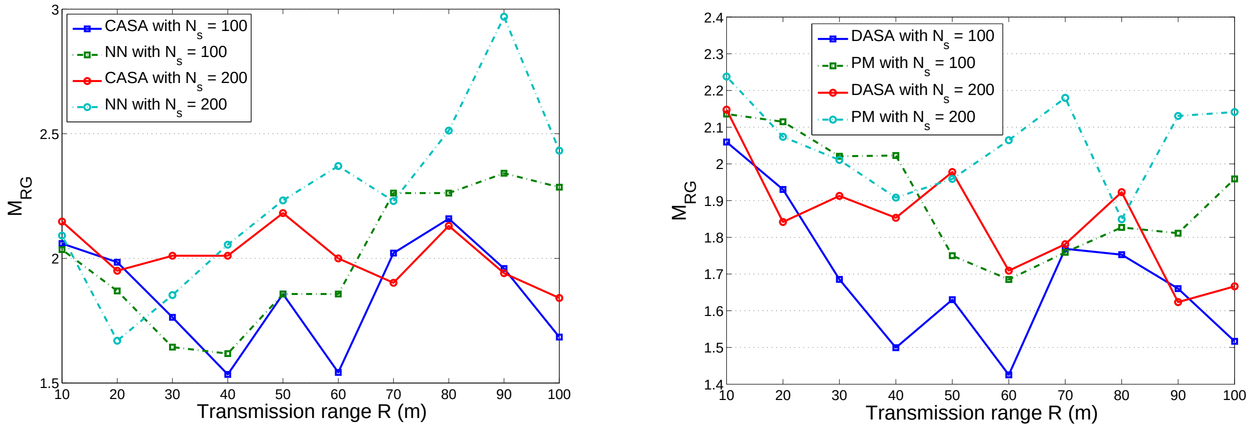


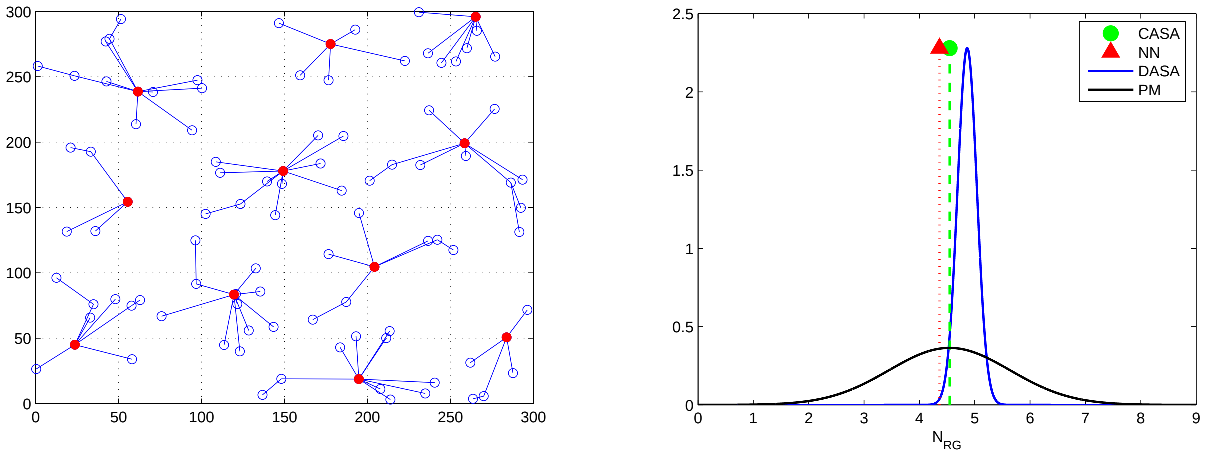
Figure 17. The number of active nodes in a round M_{RG} with varying the transmission range *R*.



larger radio coverage, a cluster has more cluster members, which increases the coverage overlap and results in a larger N_{RG} in a cluster. Thus, a large *R* may result in heavy sensing redundancy due to a large number of neighboring sensors. On the other hand, a small *R* may produce many isolated sensors in a network. Therefore, a sensible transmission range is essential to explore the performance of the scheduling approach. In [35], the authors suggest that $R \approx l\sqrt{\frac{\log l}{N_s}}$ may be a good choice for the initial range assignment for sensors in the 2-dimensional space, where N_s is the number of sensors. As a result, the clusterhead may adaptively manage the scheduling operation based on the data gathering strategy and an appropriate transmission range R in order to avoid severe communication interference.

