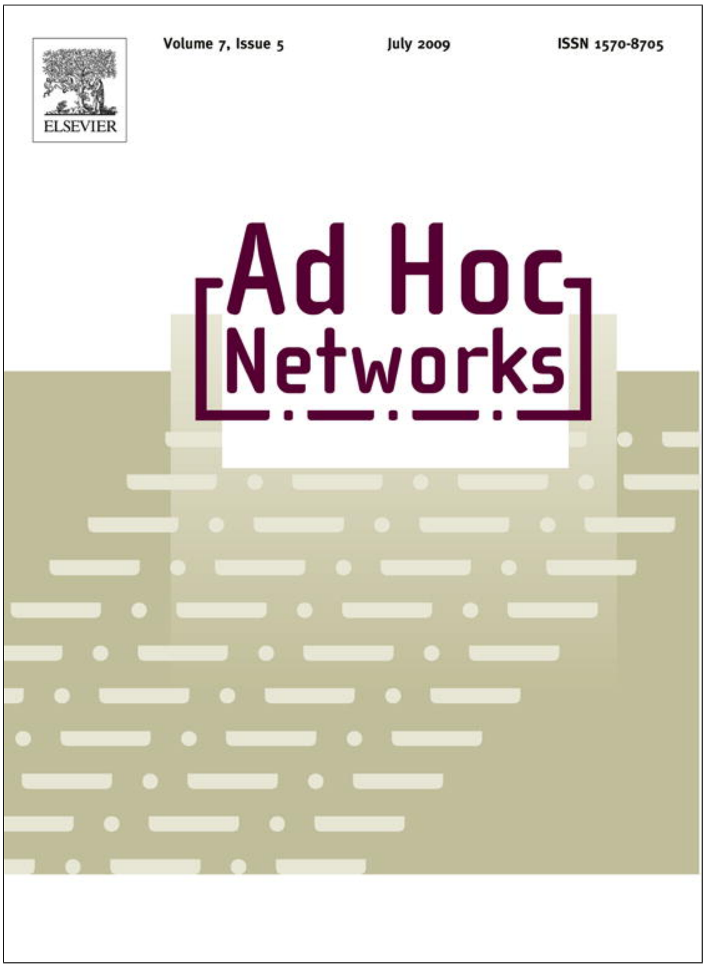

Volume 7, Issue 5
July 2009
ISSN 1570-8705
ELSEVIER
Ad Hoc
Networks

Ad Hoc Networks 7 (2009) 955–972

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Cluster head election techniques for coverage preservation in wireless sensor networks

Stanislava Soro *, Wendi B. Heinzelman

Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, United States

ARTICLE INFO

Article history:
Received 7 March 2008
Received in revised form 16 August 2008
Accepted 29 August 2008
Available online 11 September 2008

Keywords:
Wireless sensor networks
Clustering protocol
Coverage
Energy-efficiency
Cluster head selection

ABSTRACT

Coverage preservation is one of the basic QoS requirements of wireless sensor networks, yet this problem has not been sufficiently explored in the context of cluster-based sensor networks. Specifically, it is not known how to select the best candidates for the cluster head roles in applications that require *complete* coverage of the monitored area over long periods of time. In this paper, we take a unique look at the cluster head election problem, specifically concentrating on applications where the maintenance of full network coverage is the main requirement. Our approach for cluster-based network organization is based on a set of coverage-aware cost metrics that favor nodes deployed in densely populated network areas as better candidates for cluster head nodes, active sensor nodes and routers. Compared with using traditional energy-based selection methods, using coverage-aware selection of cluster head nodes, active sensor nodes and routers in a clustered sensor network increases the time during which full coverage of the monitored area can be maintained anywhere from 25% to 4.5×, depending on the application scenario.

1. Introduction

Organizing sensor networks into clustered architectures has been extensively explored over the last few years, leading to the appearance of a great number of task-specific clustering protocols. Clustering is one of the basic approaches for designing energy-efficient, robust and highly scalable distributed sensor networks. Utilizing clusters reduces the communication overhead, thereby decreasing the energy consumption and interference among the sensor nodes. In many applications, cluster organization is a natural way to group spatially close sensor nodes, in order to exploit the correlation and eliminate redundancy that often exists among the sensor readings. Through data fusion and aggregation of the sensors' data at the cluster centers, called cluster heads, the total amount of data sent to the sink can be significantly reduced, saving energy and bandwidth resources.

Oftentimes sensor networks must provide persistent coverage of the entire monitored area. Many applications require the ability to provide information from each part of the monitored area at any moment in order to meet the application's quality of service (QoS). As an example, in surveillance applications, the sensors must detect and track intruders, requiring that the entire monitored area be "covered" by the sensing region of the sensors. Furthermore, oftentimes sensors are deployed with much greater density than is needed to satisfy coverage requirements, which enables the redundantly covered nodes to conserve their energy by entering a low-power sleep mode.

While both cluster-based sensor network organization and coverage-maintenance protocols have been extensively studied in the past, these have not been integrated in a coherent manner. Existing techniques for the selection of cluster head nodes base this decision on various criteria, such as: maximum residual energy [5,7], location of the cluster head candidate relative to the other nodes [18], topology information [2,3,8], or previous activity of the sensor node as a cluster head [4]. Most of these cluster

* Corresponding author. Tel.: +1 585 242 9035.
E-mail address: soro@ece.rochester.edu (S. Soro).

1570-8705/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2008.08.006

head selection approaches are designed with the goal to provide balanced energy consumption among sensor nodes, but at the same time, these approaches do not consider the network's requirement for full coverage over extended periods of time. In other words, energy-balanced clustered network organization does not ensure that the wireless sensor network is able to *persistently* provide coverage of the *entire* monitored area. However, sensor coverage is one of the basic network QoS metrics, as it expresses the network's ability to provide constant monitoring/sensing of some area of interest [20,21]. Therefore, in this paper we explore the differences between energy-balanced and coverage-aware sensor network organization, specifically concentrating on clustered wireless sensor networks.

Intuitively, all sensor nodes do not equally contribute to network coverage. The loss of a sensor node deployed in a densely populated area is not as significant for network coverage compared to the loss of nodes from regions that are scarcely populated with sensor nodes. The importance of each sensor node to the coverage-preserving task can be quantitatively expressed by a *coverage-aware* cost metric, which is a metric originally introduced in [1]. This cost metric considers the node's remaining energy as well as the coverage redundancy of its sensing range, thereby measuring the contribution of this node to the network's coverage task.

In this paper we analyze how different coverage-aware cost metrics, some of which were defined in [1], can be utilized in the periodic election of cluster head nodes in homogeneous networks, ensuring that sensors that are more important to the network coverage task are less likely to be selected as cluster head nodes. Furthermore, the same coverage-aware cost metrics are used to find the set of active sensor nodes that provide full network coverage, as well as the set of routers that forward the cluster head nodes' data load to the sink. By comparing our approach with the HEED [7] clustering protocol we show the benefits of using this coverage-aware approach over traditional energy-based clustering in coverage-preserving applications. Our results indicate that clustering in sensor networks should be directed by two fundamental requirements – energy conservation and coverage preservation.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work on clustering and sensor management in wireless sensor networks. In Section 3, we introduce several cost metrics that are used to assign various roles to the sensor nodes. A new clustering algorithm for the election of cluster head, active sensor and router nodes is outlined in Section 4. Section 5 provides details on the simulation set-up, and Section 6 presents the simulation results comparing the different cost metrics for the selection of cluster head, active sensor and router nodes. In Section 7, we compare our coverage-preserving clustering approach with the HEED clustering protocol as one representative from a group of energy-aware clustering algorithms. In Section 8, we discuss the most relevant application scenarios where each cost metric can be used. Finally, in Section 9, we conclude this work and provide directions for our future work.

2. Related work

Work in the area of cluster-based wireless sensor networks is quite extensive, with energy efficiency and scalability being the main focus of many of the clustering protocols proposed so far. Similarly, much work has been done on sensor activation protocols, which focus on selecting a subset of the active sensor nodes that are sufficient to satisfy the network's coverage requirements, while allowing the remainder of the sensors to conserve their energy by entering the sleep mode. In this section, we discuss the related work that has been done in both these areas.

2.1. Clustering protocols

Probabilistic approaches for cluster head election, such as those proposed in [4,5], ensure that the cluster head role is shared equally among the nodes in the network, therefore prolonging the lifetime of the sensor nodes through balanced energy consumption. In [6] the authors found the optimal clustering parameters such as the probability of becoming a cluster head and the cluster radius for a network organized into single and multi-level clusters by minimizing the communication cost of the network. The HEED clustering protocol [7] uses a hybrid criterion for cluster head selection, which considers the residual energy of the node and a secondary parameter, such as the node's proximity to its neighbors or the node's neighbor degree. HEED prolongs the network lifetime by ensuring balanced energy dissipation as well as uniform distribution of cluster head nodes in network scenarios that contain uniformly dispersed sensor nodes. All of this work aims to extend network lifetime by balancing energy dissipation among the nodes. In our approach, we aim to keep alive the most critical sensors, thereby preserving coverage and extending network lifetime for coverage-based applications.

The works presented in [8,9] deal with the problem of power-balanced energy consumption among the cluster head nodes. In both papers the *coverage-time* is defined as the time until one of the cluster head nodes runs out of energy, leaving a hole the network's coverage. Shu et al. [8] stress the importance of simultaneous design of clustering strategies and routing, and they provide two mechanisms for balancing power consumption, called routing-aware optimal cluster planning and clustering-aware optimal random relay. However, in this work we are concerned with sensor networks where cluster head nodes are not deployed deterministically. Also, the definition of coverage-time in this work is stricter, corresponding to the time for which every part[1] of the monitored area is under the sensing range of at least one sensor node.

The partitioning of a network into uniformly dispersed clusters is the focus of the ACE clustering algorithm [10]. ACE provides uniform clusters by reducing the overlap among the clusters established in the initial phase. Those

[1] Due to the random deployment of sensor nodes, it may happen that some parts of the monitored area remain uncovered. Thus, "every part of the monitored area" corresponds to all parts which are under the sensing range of at least one sensor node at the beginning of the network lifetime.

nodes that have the largest number of either "uncovered" neighbors or neighbors in non-overlapping cluster areas are recruited as favorable new cluster head nodes. However, energy issues are not discussed in this paper.

Several papers deal with the design of clustering methods for the case of non-uniform deployment of sensor nodes. For example in [11], the authors consider the problem of power control and clustering in heterogeneous sensor networks. A clustered network structure is established to ensure that transmit power used by all nodes within the cluster converges to the same level. The authors in [12] notice that in non-uniformly deployed networks where the node density is globally high, the network can be partitioned into locally isotropic non-overlapping clusters with small density variations that will have high correlation in sensor measurements. While these techniques aim to create uniform or coherent clusters, they do not consider the coverage-preserving task required by many sensor network applications.

Clustering in the context of sensor management, topology control and routing was extensively investigated in the past. For example, the authors in [13] present the GAF routing protocol that controls the network topology and exploits node density to prolong the network lifetime. GAF uses geographical information to build a virtual grid across the network, and it selects only one sensor node to be active in each cell. The CEC [14] clustering protocol improves GAF by not relying on location information, further reducing redundant nodes by grouping them into clusters, where nodes with highest remaining energy are selected as cluster head nodes. Neither GAF nor CEC guarantee the complete coverage of the area covered by the network.