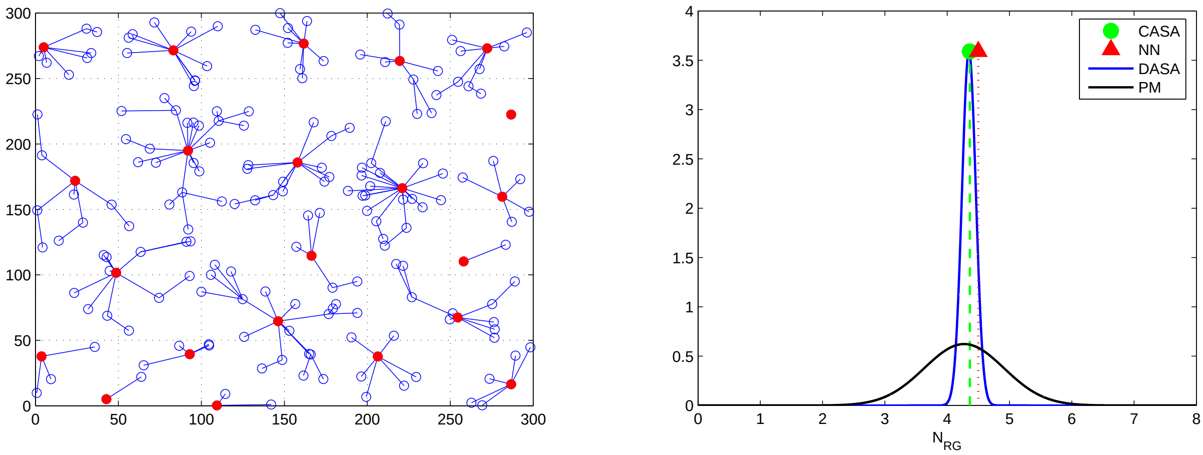
Accordingly, an appropriate transmission range R is applied when comparing the proposed schemes and the analytical models. Figures 18 and 19 show the standard deviation of the mean number of sensing rounds N_{RG} with $l = 300$ m, $N_s = 100$, and $R \approx 50$ m and with $l = 300$ m, $N_s = 200$, and $R \approx 33.387$ m, respectively. In Figure 18 (right), the neural network (NN) architecture well approximates the CASA performance since the NN retains global information from the training process. Because of the sensor spatial distribution in Figure 18 (left), for the DASA scheme, the result provides evidence that the Probabilistic Model (PM) provides a way to roughly predict the performance of the DASA. However, due to the uniform convergence of the sensor spatial distribution ($N_s = 200$) in Figure 19 (left), the PM method has done well to describe the performance of the DASA. Moreover, with an appropriate transmission range (with $l = 300$ m, $N_s = 100$, and $R \approx 50$ m and with $l = 300$ m, $N_s = 200$, and $R \approx 33.387$ m), Figures 18 and 19 show that both the proposed schemes and the analytical models are close, which coincides with the results in Figure 16. Therefore, the average number of round groups N_{RG} in a sensing cycle can be clearly specified for a sensing task in a cluster.

Figure 18. The distribution of the number of sensing rounds N_{RG} applying the proposed scheduling schemes and those using the analytical models with $N_s = 100$.