The problem of scheduling nodes to enter the sleep mode in cluster-based sensor networks was studied in [15]. The authors proposed a linear distance-based sleep scheduling scheme, where the probability that a sensor enters the sleeping state is proportional to its distance from the cluster head. Since such a scheme leads to unequal energy consumption of sensor nodes in the cluster, the same problem is further investigated in [16]. Here the authors present a balanced energy scheduling scheme, which accounts for the total energy spent in communication and sensing, thereby assuring that energy is uniformly spent by the nodes. Again, these approaches balance the nodes' energy consumption rather than reducing the energy consumption of the critical sensors, as is done in our approach.

An autonomous clustering algorithm based on coverage estimation self-pruning is presented in [17]. The sensor nodes with the largest expected coverage are the best candidates for the cluster head roles. The algorithm minimizes the clustering overhead compared to LEACH [4], and provides lower variation in the number of cluster head roles over time. However, this clustering scheme does not ensure full coverage of the network.

2.2. Active sensor selection protocols

Network coverage is one of the fundamental problems in sensor networks, since it affects the outcome of the network sensing task. Therefore, coverage together with sensor management has been a strong research focus for the last few years [19,21]. In [22] the authors provide a method for achieving full coverage of targets by dividing the sensors into disjoint cover sets that are active successively. In each cover set, a sufficient number of sensor nodes necessary to cover the targets is active, while the remainder of the nodes are put to sleep. However, their approach is based on a centralized solution. In PEAS [23] the nodes use a simple rule to decide about their activity. If a node cannot find another active node in its probing range, it becomes active; otherwise it returns to the sleeping mode. Although this approach eliminates the complexity of maintaining neighbor state and it does not require location information, it does not guarantee full sensing coverage of the network, which is our main concern.

In [24] the authors propose a scheduling scheme that enables each node to enter the active or sleeping mode based on the coverage information obtained from its neighbors, without compromising full network coverage. In order to avoid the "blind point" problem that occurs when two neighboring nodes simultaneously decide to turn off leaving a part of the area uncovered, the authors introduce a random back-off time before the node makes a decision about its status. In our clustering scheme the "blind problem" is solved by introducing delays in node activation based on the nodes' current cost values, thereby giving priority to low cost nodes to deactivate, anticipating the network's need for full coverage.

The problem of achieving full coverage in wireless sensor networks was also explored in [25]. The proposed algorithm (OGDC) tries to minimize the number of active nodes by reducing the overlapped area between the active sensors. To ensure that different nodes are active in each round, the starting node broadcasts a power-on message in a random direction along which working nodes are found. A node decides to turn off if it covers an intersection point between two active sensors and if it minimizes the overlapped area with active sensors. However, nodes do not consider the energy levels of their neighbors, so they can send the power-on messages in the direction of nodes with low remaining energy.

3. Family of coverage-aware cost metrics

The distributed activation with predetermined routes (DAPR) protocol proposed in [1] is the first routing protocol designed to avoid routing of data through areas sparsely covered by the sensor nodes. The idea behind this approach is that nodes in sparsely deployed areas, as well as nodes with small remaining energies are used less often as data routers, so that these nodes can collect data for longer periods of time. To accomplish this goal, the importance of every sensor node for the coverage-preserving task is quantified by a *coverage-aware* cost metric, which combines the information about the node's remaining energy with information about how redundantly this node's sensing area is covered by its neighboring nodes' sensing areas.

To explore the benefit of this approach in cluster-based sensor networks, we introduce several coverage-aware cost metrics. We assume that N_s sensor nodes from a set S, $s_i \in S$, $i = 1, \ldots, N_s$ are scattered randomly over a rectangular monitored area A. We assume the application requires that every part of the area be covered by the sensors throughout the network lifetime. Each sensor performs reliable sensing within its sensing area $C(s_i)$, which is approximated by a circular area around the node with radius R_{sense}. Note that this is a simple model for sensor coverage. Other techniques such as utilizing a learning phase where sensors learn their sensing area $C(s_i)$ based on training data can be used as well.

For every sensor node s_i we define a group of neighboring nodes $N(i)$ that includes all nodes with sensing areas either partially or fully overlapped with the sensing area of node s_i. Using our model for sensing area, we obtain

$$N(i) = \{s_j | d(s_i, s_j) <= 2 \cdot R_{sense}\}, \quad (1)$$

where $d(s_i, s_j)$ is the Euclidean distance between nodes s_i and s_j.

To reduce the number of active nodes while ensuring that every point (x, y) of the monitored region is covered by at least one sensor, each node needs to determine the overlap of its sensing area with the sensing areas of its neighboring nodes. For this, we assume that sensor nodes have localization capabilities. Considering each node's position and its residual energy, for each point (x, y) of the monitored area A we define the total energy $E_{total}(x, y)$ that is available for monitoring that location

$$E_{total}(x, y) = \sum_{s_j:(x,y) \in C(s_j)} E(s_j), \quad (2)$$

where $E(s_j)$ is the remaining energy of node s_j.

The first two cost metrics presented below, and defined in [1], are based on the total energy $E_{total}(x, y)$ available for monitoring each location in the sensor field.

3.1. Minimum-weight coverage cost

The minimum-weight coverage cost is defined as

$$C_{mw}(s_i) = \max \frac{1}{E_{total}(x, y)} \quad (x, y) \in C(s_i). \quad (3)$$

This cost metric measures node s_i's importance for the network coverage task by considering the energy of the most critically covered location (x, y) within the sensing area of the node.

3.2. Weighted sum coverage cost

The weighted-sum coverage cost is defined as

$$C_{ws}(s_i) = \int_{C(s_i)} \frac{dxdy}{E_{total}(x, y)} = \int_{C(s_i)} \frac{dxdy}{\sum_{s_j:(x,y) \in C(s_j)} E(s_j)}. \quad (4)$$

This cost metric measures the weighted average of the total energies of all points that are covered by the sensing area of node s_i.

3.3. Coverage redundancy cost

The coverage redundancy cost metric does not depend on a node's remaining energy nor on the remaining energies of its neighbors. Instead, this cost considers only the coverage redundancy of the overlapped sensing areas between the sensor and its neighboring nodes. Similarly to the previously defined $E_{total}(x, y)$, we define a total coverage $O_{total}(x, y)$, which provides the number of nodes that cover each point (x, y) of the area A

$$O_{total}(x, y) = \sum_{s_j:(x,y) \in C(s_j)} 1. \quad (5)$$

Then, the coverage redundancy cost of sensor s_i is

$$C_{cc}(s_i) = \int_{C(s_i)} \frac{dxdy}{O_{total}(x, y)} = \int_{C(s_i)} \frac{dxdy}{\sum_{s_j:(x,y) \in C(s_j)} 1}. \quad (6)$$

Fig. 1 provides an example that illustrates the minimum-weight, weighted-sum and coverage redundancy cost metrics. This example considers three nodes s_1, s_2 and s_3 with remaining energies $E(s_1)$, $E(s_2)$ and $E(s_3)$. The parameters A_i, $A_{i,j}$ and A_{ijk}, $i, j, k \in \{1, 2, 3\}$ are the areas of overlapped portions of the nodes' sensing areas.

3.4. Energy-aware cost

The energy-aware cost function evaluates the sensor's ability to take part in the sensing task based solely on its remaining energy $E(s_i)$

$$C_{ea}(s_i) = \frac{1}{E(s_i)}. \quad (7)$$

3.5. Coverage-aware routing cost

The cost metrics introduced in the previous subsections are the basis for *coverage-aware routing*, where the mini-

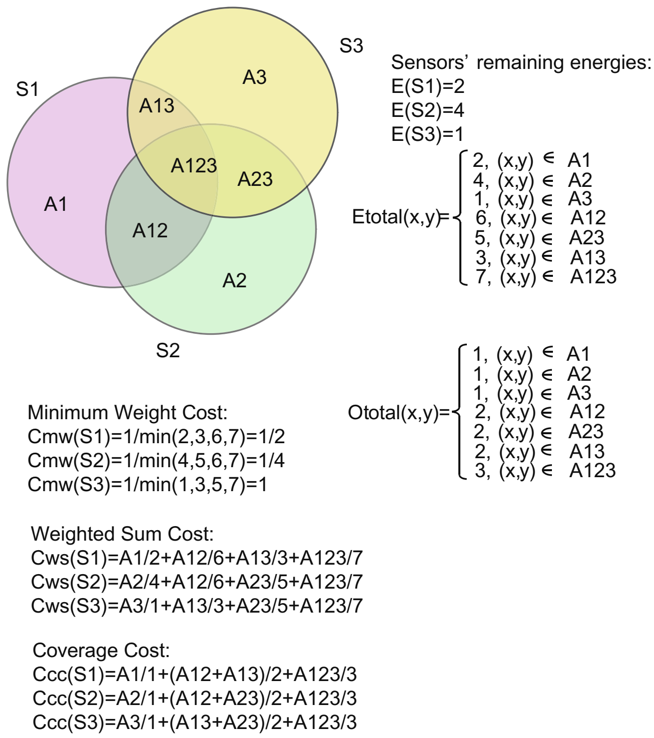


Fig. 1. Illustration of the coverage-aware cost metrics.

mum cost routing paths are determined such that high cost nodes are excluded from the routing task. The cost of a link between two nodes s_i and s_j is equal to the energy spent by these nodes to transmit $(E_{tx}(s_i, s_j))$ and to receive $(E_{rx}(s_i, s_j))$ one data packet, weighted by the costs of these nodes

$$C_{link}(s_i, s_j) = C_{aa}(s_i) \cdot E_{tx}(s_i, s_j) + C_{aa}(s_j) \cdot E_{rx}(s_i, s_j), \quad (8)$$

where C_{aa} represents any of the cost metrics described above. Therefore, the minimum cumulative cost path from each node to the sink is found as

$$C_{final}(s_i) = \sum_{s_j, s_k \in p(s_i, S_{dst})} C_{link}(s_j, s_k), \quad (9)$$

where p is the minimum cost path from node s_i to the sink S_{dst}. The cost defined by Eq. (9) is called the coverage-aware routing cost.

Data routing from every cluster head to the sink is done over multi-hop paths, which are found by minimizing C_{final} in Eq. (9). More details about routing from the cluster head nodes to the data sink are provided in Section 4.

4. Coverage-preserving clustering protocol (CPCP)

To ensure balanced energy consumption among the cluster head nodes throughout the network lifetime, many clustering protocols favor uniformly distributed clusters with stable average cluster sizes. However, obtaining the same number of well distributed clusters over time is a real challenge in clustered sensor networks.