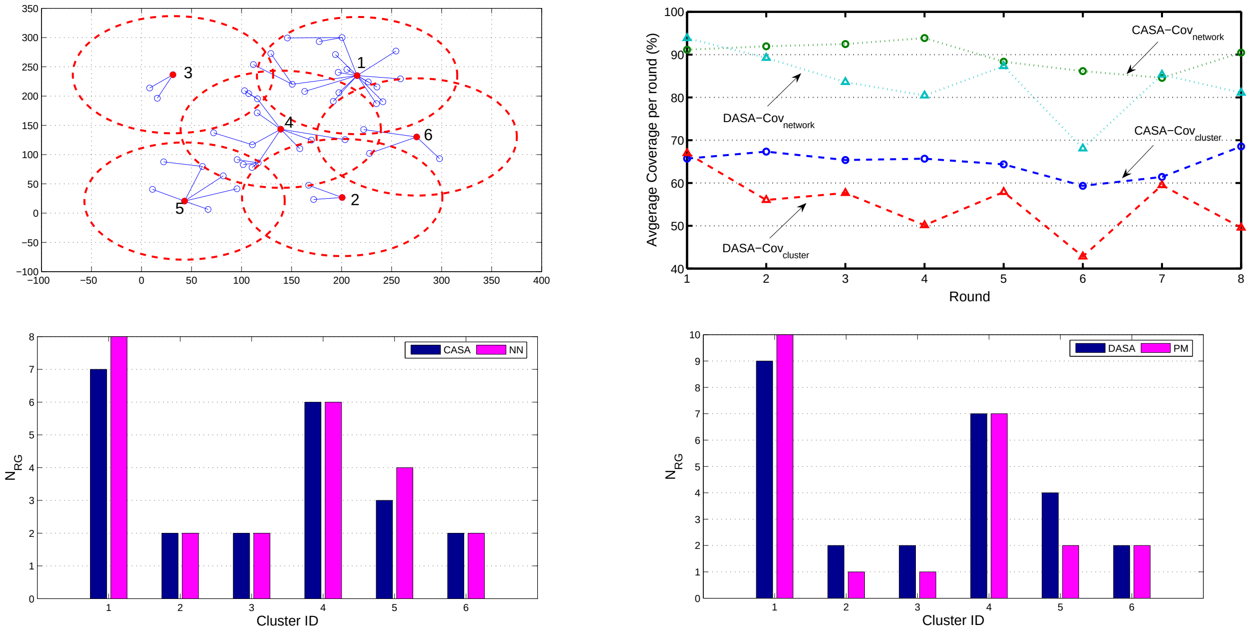
Note that Figures 15 and 16 investigate the average number of sensing rounds N_{RG} considering all clusters in a network. In order to further explore the performance of the proposed schemes and the analytical models, the fifth set of experiments examines the accuracy of the neural network architecture and the accuracy of the Probabilistic Model (PM) in each individual cluster. Given a cluster-based network topology (Figure 20 (top left)), the bottom-left and bottom-right quadrants in Figure 20 show that the accuracy of the network for new data and the approximation of the PM model match the performance of the proposed schemes well, which may provide clusterheads a way to estimate the number of sensing rounds given local topology information. Furthermore, the top-right quadrant in Figure 20 illustrates the coverage percentage of the whole network and each cluster, respectively, which suggests that the proposed schemes allow the network to obtain high coverage percentage with an appropriate sensing range. Notice that the operation of the CASA scheme achieves 65% average sensing coverage in a cluster and 90% average sensing coverage in a random network. This is attributed to the fact that

Figure 19. The distribution of the number of sensing rounds N_{RG} applying the proposed scheduling schemes and those using the analytical models with $N_s = 200$.



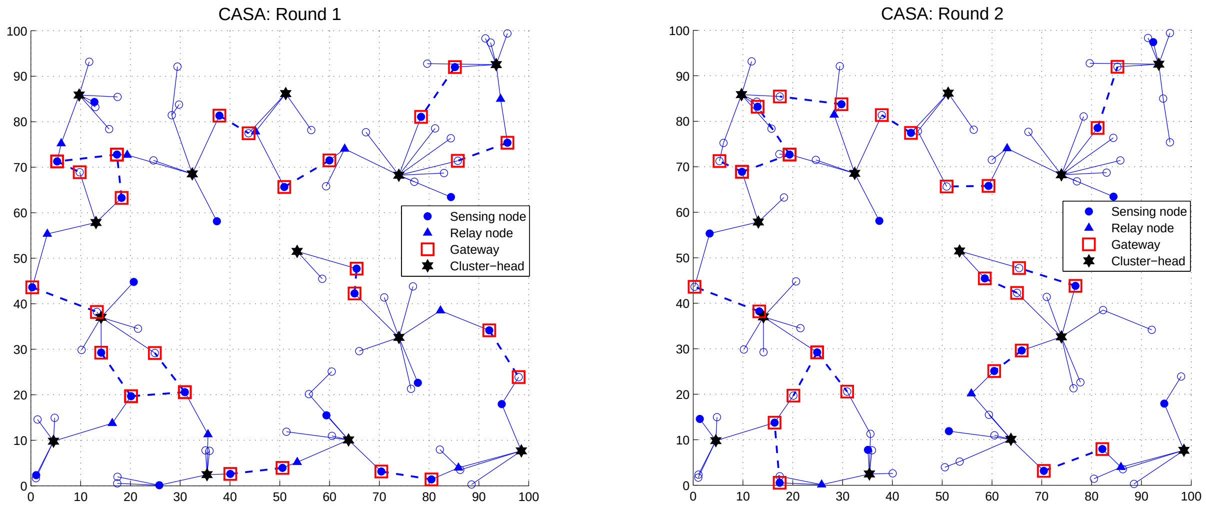
the sensing compensation from nearby clusters results in a higher percentage of sensing coverage in a network. Similarly, given the same network topology, the operation of the DASA scheme achieves 55% average sensing coverage in a cluster and 80% average sensing coverage in a random network. Since the CASA scheme executes the scheduling management in a centralized way, it may have better spatial arrangement of sensors in each round compared with the DASA scheme.