In coverage-based applications, the best candidates for cluster head roles should be the redundantly covered nodes in densely populated areas with high remaining energy. These nodes can support clusters with a large number of members. While the excessive energy consumption of the cluster head nodes makes these nodes die before the other nodes, their death should not affect the overall network coverage since these nodes are located in densely populated areas. By our approach, which considers the application's requirements for full network coverage, the set of cluster head nodes can be selected based on the cost metrics defined in Section 3. However, cluster head selection based solely on any of the proposed cost metrics using existing clustering techniques will lead to an undesirable situation: the densely populated parts of the network will be overcrowded with cluster head nodes, while the scarcely covered areas will be left without any cluster head nodes. In such a situation, it is likely that the high cost sensors from poorly covered areas will have to perform expensive data transmissions to distant cluster head nodes, further reducing their lifetime.

In order to avoid this situation, we propose the clustering method called coverage-preserving clustering protocol (CPCP). CPCP spreads cluster head nodes more uniformly throughout the network by limiting the maximum cluster area. Thus, clusters in sparsely covered areas are formed as well as clusters in densely covered areas, which prevents the high cost nodes from having to perform costly packet transmissions to distant cluster head nodes. Also, nodes from the sparsely covered areas elected to serve as cluster head nodes support clusters with a smaller number of nodes compared to cluster head nodes in dense areas.

We define the cluster radius $R_{cluster}$ as a tunable parameter that determines the minimum distance between any two cluster head nodes in the network. Using this parameter, CPCP prevents the appearance of non-uniformly distributed clusters within the network. $R_{cluster}$ can be easily tuned by changing the transmission power of the cluster head nodes.

In CPCP the sensor nodes communicate directly with their elected cluster head nodes, while data routing from the cluster head nodes to the sink is done over multi-hop paths using the sensors. CPCP consists of six phases: information update, cluster head election, route update, cluster formation, sensor activation and data communication, as described below.

4.1. Phase I: information update

The first phase of CPCP is reserved for updating information on the remaining energies of the nodes. Each sensor node broadcasts an update packet with information about its remaining energy to all its neighbors in the range $2 \cdot R_{sense}$. In order to reduce packet collisions, the nodes use random back-offs before sending the update packets. Upon receiving the update information from all neighbors, each node calculates its coverage-aware cost (C_{mv} or C_{ws}), as described in Section 3.[2] Assuming that the sensor nodes are static, the neighboring nodes must exchange their location information only once, at the beginning of the network lifetime.

If the coverage redundancy cost C_{cc} or the energy-aware cost C_{ea} are used, then this Information Update phase can be skipped, since these cost metrics do not depend on the neighboring nodes' remaining energies.

4.2. Phase II: cluster head election

At the beginning of this phase every sensor determines its "activation time" – an amount of time proportional to its current cost (i.e., C_{mw}, C_{ws}, C_{cc} or C_{ea}). Each sensor has to wait for the expiration of its "activation time" before deciding weather or not it should announce itself as a new cluster head for the upcoming communication round. If during the "activation time" a node does not hear an announcement message from any other sensor node, then, upon expiration of its "activation time" it declares itself to be a new cluster head, by sending an announcement message to all the nodes within the $R_{cluster}$ range. The announcement message contains information about the new cluster head's location.

After receiving an announcement message from a new cluster head node, all sensor nodes in $R_{cluster}$ range exclude themselves from further consideration for the cluster head role. Each sensor node maintains a table of all cluster head nodes from which it has received the announcement message so far, as well as the distance to each cluster head

[2] In our implementation, we assume that each sensor node creates a grid of equidistant points under its sensing range, and it checks whether each point is under the sensing range of other neighboring nodes. Since the nodes are static, this checking needs to be done only once, at the beginning of the network lifetime.

node. This information is used later by the sensor node to decide about its cluster membership. Rarely it may happen that two sensor nodes with the same costs and within each other's R_{cluster} range simultaneously declare themselves to be new cluster head nodes – this conflict can be solved by giving priority to the node with the higher remaining energy.

Algorithm 1

The cluster head election and cluster formation phases of CPCP

```
S = {s | E(s) > 0}, E(s) – residual energy of sensor
node s
S_CH = {}
T_ch(i) = {}, i = 1, ..., N
while S ∉ {} do
   (s_k – node with minimum cost) and (E(s_k) > E_th)
   S_CH = S_CH ∪ s_k
   N(k) = {s | dist(s, s_k) < R_cluster}
   ∀s ∈ N(k), T_ch(s) = T_ch(s) ∪ s_k
   S = S \ N(k)
end while
∀s | S_CH ∩ s = {∅}
s sends JOIN message to cluster head s_CH for which
dist(s, s_CH) = min(dist(s, s_i)),   ∀s_i ∈ T_ch(s)
S_un = {s | (S_ch(s) = {∅}) & (S_CH ∪ s = {∅})}
if S_un ∉ {} then
   ∀s ∈ S_un, find
s_n | dist(s, s_n) = min(dist(s, s_i)),   ∀s_i ∈ N(s)
   s sends data packet to s_n
end if
```

When cost metrics C_{mw}, C_{ws} or C_{cc} are used, it can happen that a sensor node with low remaining energy is elected to serve as a cluster head. This may cause the loss of the cluster's data during the communication round. This outcome can be avoided by preventing those sensor nodes that have remaining energy below a certain threshold E_{th} from taking part in the cluster head election process. If, after the cluster head election phase, these sensor nodes do not belong to any of the elected cluster head nodes, they find the nearest sensor node to which they forward their data. The pseudo code for the cluster head election phase of CPCP is provided in Algorithm 1.

4.3. Phase III: route update

The cluster head nodes send their data over multi-hop paths to the sink. To obtain these routes, the sink node first generates a *route discovery* message that is broadcasted throughout the network. Upon receiving the broadcast message, each sensor node introduces a delay proportional to its cost before it forwards the *route discovery* message to nodes in range R_{bc}. In this way a message arrives at each node along the desired minimum cost path. The cumulative cost of the routing path from the sink to the node obtained in this phase is called the *coverage-aware routing cost* of the node, as described in Eq. (9).

4.4. Phase IV: cluster formation

In the fourth phase of CPCP, each non-cluster head node decides to join the closest cluster head node. The sensor nodes send short *JOIN* messages to their selected cluster head nodes. These *JOIN* messages serve as an acknowledgement that a node will become a member of the cluster for the upcoming round. Note that there is no restriction on the number of cluster members, only on cluster area, and thus any node that wants to join a cluster head, and is within the cluster range R_{cluster}, can do so. In this way, Voronoi-shaped clusters are formed around cluster head nodes, as shown in Fig. 2.

4.5. Phase V: sensor activation

In the fifth phase, a subset of sensor nodes is selected to perform the sensing task for the upcoming round, while the rest of the sensor nodes go to sleep. The selected active nodes provide full coverage over the monitored field during this communication round.

In the sensor activation phase, which is shown in Algorithm 2, each sensor node assigns itself an activation delay that is proportional to its current application cost (i.e., C_{mw}, C_{ws}, C_{cc} or C_{ea}). In this way, sensor nodes with smaller cost have a better chance of becoming active sensors in the upcoming communication round. Each sensor node then waits for this period of time before deciding whether it will stay awake during the next communication round. While waiting for its activation delay to expire, the sensor node can receive the ACTIVATION messages from its neighboring nodes, which have smaller activation delays (smaller cost), if they decide to become active during the upcoming communication round. If, after its activation delay time expires, the sensor node determines that its sensing area is completely covered by its neighboring nodes, it turns itself off for the upcoming round. Otherwise, the sensor node broadcasts an ACTIVATION message to inform its neighbors about its decision to remain active. In this way, the lower cost nodes have priority to decide whether they should be active. All sensor nodes in the network jointly take part in the activation phase, regardless of the cluster to which they belong. This eliminates the redundant activation of sensor nodes on the borders of the clusters, which may happen when the activation of nodes is done in each cluster independently.

Algorithm 2

The sensor activation phase of CPCP

```
S = {s | E(s) > 0, s ∉ S_CH}, E(s) – residual energy of node
s
T_a(s) ∝ c(s), c(s) ∈ {C_ea, C_mw, C_ws, C_cc}
During period T_a(s):
s can receive ACTIVATION messages from its
neighboring nodes
T_a(s) expired:
s checks whether its sensing range is fully covered
if sensing range of node s is not fully covered then
   send ACTIVATION message to its neighboring nodes
end if
```

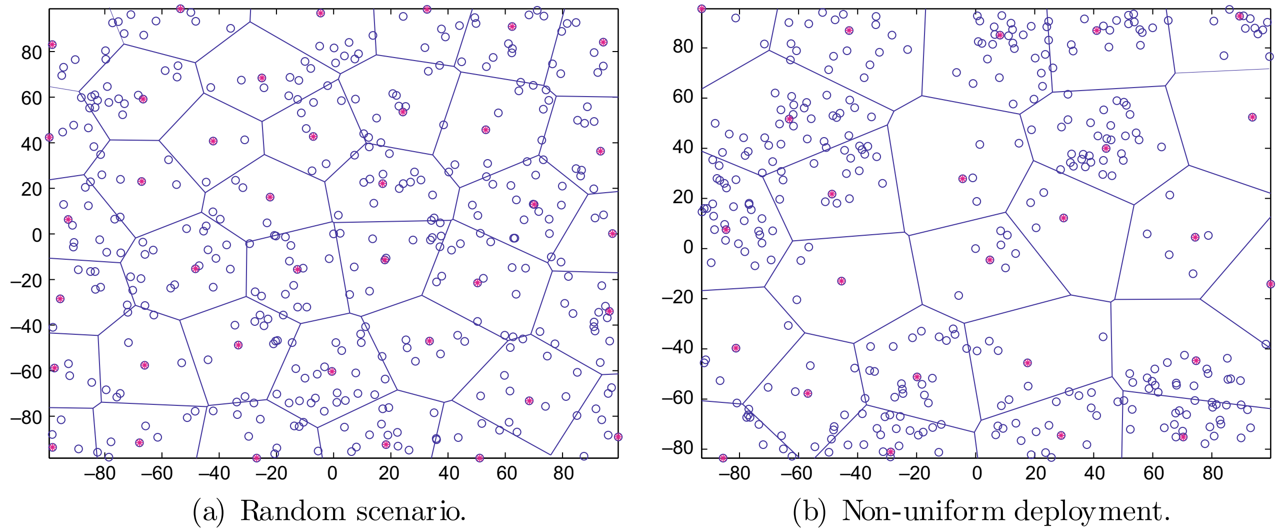


Fig. 2. Examples of random and non-uniform deployment scenarios. CPCP achieves uniform distribution of the cluster head nodes. (a) Random scenario and (b) non-uniform deployment.

4.6. Phase VI: data communication

Once clusters are formed and active sensors are selected, the data communication phase begins where the active sensor nodes periodically collect data and send it to the cluster head nodes. The cluster head nodes aggregate the data from the cluster members and route the aggregated data packets over the pre-determined multi-hop paths to the sink.

5. Simulation set-up

In this section we discuss the set-up for Matlab simulations performed with CPCP. In all the simulations we measure the percentage of the area covered by the active sensor nodes over time. Since active nodes selected in the activation phase of CPCP maximally cover the monitored area, the measured network coverage provided by these active nodes is the same as the coverage that would be provided by all alive nodes in the network.