Figure 20. The coverage percentage of the whole network and each cluster (the top-right quadrant); the accuracy of the neural network architecture (the bottom-left quadrant) and the accuracy of the Probabilistic Model (PM) in a random network (the bottom-right quadrant).



The sixth set of experiments studies the network connectivity when using the proposed scheduling approaches. Given a random network with $N_s = 100$ and the period of the observation phase $5T_{cycle}$, Figures 21 and 22 show the maintenance of network connectivity in round 1 and round 2. The circle ('○') represents the sleep node and a connection between a pair of distributed gateways ('□') is indicated by a dashed line. Observe that in order to conserve energy, the active nodes (the sensing node '●' and the relay node '▲') are good candidates for being gateways in each round. As demonstrated in Figures 21 and 22, although the network may not be fully connected in each round, the operations of the proposed schemes maintain sufficient network connectivity and provide a way for inter-cluster communication and data dissemination.

Figure 21. The network connectivity using the CASA scheme in round 1 (left) and round 2 (right).



By following the analysis as detailed in Section 4.4, the seventh set of experiments illustrates the mean memory usage in the cluster members and the clusterhead, respectively. Assume each node has a 36-byte data packet to transmit. Figure 23 (right) depicts the mean memory utilization for gathering the information in a clusterhead. Figure 23 (left) shows the total mean memory usage for running the proposed scheduling algorithms in the sensors in a cluster. As shown in Figure 23 (left), compared with the CASA, the cluster members consume more memory as establishing the sensing schedule with the DASA scheme. The main reason can be contributed to the tracking operation of round ID, which is used to create the order of sensing round. Moreover, for the memory usage in a clusterhead [Figure 23 (right)], due to the centralized operation, the clusterhead using the CASA scheme consumes more memory, compared with the clusterhead using the DASA with only local information. However, when measuring the memory utilization in a sensing round, the memory usage performance of the proposed approaches are comparable since the number of active nodes in a round M_{RG} is considered to be close in each approach as demonstrated in Figure 17. Accordingly, the current wireless sensor networking products (e.g. Crossbow's IRIS Mote with 8K bytes RAM and 512K bytes Flash or Tmote Sky with 10K bytes RAM and 48K bytes Flash) are capable of running the proposed scheduling schemes.

Figure 22. The network connectivity using the DASA scheme in round 1 (left) and round 2 (right).

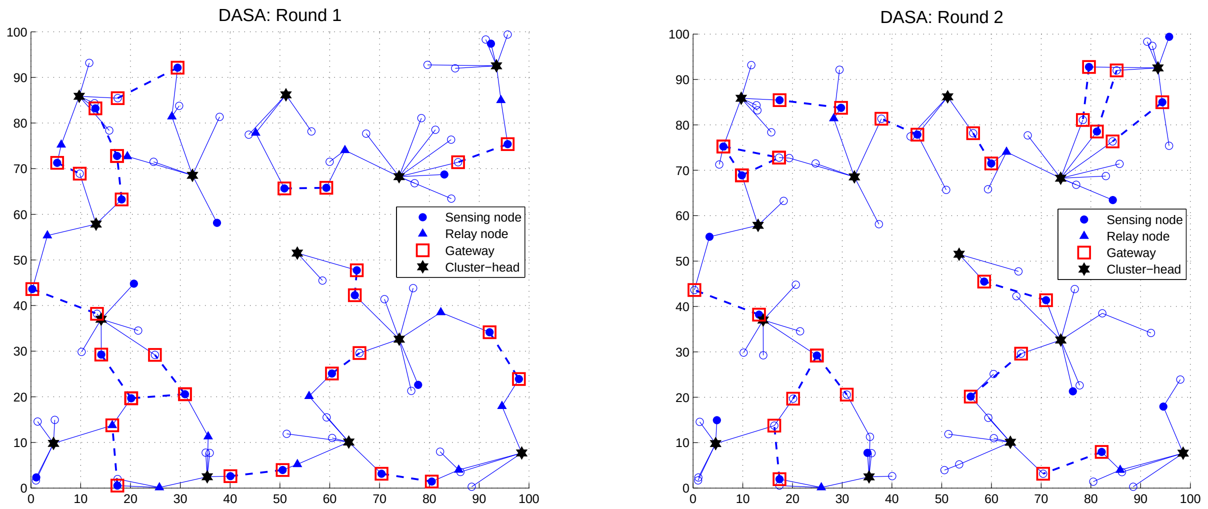
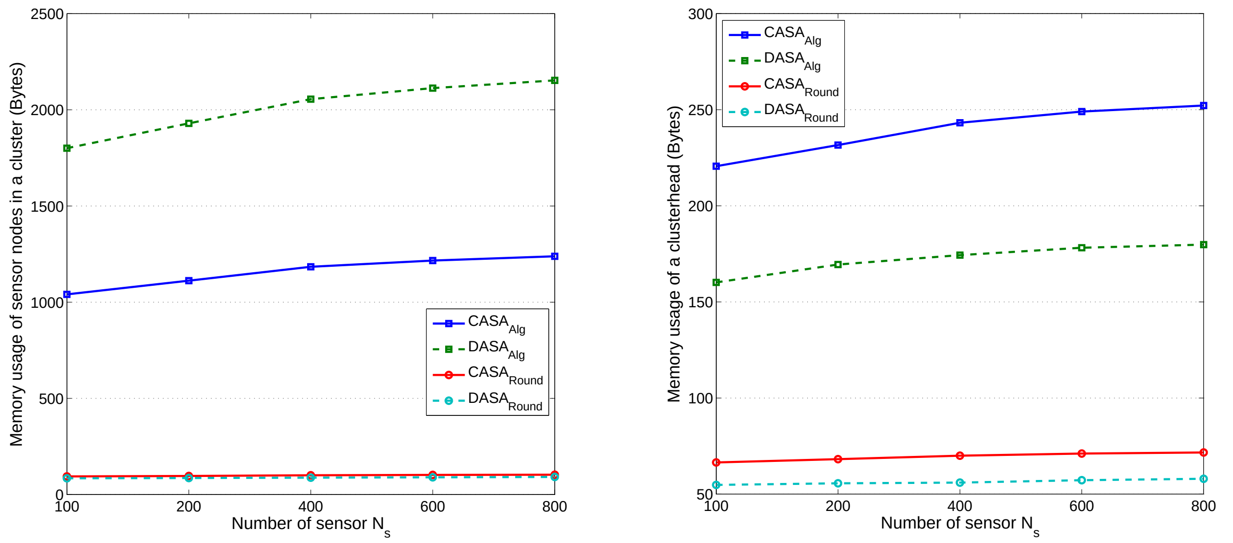
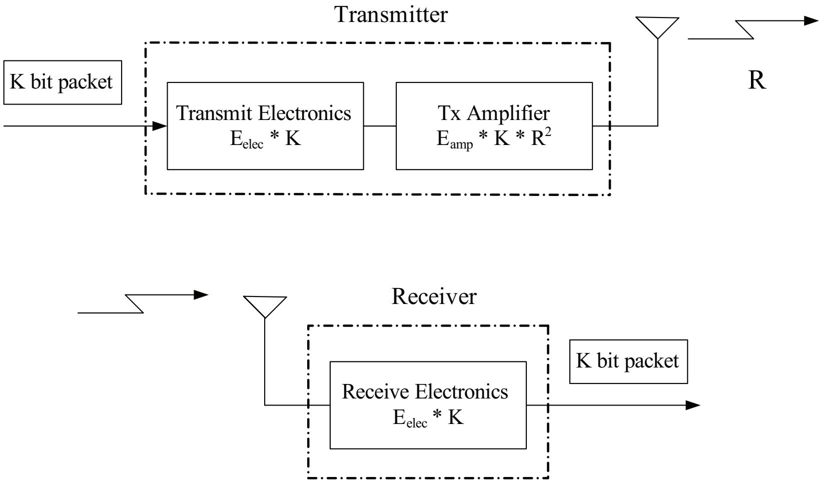


Figure 23. Memory usage of sensor nodes in a cluster versus number of sensors N_S (left); memory usage of a clusterhead versus number of sensors N_S (right).



The last set of experiments depicts the energy consumption of the proposed algorithms and compare the results with those of other cluster-based scheduling protocols. Assume that clusters are formed in a random network of 100 sensors with side length $l = 100$ m. A simple model [10] where the radio dissipates E_{elec} = 50 nJ/bit to run the transmitter or receiver circuitry and E_{amp} = 100 pJ/bit/m^2 for the transmit amplifier is applied in order to achieve an acceptable SNR. Suppose an r^2 energy loss due to channel transmission. Thus, to transmit a K-bit message a distance R using the radio model, the radio expends: $E_T = E_{elec} \cdot K + E_{amp} \cdot K \cdot R^2$ and to receive this message, the radio expends: $E_R = E_{elec} \cdot K$ (Figure 24). For data dissemination, the cluster-based hierarchical routing protocol [36] may be used for inter-cluster routing. The intra-cluster routing is built upon the node-level topology of cluster, which is obtained by the CAWT mechanism. Thus, data traffic between two clusters will be relayed through the gateway nodes.