We perform two sets of simulations. In the first set of simulations, we compare the performance of CPCP in the cases when the different cost metrics introduced in Section 3 are used for the selection of cluster head nodes, active sensors and routers. In the second set of simulations, we compare CPCP with the HEED clustering protocol as a representative of the energy-aware clustering protocols. Furthermore, in our simulations we vary the amount of data aggregation and the network scenario, as described next, to determine the performance of CPCP over a wide range of conditions.

5.1. Data aggregation

In many applications, the cluster head nodes aggregate the received data, thereby reducing the total energy required for transmitting data back to the sink. The amount of aggregated data produced by the cluster head nodes depends on the data aggregation algorithm as well as on the application requirements and the type of sensor data. In our simulations, we provide results for scenarios when the cluster head nodes aggregate their received data more or less efficiently, meaning that they provide different numbers of aggregated data packets. In particular, we present results of simulations where the cluster head nodes aggregate all data into a single outgoing packet, as well as when they reduce the amount of collected data by half, and when the aggregated data is 80% of the total data load collected within the cluster.

5.2. Network scenario

We conduct simulations for two scenarios: a network with 200 nodes deployed over an area of size $100 \times 100\ \text{m}^2$, and a network with 400 nodes deployed over an area of size $200 \times 200\ \text{m}^2$. The nodes are deployed either randomly or non-uniformly. In the case of the random deployment, the (x, y) locations of the sensor nodes are randomly chosen based on a uniform distribution. For simplicity, we call this deployment the *random* deployment. The non-uniform deployment corresponds to the case when nodes in certain parts of the network are more "grouped" together, so that they provide higher redundancy in coverage than the nodes located in scarcely covered areas of the network. We call this deployment the *non-uniform* deployment. The data sink is fixed and located in the center of the network. The simulations are conducted with $R_{\text{cluster}} = 2 \cdot R_{\text{sense}}$. The simulation parameters are summarized in Table 1.

5.3. Energy model

We assume that the sensor nodes have the ability to adjust their transmission power according to the distance of the receiving node. We use the free-space energy model defined in [4], where the energy required to transmit a p-bit packet is equal to

$$E_{\text{tx}} = p \cdot (E_{\text{amp}} + \epsilon_{\text{fs}} \cdot d^n), \tag{10}$$

Table 1
Simulation parameters

Parameter	Acronym	Value
Tx/Rx electronics constant	E_{amp}	50 nJ/bit
Amplifier constant	ϵ_{fs}	10 pJ/bit/m^2
Path-loss exponent	n	2
CH energy threshold	E_{th}	10^{-4} J
Packet size	p	30 bytes
Packet rate	B	1 packet/s
Transmission range (broadcast)	R_{bc}	70 m
Sensing range	R_{sense}	15 m
Cluster range	$R_{cluster}$	30 m

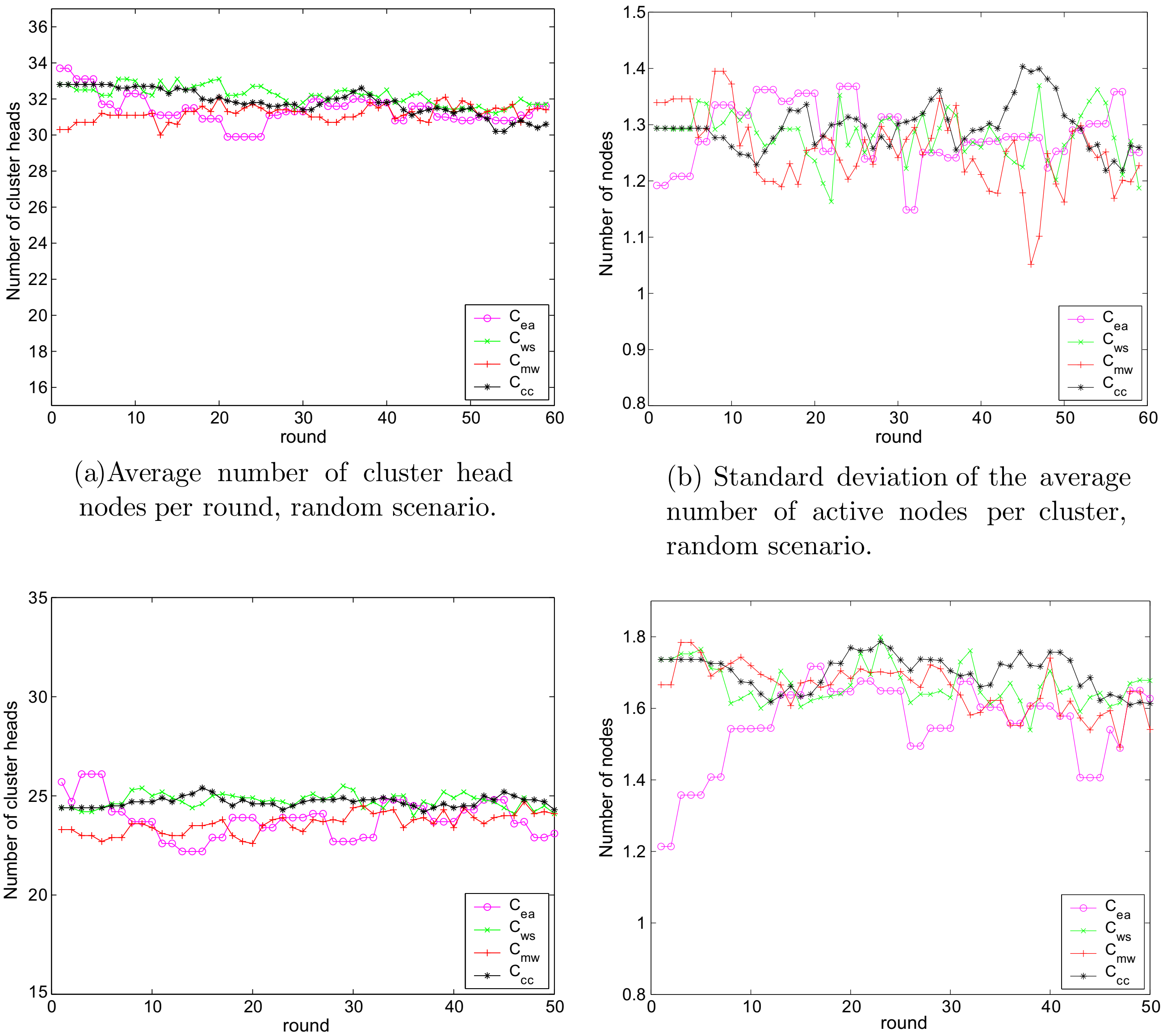
and the energy to receive a p-bit packet is

$$E_{rx} = p \cdot E_{amp}. \quad (11)$$

The parameters E_{amp} and ϵ_{fs} are the parameters of the transmission/reception circuitry, and n is the path-loss exponent, listed in Table 1.

5.4. Clusters created using CPCP

The results of simulations show that CPCP disperses cluster head nodes uniformly, as shown in Fig. 2, there-



(a) Average number of cluster head nodes per round, random scenario.
(b) Standard deviation of the average number of active nodes per cluster, random scenario.
(c) Average number of cluster head nodes per round, non-uniform scenario.
(d) Standard deviation of the average number of active nodes per cluster, non-uniform scenario.

Fig. 3. Performance of CPCP: the average number of cluster head nodes per round and the standard deviation of the average number of active nodes per cluster when the network is operating at 100% coverage. (a) Average number of cluster head nodes per round, random scenario, (b) standard deviation of the average number of active nodes per cluster, random scenario, (c) average number of cluster head nodes per round, non-uniform scenario and (d) standard deviation of the average number of active nodes per cluster, non-uniform scenario.

by producing small variations in the number of cluster head nodes elected in successive communication rounds. Thus, in the case of the random deployment scenario (Fig. 2a), the data load produced in the network is more uniformly distributed across the cluster head nodes over time. In the case of the non-uniform scenario (Fig. 2b) the cluster head nodes in redundantly covered areas serve clusters with a higher number of nodes than the cluster head nodes in sparsely covered network areas.

Fig. 3 shows the average number of cluster head nodes per round as well as the standard deviation of the average number of active nodes per cluster over the time period during which the network provides full coverage of the monitored area. For both scenarios (random and non-uniform) the variations in the number of cluster head nodes per round over time are small. The number of cluster head nodes is lower in the non-uniform deployment scenarios due to the existence of larger areas with very low densities of sensor nodes. Also, when the network is deployed in a non-uniform manner, the standard deviation in the average number of active nodes per cluster is slightly higher than in case of random deployment, as shown in Fig. 3b and d.

6. Case I: performance of CPCP using different cost metrics

Our goal with this first set of simulations is to show the effects of the different cost metrics on the performance of the network, specifically focusing on coverage-time. These costs are used to select cluster head nodes, active sensors and routing nodes. The cluster head nodes aggregate the data packets received from the active sensors within the cluster into one outgoing packet, and this packet is routed to the sink via shortest-cost routes determined in the route update phase.

6.1. Coverage-time as the network scales

First we find the coverage-time using the different cost metrics as the network scales from 100×100 m^2 with 200 sensor nodes to 200×200 m^2 with 400 sensor nodes for both random and non-uniform deployment scenarios. The results for the network of size 100×100 m^2 with 200 sensor nodes are shown in Fig. 4. When the selection of cluster head nodes, active nodes and routers is done using the minimum-weight cost (C_{mw}) and the weighted-sum cost (C_{ws}), the improvement in the time during which the randomly deployed network can provide full coverage (100%) over the energy-aware cost (C_{ea}) is 30% and 22%, respectively (Fig. 4a). For the non-uniform scenario, these improvements of coverage-time increase to 38% and 25%, respectively (Fig. 4b). After the network coverage drops below 95%, C_{mw} and C_{ws} improve the coverage-time by 15–20% in the case of random deployment, and by 20–25% in the case of non-uniform deployment. Overall, these two metrics are able to provide longer coverage-time over C_{ea} in both the random and non-uniform network deployment scenarios.

Fig. 5 shows the results of the simulations for the larger network (200×200 m^2 with 400 nodes). Again, the C_{mw} and C_{ws} metrics provide longer coverage-time compared to the C_{ea} metric. The improvement in the coverage-time is higher in the non-uniform deployment scenario, with an improvement in coverage-time for 100% coverage of 28% using C_{mw} and 26% using C_{ws}. The minimum-weight cost C_{mw} again provides the longest time during which 100% of the network is covered compared to all the other cost metrics in both the random (Fig. 5a) and non-uniform (Fig. 5b) network deployments. This is expected, since the minimum-weight cost assigns a high cost to the nodes that are critical to maintaining 100% coverage.