Figure 24. First order radio model as described in [10].



Assuming that each node has a 36-byte data packet to transmit, Figure 25 (left) illustrates the average energy consumption per round under different transmission range with varying the number of sensors in the network. Furthermore, with varying the size of data packet in each sensor node, Figure 25 (right) presents the number of sensing rounds given the energy constraint. As expected, due to an increasing packet size and a higher energy dissipation rate, the number of sensing rounds decreases given a fixed initial energy (0.5 J). On the other hand, with an appropriate transmission ranges according to the network density (with $l = 100$ m, $N_s = 100$, and $R \approx 14.14$ m; with $l = 100$ m, $N_s = 200$, and $R \approx 10.0$ m; with $l = 100$ m, $N_s = 300$, and $R \approx 8.17$ m; with $l = 100$ m, $N_s = 400$, and $R \approx 7.07$ m) and varying the packet size, the number of sensing rounds in the network with different network densities suffers only small variations, which suggests that the proposed schemes may achieve high scalability for sensor scheduling.

Figure 26 shows the number of rounds when the first node dies in the network using the LEACH [10], the MECH [11], the CASA, and the DASA with varying the initial energy of each node from 0.25 J to 1.0 J. The LEACH and the MECH are clustering-based protocols that tries to minimize the energy dissipation in sensor networks. Observe that the proposed approaches are superior to the LEACH and the MECH approaches, while the number of sensing rounds grows nearly linearly as the initial energy increases. The simulation results demonstrate that the proposed schemes are more energy efficient than the LEACH and the MECH schemes.

6. Conclusions

This paper presents hierarchical scheduling algorithms, which use a local criteria to simultaneously undertake the sensing coverage and connectivity such that dynamic cluster-based sleep scheduling can

Figure 25. The relationship between energy consumption per round (Joules) and transmission range *R* (left); the number of sensing rounds with varying the size of data packet (right).

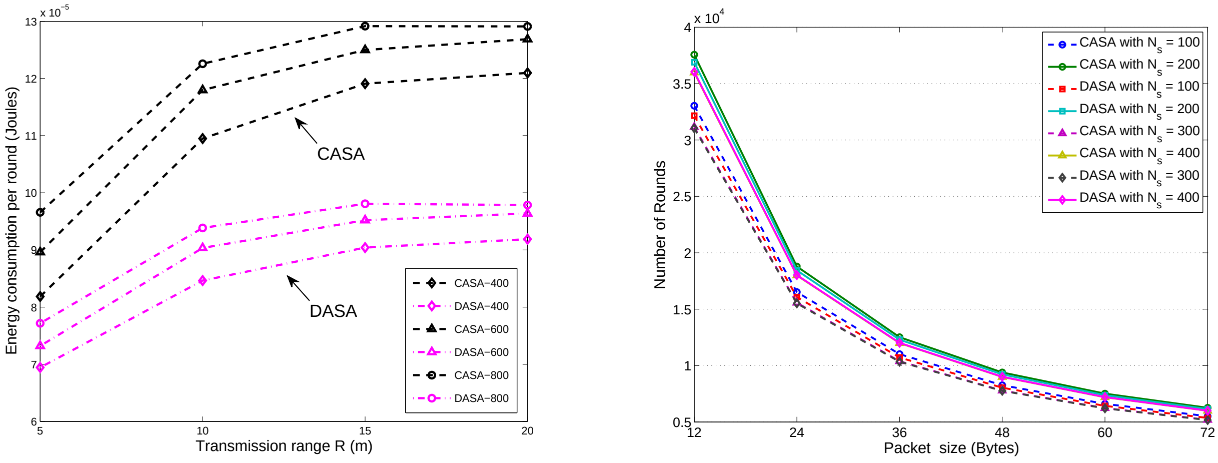
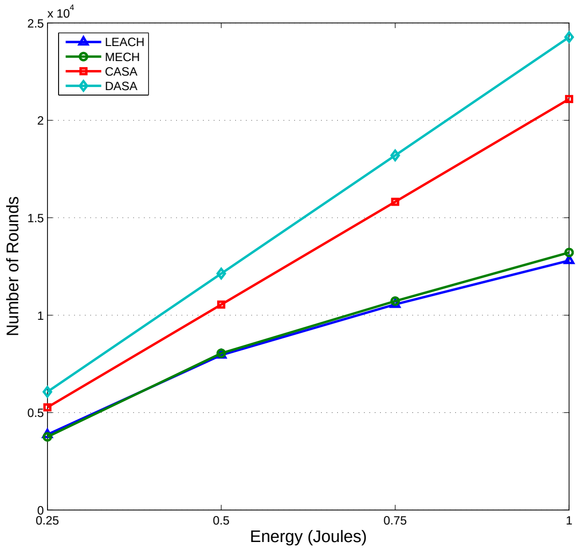


Figure 26. The comparison of the number of rounds as the first sensor node dies in the network using the LEACH, MECH (10 members), CASA, and DASA.



be achieved. An analytical network architecture and a probabilistic model are derived to describe the behaviors of the proposed schemes. The clusterheads may apply the established models to estimate the number of sensing rounds given local topology information. The main objective of the proposed dynamic sleep scheduling approaches is to extend the lifetime of the clusters so that the network may remain functional longer. The experimental results show that the proposed algorithms provide efficient network power control and achieve high scalability in wireless sensor networks.

There are several ways this work may be generalized. For instance, the CASA scheme may exploit the relationship between the monitored area in a cluster and the cluster topology to determine a proper number of group members in a round for the sensing task. Also, the DASA scheme can be generalized to a d-hop cluster-based network topology. By following the procedures of the DASA scheme, a d-hop cluster member may be a good candidate to initialize a round group and a nearby (d-1)-hop cluster member may choose to wait and join the following round group later. The random timer may be adjusted using local information and energy constraints and adapt based on the requirements of the sensing task in order to achieve network robustness and scalability.

In the proposed scheduling solutions, trade-offs are found between model complexity, energy consumption, computational complexity, and sensible model description in real systems. Future plans will involve generalizing the methods to design energy-efficient data dissemination protocols, to consider certain failure scenarios, to explore the sensitivity of the proposed schemes to data gathering strategies and network operation, and to perform actual measurements to investigate the impact of parameter settings on network performance.

References and Notes

1. Yan, T.; He, T.; Stankovic, J. Differentiated surveillance for sensor networks, In *Proc. of ACM Conference on Embedded Networked Sensor Systems*. Los Angeles, CA, USA, 2003; pp. 51-62.
2. Liu, B.; Towsley, D. A study on the coverage of large-scale sensor networks, In *Proc. of the First IEEE International Conf. Mobile Ad-Hoc and Sensor Systems*. Fort Lauderdale, FL, USA, 2004; pp. 475-483.
3. Ren, S.; Li, Q.; Wang, H.; Chen, X.; Zhang, X. Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 334–350.
4. Turau, V.; Weyer, C. Scheduling transmission of bulk data in sensor networks using a dynamic tdma protocol, In *Proc. of the International Workshop on Data Intensive Sensor Networks*. Mannheim, Germany, 2007; pp. 321-325.
5. Hohlt, B.; Doherty, L.; Brewer, E. Flexible power scheduling for sensor networks, In *Proc. of the 3rd International Symposium on Information Processing in Sensor Networks*. Berkeley, CA, USA, 2004; pp. 205-214.
6. Schrage, D.; Gonsalves, P. G. Sensor scheduling using ant colony optimization, In *Proc. of the 6th International Conference of Information Fusion*, Vol. 1. Cairns, Australia, 2003; pp. 379-385.
7. Decker, C.; Riedel, T.; Peev, E.; Beigl, M. Adaptation of on-line scheduling strategies for sensor network platforms, In *Proc. of the Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems*. Vancouver, Canada, 2006; pp. 534-537.
8. Chamberland, J.-F.; Veeravalli, V. V. The art of sleeping in wireless sensing systems, In *Proc. of the IEEE Workshop on Statistical Signal Processing*. St. Louis, Missouri, USA, 2003; pp. 17-20.
9. Tian, D.; Georganas, N. D. A node scheduling scheme for energy conservation in large wireless sensor networks. *Wirel. Commun. Mob. Comput.* **2003**, *3*, 271–290.
10. Heinzelman, W. R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks, In *Proc. of the 33rd Hawaii International Conference on System Sciences*. Hawaii, USA, 2000; pp. 1-10.
11. Chang, R.-S.; Kuo, C.-J. An energy efficient routing mechanism for wireless sensor networks, In

Proc. of the 20th International Conference on Advanced Information Networking and Applications. Vienna, Austria, 2006; pp. 308-312.