Compared with the other metrics, the coverage redundancy cost metric (C_{cc}) provides the worst time during

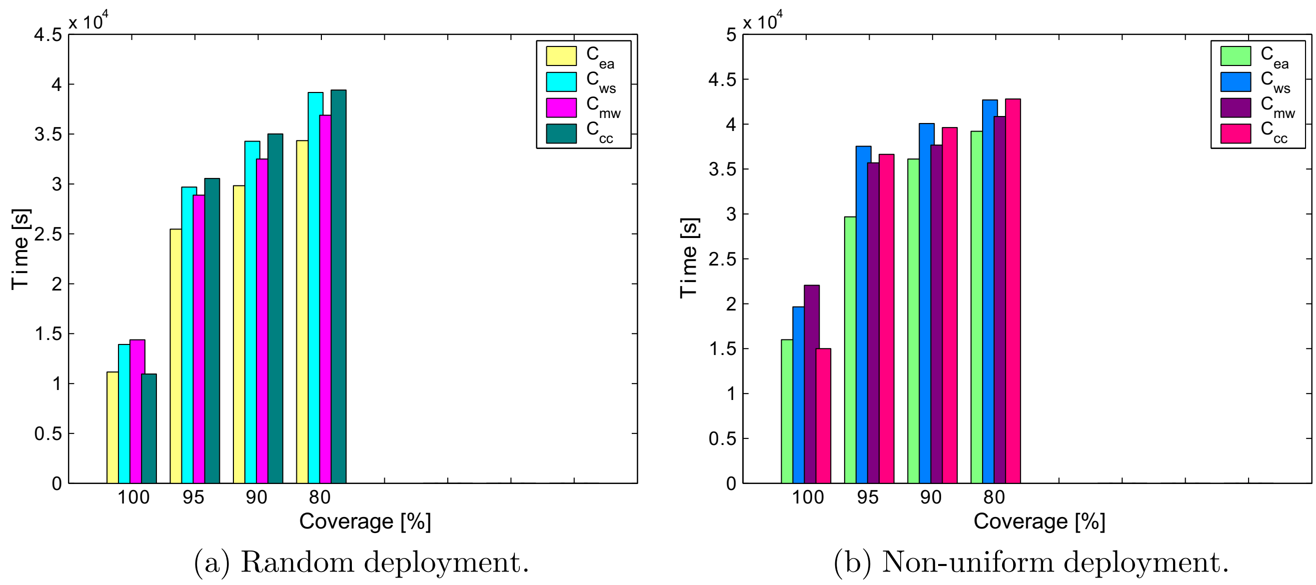
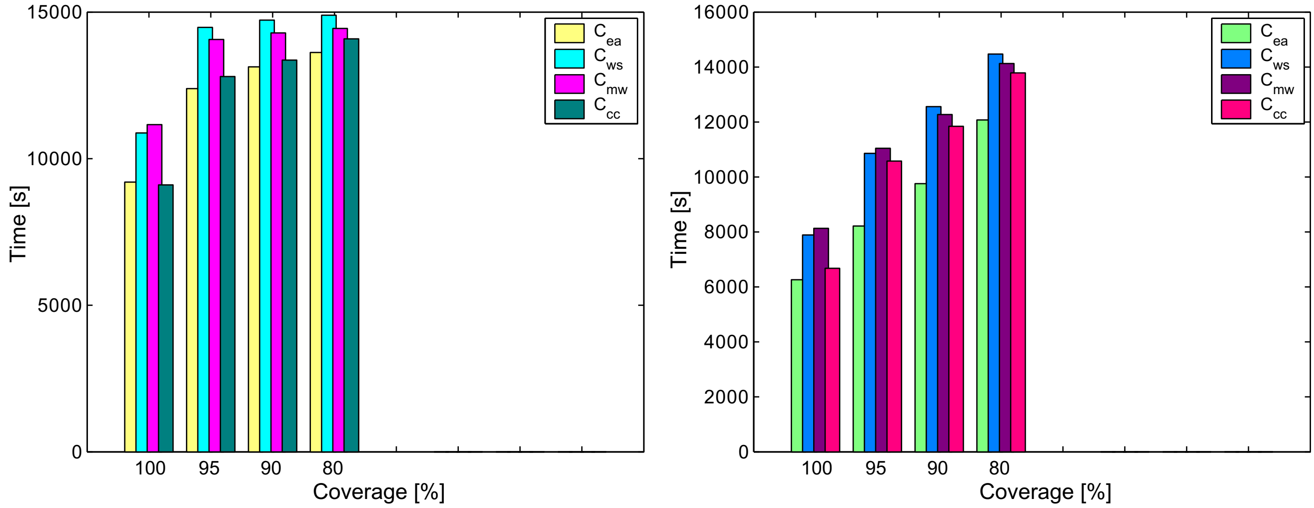


Fig. 4. Coverage-time for a network of size 100×100 m^2 with 200 nodes utilizing CPCP with different cost metrics. (a) Random deployment and (b) non-uniform deployment.



(a) Random deployment. (b) Non-uniform deployment.

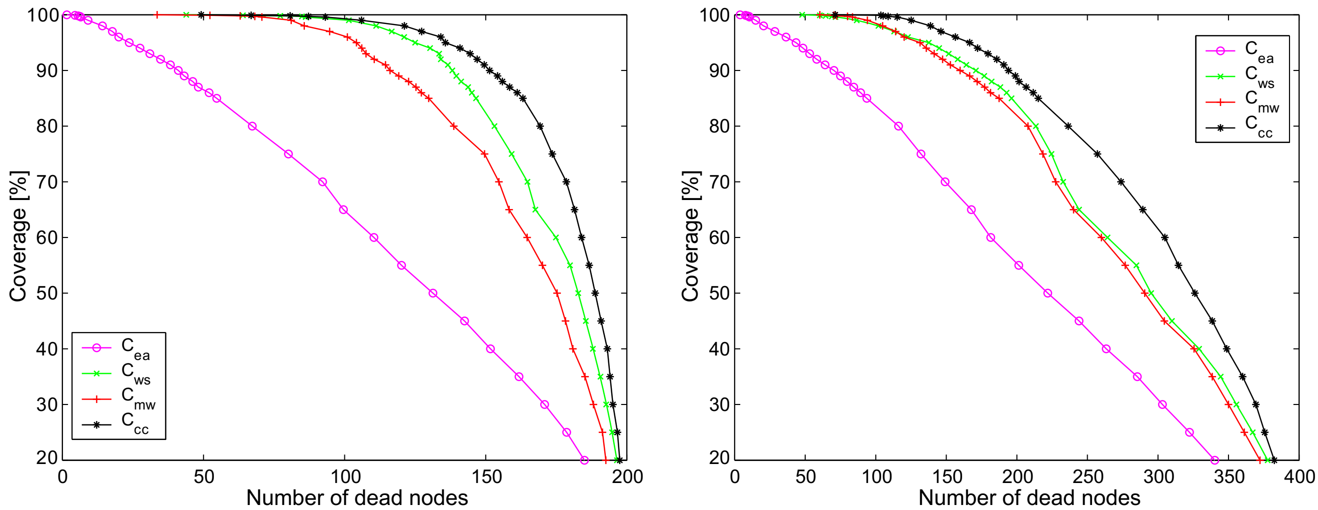
Fig. 5. Coverage-time for a network of size 200 × 200 m² with 400 nodes utilizing CPCP with different cost metrics. (a) Random deployment and (b) non-uniform deployment.

which the network is able to monitor the entire (100%) area. However, in the case of smaller networks (100 × 100 m²), after the coverage starts to drop below 85%, C_{cc} shows slightly better performance than the other cost metrics. In the larger network (200 × 200 m²), C_{cc} always performs worse than all the other cost metrics. Although the coverage-redundancy cost metric selects redundantly covered nodes, it does not consider the node's remaining energy, resulting more often in the loss of nodes compared with the other metrics. In the case of smaller networks, this is less obvious than in the case of larger networks, which is one reason that the coverage-redundancy cost metric never outperforms the other metrics.

Thus, the difference in the results obtained with the coverage redundancy metric for both simulated scenarios (small and large network) illustrates the importance of applying the same coverage-aware approach in the selection of not only cluster head nodes, but also in the selection of data routers as well. With the increase of network size, routing is done over a larger number of hops; therefore, there is a greater need to avoid the critical nodes (non-redundantly covered nodes or nodes with low energy).

6.2. Loss of sensor nodes

Fig. 6 shows how the network coverage decreases as the number of dead nodes increases for the two network scenarios (100 × 100 m² with 200 nodes and 200 × 200 m² with 400 nodes). The C_{ea} cost metric contributes to uniform energy dissipation among the sensor nodes, resulting in the highest amount of lost coverage for a given number of dead nodes compared to the other three metrics. On the



(a) Network 200 × 200m^2. (b) Network 400 × 400m^2

Fig. 6. Network coverage as a function of the number of dead nodes. (a) Network 200 × 200 m² and (b) network 400 × 400 m².

other hand, the coverage redundancy cost metric C_{cc} has the least amount of coverage loss for a given number of dead nodes. This shows that the energy-aware cost metric treats all nodes equally, while the coverage redundancy cost metric preserves nodes that are critical for the coverage task. As the coverage redundancy cost metric does not consider a node's remaining energy, a node's cost only changes when one of its neighboring nodes dies. This infrequent change in node cost results in non-balanced energy consumption, with the result that redundantly covered sensors that are not critical to the coverage of the network, are used first.

Coverage as a function of the number of dead nodes using the other two cost metrics (C_{mw} and C_{ws}) is in between that of C_{ea} and C_{cc}. Note, however, that C_{mw} and C_{ws} provide the longest coverage-time compared to the other two metrics. This clearly demonstrates the fact that it is important to look at *both* minimizing or balancing energy dissipation and preserving critical nodes to maintain high levels of coverage for a longer time.

6.3. Average remaining energy of cluster head nodes

Fig. 7 shows the average remaining energy of the selected cluster head nodes over time. When C_{ea} is used, sensor nodes with the highest remaining energy are selected as cluster head nodes. Compared to using the C_{ea} cost, using the C_{mw} and C_{ws} cost metrics prevent non-redundantly covered nodes from being selected as cluster head nodes at the beginning of the network lifetime, resulting in rotation of cluster head roles among the most redundantly covered sensor nodes. The frequent selection and excessive energy consumption of elected cluster head nodes using the C_{mw} and C_{ws} costs lead to loss of the most redundantly covered nodes. At that point, the low redundantly covered sensor nodes resume the cluster head roles and, since they have not yet served as cluster heads previously, these nodes still have relatively high remaining energies. This is the reason why at a certain point, after the network coverage using the C_{ea} cost drops below 95% (by comparing Figs. 4b and 7a, and Figs. 5b and 7b) the average energy of elected cluster head nodes using C_{mw} and C_{ws} is larger than the average energy of cluster head nodes elected using the C_{ea} metric.

6.4. Coverage-aware routing

Fig. 8 shows the average number of hops in the routes from the cluster head nodes to the sink as the network coverage changes. These results show that the C_{cc} cost metric finds routes with the smallest number of hops compared with the other cost metrics. On the other hand, the weighted-sum (C_{ws}) and minimum-weight (C_{mw}) metrics route data packets over the longest paths, since these metrics avoid the high cost nodes (those with low remaining energy and/or with low redundancy in coverage). In the case of the smaller networks ($100 \times 100\,m^2$, shown in Fig. 8a), data packets are routed over a relatively small number of hops (1–2), so the differences in the average path lengths for the various cost metrics are not significant. Therefore, the choice of routing paths does not significantly affect the network performance.

However, in the case of the larger networks ($200 \times 200\,m^2$), the differences in the lengths of the routing paths are more obvious for different costs, as shown in Fig. 8b. As network coverage decreases, the minimum-weight (C_{mw}) and weighted-sum (C_{ws}) metrics further increase the number of hops in their routing paths, trying to avoid critical nodes. When the coverage of the network starts to decrease as a result of losing nodes, the energy-aware metric also increases the average number of hops in order to balance energy consumption among sensor nodes. On the other hand, the coverage redundancy cost keeps route lengths fairly constant, since this cost metric does not depend on the nodes' remaining energies.

As a result of the increased lengths of the routing paths, the average energy spent to route packets from each cluster head node to the sink also increases for the C_{mw}, C_{ws}

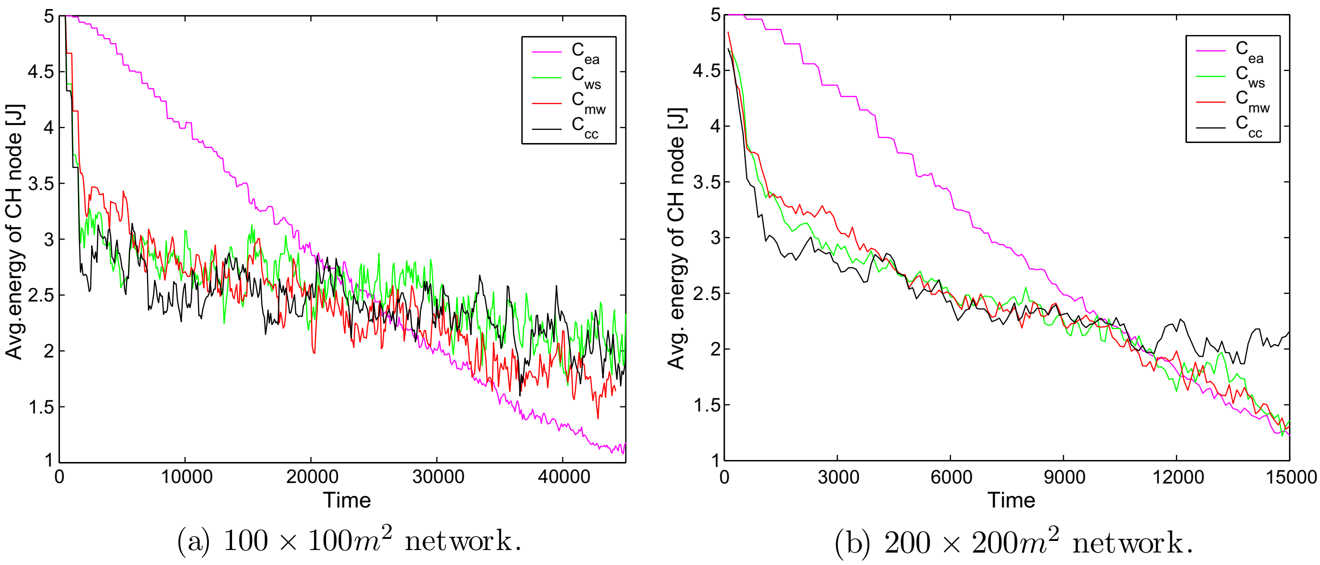
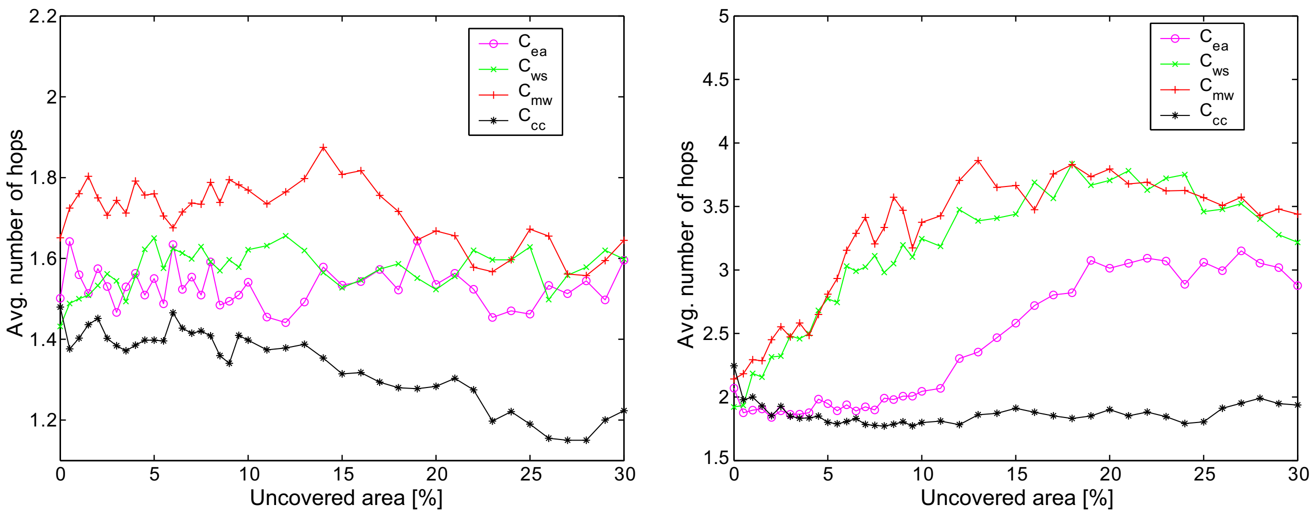


Fig. 7. Remaining energy levels of selected cluster head nodes over time. (a) $100 \times 100\,m^2$ network and (b) $200 \times 200\,m^2$ network.



(a) $200 \times 200m^2$ network. (b) $400 \times 400m^2$ network.

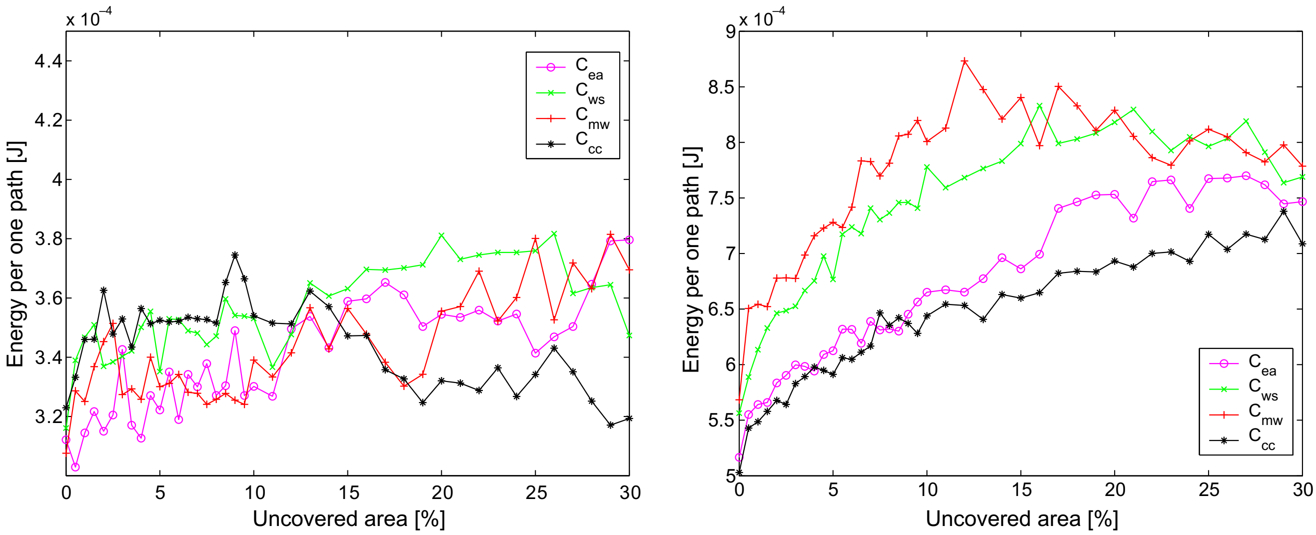
Fig. 8. Average number of hops in the routes from the cluster head nodes to the sink. (a) 200 × 200 m² network and (b) 400 × 400 m² network.

and C_{ea} cost metrics, as illustrated in Fig. 9. Again, in the large networks, this increase in average energy spent per path is more obvious than in the case of the smaller networks.

The coverage redundancy cost C_{cc} does not prevent routing over the low-energy nodes, which speeds up the loss of these nodes. The average number of hops used for data routing is small, and it stays relatively constant throughout the network lifetime (Fig. 8). Using C_{cc} the average energy spent per route is smaller compared to other cost metrics, once the network starts loosing coverage (Fig. 9). The reason for this is that the network loses a significant number of nodes, which reduces the total data load routed through the network. In the case of the smaller networks, where data are routed over a small number of hops, this is the reason that C_{cc} starts to outperform the other cost metrics when the network's coverage starts to decrease significantly. However, when data are routed over a larger number of hops, C_{cc} shows an inability to choose "good" routing paths, which is the reason this cost metric does not perform well. This again illustrates the importance of considering both energy and coverage in the selection of routing paths for coverage-preserving applications.

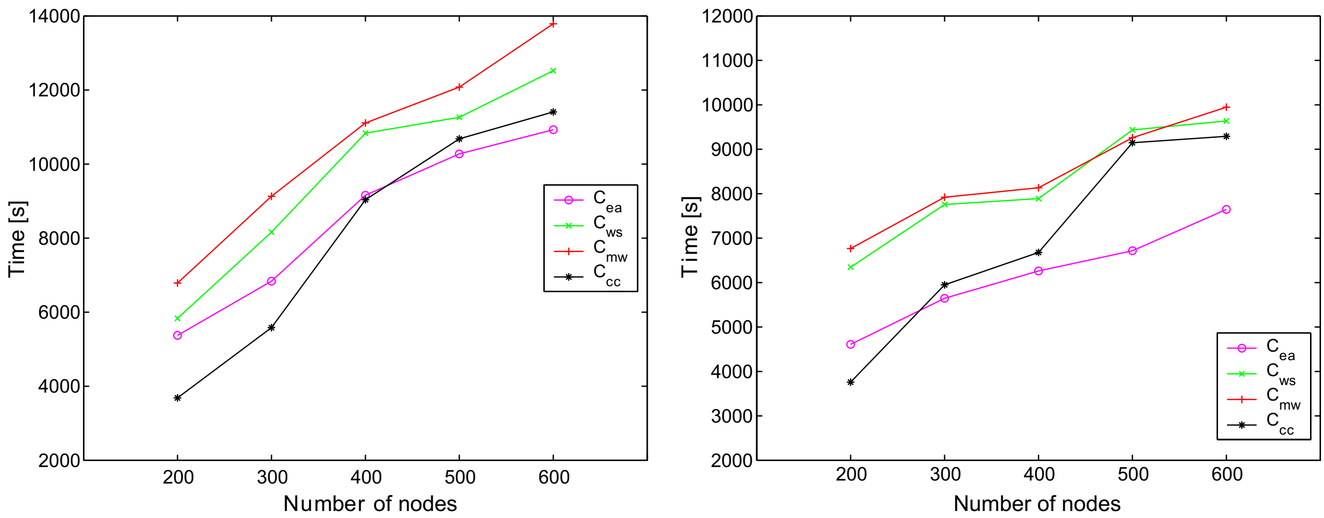
6.5. Increasing the number of nodes

Fig. 10 shows the time during which the 200 × 200 m² networks provide 100% coverage when the number of nodes increases from 200 to 600, for both the random and non-uniform deployments. In all cases the C_{ws} and C_{mw} cost metrics provide longer network coverage-time compared to the C_{ea} cost metric. The improvements in net-



(a) $200 \times 200m^2$ network. (b) $400 \times 400m^2$ network.