12. Cheng, C. T.; Tse, C. K.; Lau, F. C. M. A bio-inspired scheduling scheme for wireless sensor networks, In *Proc. of IEEE 67th Vehicular Technology Conference*. Singapore, Singapore, 2008; pp. 223-226.
13. Premkumar, K.; Kumar, A. Optimal sleep-wake scheduling for quickest intrusion detection using sensor networks, In *Proc. of IEEE INFOCOM*. Phoenix, AZ, USA, 2008; pp. 2074-2082.
14. Xiao, Y.; Zhang, Y.; Sun, X.; Chen, H. Asymptotic coverage and detection in randomized scheduling algorithm in wireless sensor networks, In *Proc. of IEEE ICC*. Glasgow, Scotland, 2007; pp. 3541-3545.
15. Xiao, Y.; Chen, H.; Zhang, Y.; Du, X.; Sun, B.; Wu, K. Intrusion objects with shapes under randomized scheduling algorithm in sensor networks, In *Proc. of the 28th International Conference on Distributed Computing Systems Workshops*. Beijing, China, 2008; pp. 315-320.
16. Abrams, Z.; Goel, A.; Plotkin, S. Set k-cover algorithms for energy efficient monitoring in wsns, In *Proc. of IPSN*. Berkeley, CA, USA, 2004, pp. 424-432.
17. Meguerdichian, S.; Koushanfar, F.; Potkonjak, M.; Srivastava, M. Coverage problems in wireless ad-hoc sensor networks, In *Proc. of IEEE INFOCOM*. Anchorage, Alaska, USA, 2001; pp. 1380-1387.
18. Wu, K.; Gao, Y.; Li, F.; Xiao, Y. Lightweight deploymentaware scheduling for wsns. *ACM/Springer Mobile Networks and Applications (MONET)* **2005**, *10*, 837–852.
19. Ye, F.; Zhong, G.; Cheng, J.; Lu, S.; Zhang, L. Peas: A robust energy conserving protocol for long-lived sensor networks, In *Proc. of ICNP*. Riverside, CA, USA, 2002; pp. 28-37.
20. Slijepcevic, S.; Potkonjak, M. Power efficient organization of wsns, In *Proc. of ICC*, Vol. 2. Helsinki, Finland, 2001, pp. 472-476.
21. Liu, C.; Wu, K.; Xiao, Y.; Sun, B. Random coverage with guaranteed connectivity: Joint scheduling for wsns. *IEEE Trans. Parallel Distrib. Syst.* **2006**, *17*, 562–575.
22. Gupta, V.; Chung, T. H.; Hassibi, B.; Murray, R. M. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica* **2006**, *42*, 251–260.
23. Shakkottai, S.; Srikant, R.; Shroff, N. Unreliable sensor grids: Coverage, connectivity and diameter, In *Proc. of INFOCOM*. San Francisco, CA, USA, 2003; pp. 1073-1080.
24. Zhang, H.; Hou, J. Maintaining coverage and connectivity in large sensor networks. *Ad Hoc Sens. Wirel. Netw.* **2004**, *1*, 89–123.
25. Choi, W.; Das, S. K. Coverage-adaptive random sensor scheduling for application-aware data gathering in wireless sensor networks. *Comput. Commun.* **2006**, *29*, 3467–3482.
26. Wang, L.; Xiao, Y. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mob. Netw. Appl.* **2006**, *11*, 723–740.
27. Wen, C.-Y.; Sethares, W. A. Automatic decentralized clustering for wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.* **2005**, *5*, 686–697.
28. Sun, K.; Ning, P.; Wang, C. Fault-tolerant cluster-wise clock synchronization for wireless sensor networks. *IEEE Trans. Dependable Secure Comput.* **2005**, *2*, 177–189.
29. Billingsley, P. *Probability and Measure*. John-Wiley & Sons, NY, USA, 1979.

30. Kumar, S. *Neural Networks: A Classroom approach*. McGraw-Hill: Singapore, 2005.
31. Garwood, F. The variance of the overlap of geometrical figures with reference to a bombing problem. *Biometrika* **1947**, *34*, 1–17.
32. Robbins, H. E. On the measure of a random set. i. *Ann. Math. Statist.* **1944**, *15*, 70–74.
33. Chatterjee, M.; Das, S. K.; Turgut, D. Wca: A weighted clustering algorithm for mobile ad hoc networks. *J. Cluster Comput.* **2002**, *5*, 193–204.
34. Demuth, H.; Beale, M.; Hagan, M. *Neural Network Toolbox 6: User's Guide*. The MathWorks, Inc., MA, USA, 2008.
35. Santi, P. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John-Wiley & Sons, Chichester, UK, 2005.
36. Joa-Ng, M.; Lu, I.-T. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *IEEE J. Sel. Areas Commun.* **1999**, *17*, 1415–1425.