Fig. 9. Average energy dissipated per route from the cluster head nodes to the sink. (a) 200 × 200 m² network and (b) 400 × 400 m² network.



(a) Random deployment.

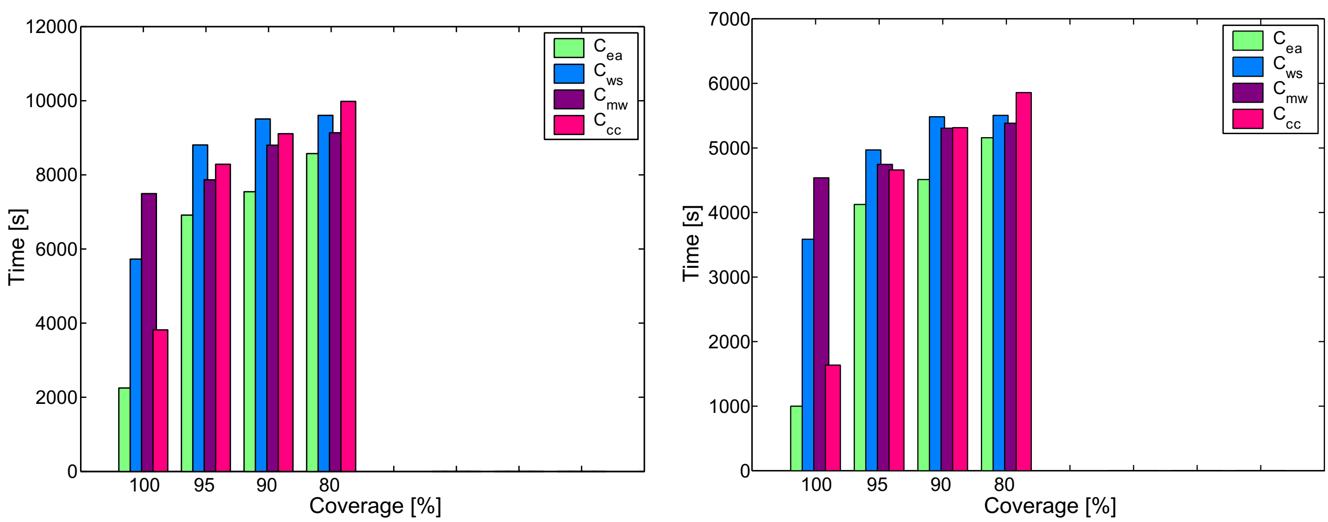
(b) Non-uniform deployment.

Fig. 10. Time during which the network preserves 100% coverage of the monitored area as a function of the number of nodes in the 200×200 m^2 network. (a) Random deployment and (b) non-uniform deployment.

work coverage-time obtained with the C_{ws} and C_{mw} cost metrics compared with the C_{ea} cost metric in the non-uniform network deployment scenarios is always larger than in the random network deployments. Therefore, the advantages of using minimum-weight and weighted sum cost metrics over energy-aware cost are even more obvious in the non-uniform sensor network deployment scenarios.

6.6. Impact of aggregation

When cluster head nodes perform less efficient data aggregation, meaning that they send more than one packet to the sink, the differences in coverage-time obtained by the coverage-aware cost metrics and the energy-aware cost metric increase. Fig. 11 shows the coverage-time obtained with different cost metrics when the cluster head nodes forward 50% and 80% of all packets received from the cluster members in one communication round. In both cases the C_{mw} and C_{ws} cost metrics perform even better compared to the C_{ea} metric than in the case when the cluster head aggregates all incoming data into one packet. The improvement in the time during which the network provides 100% coverage using C_{ws} and C_{mw} compared with using C_{ea} is 2.5× and 3.2×, respectively, when the cluster heads forward half of the total received data. When the cluster heads forward 80% of the total received load, these improvements are even higher – 3.6× using C_{ws} and 4.5× using C_{mw} compared with using the C_{ea} cost metric.



(a) Cluster head nodes aggregate all received data to 50% of the total load.

(b) Cluster head nodes aggregate all received data to 80% of the total load.

Fig. 11. The effect of aggregation efficiency on coverage-time for the 200×200 m^2 network with 400 non-uniformly deployed nodes. (a) Cluster head nodes aggregate all received data to 50% of the total load and (b) cluster head nodes aggregate all received data to 80% of the total load.

7. Case II: performance of CPCP compared with HEED

As mentioned previously, many clustering protocols are mainly focused on achieving balanced energy consumption in the network in order to prolong the lifetime of the individual sensor nodes, without regard to the network's ability to cover the region of interest. In order to illustrate the difference between coverage-preserving and energy balancing approaches for organization of cluster-based sensor networks, we compare CPCP with the HEED protocol [7]. HEED is a scalable clustering protocol that uniformly distributes cluster head nodes throughout the network and prolongs the lifetime of sensor nodes by distributing their energy consumption.

7.1. Overview of HEED

HEED (hybrid energy-efficient distributed clustering) is an iterative clustering protocol that uses information about the nodes' remaining energy and their communication costs in order to select the best set of cluster head nodes. During the clustering process, a sensor node can be either a tentative cluster head, a final cluster head, or it can be covered (meaning that it has heard an announcement message from a final cluster head node). At the beginning of the clustering phase, a node with higher remaining energy has a higher probability CH_{prob} of becoming a tentative cluster head. If the node becomes a tentative cluster head, it broadcasts a message to all sensor nodes within its cluster range to announce its new status. All nodes that hear from at least one tentative cluster head choose their cluster head nodes based on the costs of the tentative cluster head nodes. For this purpose, the authors in [7] define the *average reachability power* (AMRP), which is a cost metric used to "break ties" in the cluster head election process. The AMRP of a node u is defined as the mean of the minimum power levels required by all M nodes within the cluster range to reach the node u

$$\text{AMRP}(u) = \frac{\sum_{i=1}^{M} \text{Min}Pwr(i)}{M}. \tag{12}$$

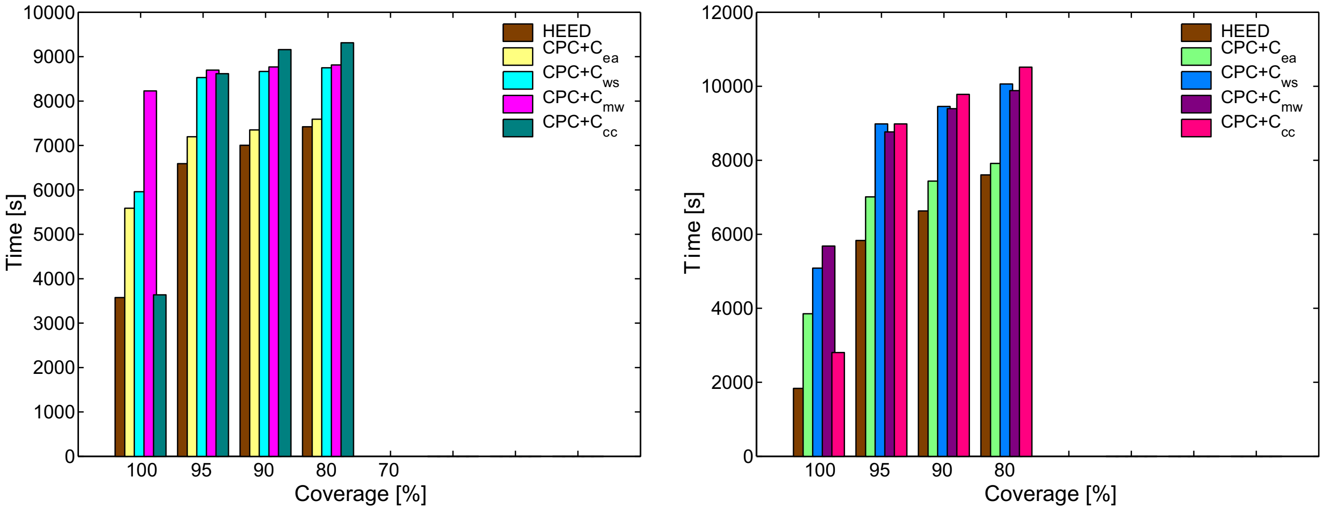
During each iteration, a node that is not "covered" by any final cluster head can elect itself to become a new tentative cluster head node based on its probability CH_{prob}. Every node then doubles its CH_{prob} and goes to the next step. Once the node's CH_{prob} reaches 1, the node can become a final cluster head, or it can choose its cluster head as the least cost node from the pool of final cluster head neighbors. If the node completes HEED execution without selecting its final cluster head, then it considers itself uncovered and becomes a final cluster head for the upcoming round.

Once the clusters are formed, all sensors send their data to the cluster head, where the data are aggregated into a single packet. The cluster head nodes form a network backbone, so packets are routed from the cluster head nodes to the sink in a multi-hop fashion over the cluster head nodes.

7.2. Simulation results: all sensors active

We compare our CPCP with HEED in scenarios where 400 sensor nodes are deployed either randomly or non-uniformly over a $200 \times 200\,\text{m}^2$ region. In simulations of CPCP we follow the same scheme introduced in HEED, where the elected cluster head nodes form a spanning tree for inter-cluster routing of data to the sink. We assume that each cluster head node aggregates its received data packets into one packet that is sent to the data sink located in the middle of the area.

In HEED, all sensor nodes continue their sensing task after the clusters are formed. Therefore, we adopt the same approach in CPCP for these simulations, and hence all sensor nodes remain in the active state during the communication phase of CPCP. In contrast to the previous set of simulations, where the intra-cluster communication was established among a small number of active sensor nodes and their cluster heads, here the cluster head nodes spend



(a) Random deployment. (b) Non-uniform deployment.

Fig. 12. Comparison of HEED and CPCP in terms of coverage-time. (a) Random deployment and (b) non-uniform deployment.

a much larger amount of energy in communicating with their cluster members.

HEED is a distributed clustering protocol that does not depend on the synchronization of sensor nodes in the network. However, the subsequent broadcasting of announcement messages from the tentative cluster head nodes in each clustering phase requires quite a bit of energy. In CPCP however, the nodes using cost C_{mw} and C_{ws} need to periodically broadcast their remaining energy, which is an additional burden on the limited energy resources. Both clustering algorithms generate uniformly dispersed cluster head nodes. However, in applications where the sensor network has to maintain full coverage, the choice of cluster head nodes significantly impacts the network's coverage-time.

Fig. 12 shows the network coverage over time for HEED and for CPCP using different cost metrics. As shown in this figure, the results for CPCP from these simulations are quite similar to the results presented in Section 6. The minimum-weight cost metric C_{mw} provides 100% coverage for the longest time and the weighted-sum cost metric C_{ws} provides almost full coverage for the longest period of time. The improvement of CPCP over HEED in terms of 100% coverage-time is noticeable using all the cost metrics. Compared with HEED, the time during which the randomly deployed network provides full coverage of the monitored area on average increases by 67% and 125% using C_{ws} and C_{mw}, respectively. In the non-uniformly deployed network this time of full coverage increases even more – by 180% and 260% using C_{ws} and C_{mw}, respectively, compared with HEED.

HEED gives priority to the nodes with higher remaining energy to be elected as cluster heads. In the case when nodes can manage variable transmission power, the AMRP cost metric (used by the nodes to decide among the best cluster head candidate) depends on the distance between the potential cluster head and its neighboring nodes. However, AMRP does not provide any information about the nodes' spatial distribution and therefore about the redundancy in coverage provided by the nodes. For example, Fig. 13 shows two cases of a node S with three neighboring nodes that are all at the same distance from the node S. In both cases node S had the same AMRP cost since it needs the same transmission power to reach all three neighboring nodes. However, in the first case the sensing area of node S is completely covered by the sensing areas of its neighboring nodes, while in the second case this is not true. Therefore, node S will have higher C_{mw} and C_{ws} costs

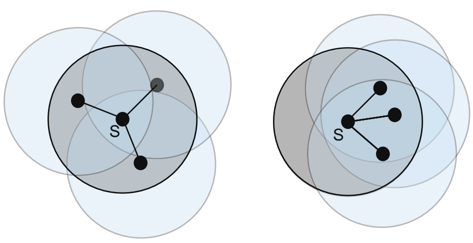


Fig. 13. Two situations where sensor node S has the same AMRP cost but different coverage redundancy.

in the second case. This shows that in coverage-preserving applications the information about the coverage redundancy is crucial to maintaining complete coverage for long periods of time.

Furthermore, CPCP and HEED produce similar numbers of clusters, as illustrated in Fig. 14a, which shows the number of cluster head nodes during the time in which the network provides up to 90% coverage. The cost for extended coverage-time using coverage-aware cluster head selection is paid by more dead nodes compared to HEED, as shown in Fig. 14b. However, while the network loses fewer nodes using HEED, the network is not able to provide coverage as long as it can using CPCP, as the nodes that die in HEED are more important to coverage.

7.3. *Hybrid HEED: HEED combined with coverage-preserving sensor activation*

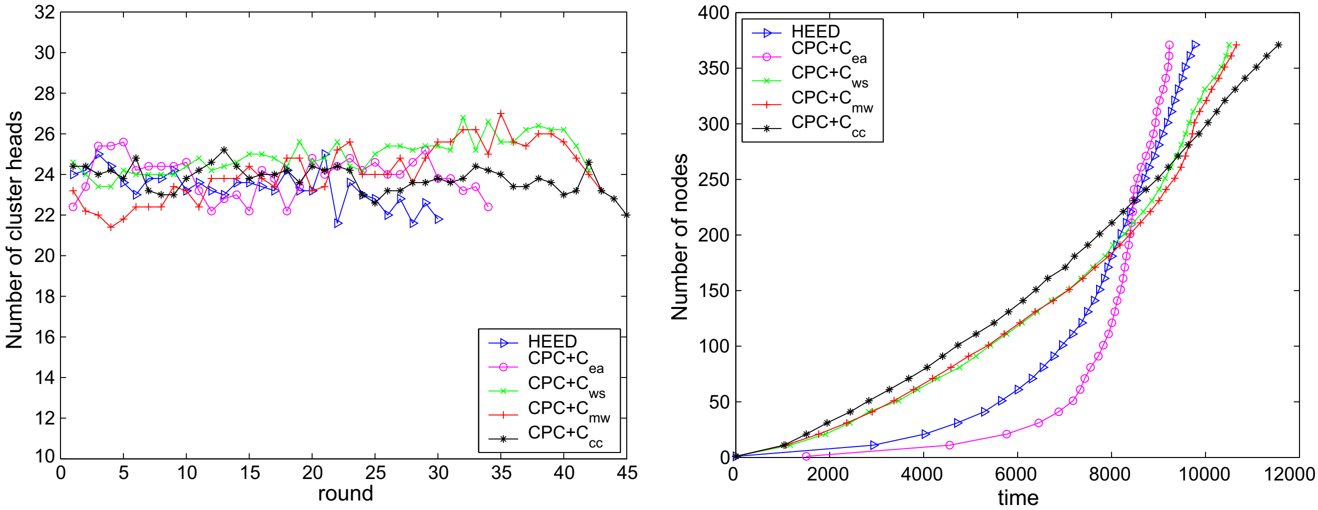
Finally, we measure the coverage-time obtained using a hybrid version of HEED. In the *hybrid HEED* protocol, the clusters are formed according to the original HEED algorithm, and this cluster formation phase is followed by the selection of active sensor nodes that are able to maximally cover the network. The C_{ws} and C_{mw} cost metrics are used for the selection of active sensors, while the rest of the nodes are put to sleep. We compare hybrid HEED with two cases of CPCP. The first case corresponds to CPCP described in Section 6. The second case corresponds to CPCP where the routing of cluster head packets is done using the cluster head nodes rather than the sensor nodes.

Fig. 15 shows the lifetime of the network, defined as the time for which 100% and 90% network coverage is preserved. Both variants of CPCP significantly outperform the "hybrid HEED" protocol, which again illustrates the importance of making suitable choices for the cluster head nodes for coverage-preserving sensor network applications.

8. Which cost metric to use?

In all of our simulations, both coverage-aware cost metrics C_{mw} and C_{ws} outperform the energy-aware cost metric C_{ea} in terms of coverage-time. The minimum-weight C_{mw} cost metric provides the best results (longest coverage-lifetime) in all scenarios where the sensor network has to provide complete (100%) coverage of the monitored area. However, the maintenance of full coverage over the monitored area is extremely expensive, since it requires that non-redundantly covered nodes are always turned on, which shortens their lifetime. The weighted-sum cost metric C_{ws} shows better performance than the minimum-weight cost metric after coverage drops a few percentages, since it provides a more balanced relationship between the node's coverage redundancy and its remaining energy. Therefore, in applications that require the maintenance of full coverage, the minimum-weight cost is the best choice, while for applications that can relax this requirement slightly, the weighted-sum cost is the best choice.

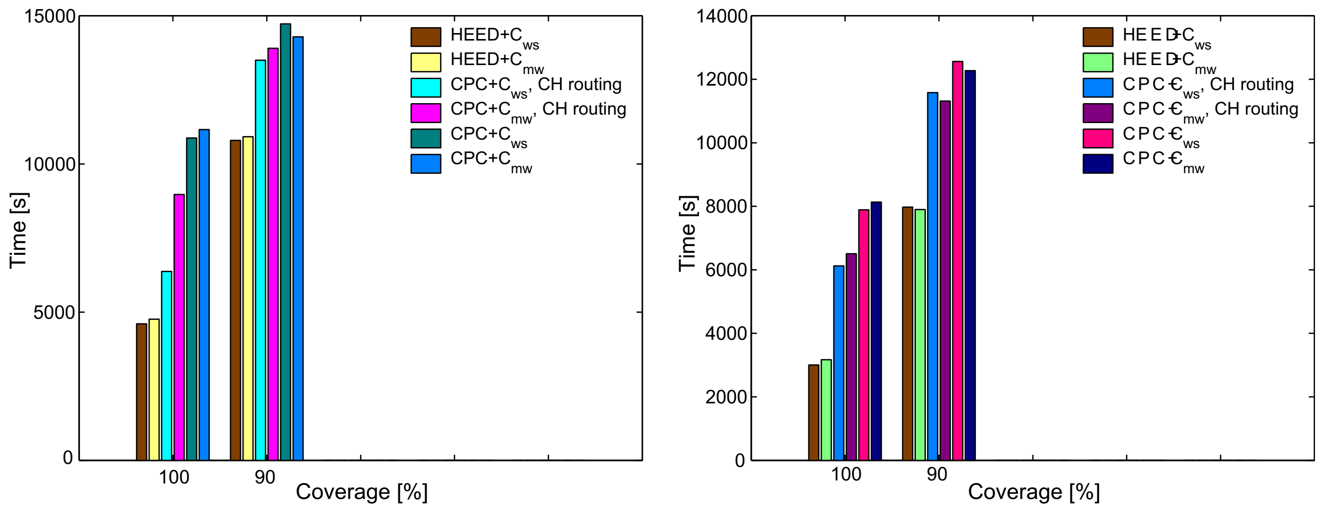
Although the coverage redundancy cost metric C_{cc} depends only on the sensor's coverage, it does not perform well when full coverage is required. This cost metric can



(a) Average number of cluster head nodes.

(b) Number of dead nodes.

Fig. 14. Comparison of HEED and CPCP in terms of the number of cluster heads per round and the number of dead nodes over time. (a) Average number of cluster head nodes and (b) number of dead nodes.



(a) Random deployment of sensor nodes.

(b) Non-uniform deployment of sensor nodes.

Fig. 15. Comparison of Hybrid HEED and CPCP in terms of coverage-time for random and non-uniform deployment scenarios. (a) Random deployment of sensor nodes and (b) non-uniform deployment of sensor nodes.

potentially be used in small size networks, where data routing is not needed or is done over very small numbers of hops. Finally, C_{ea} performs worse than any other cost metric, and it should not be the choice for any application that requires persistent coverage of the monitored area.

9. Conclusion

In this paper we explore different coverage-aware cost metrics for the selection of the cluster head nodes, active nodes and routers in wireless sensor networks whose aim is to maintain coverage of a monitored space. In such coverage-preserving applications, both the remaining energy of the sensor nodes as well as the redundancy in their coverage have to be jointly considered when determining the best candidates for cluster head nodes, active nodes and data routers. Through extensive simulations we illustrated the shortcomings of using remaining energy or coverage redundancy as the only criteria for the decision about the nodes' roles in cluster-based wireless sensor networks. Instead, using the coverage-aware cost metrics prolong coverage-time over the monitored area, by minimizing the use of sensors in sparsely covered areas and those with low remaining energy.

Our future work will include the implementation of CPCP in a sensor network test-bed and full experimental evaluation of the proposed coverage-aware cost metrics. Furthermore, we are interested in exploring this coverage-aware approach in camera-based wireless sensor networks. Our first results of using the modified cost metrics defined here for the selection of cameras in a visual sensor networks [26] prove that these cost metrics can be successfully applied into this type of ad hoc network. Cameras' sensing areas are strictly defined by their viewing volumes, which enables us to exactly determine the amount of overlap among cameras' sensing areas once the cameras are calibrated. The redundancy of image data captured by different cameras with overlapped views is high, which in energy constrained visual sensor networks provides space for minimization of the total data load by extracting the most important image data from all the captured images. In a distributed camera-based sensor network, finding a central camera-node that can perform the cluster head role, which collects image data from the other cameras and extracts the most relevant image information, is crucial. The work presented in this paper will serve as a corner stone to analyze this problem.

Acknowledgements

This work was supported in part by the National Science Foundation under Grants ECS-0428157 and CNS-0448046.

References

[1] M. Perillo, W. Heinzelman, DAPR: a protocol for wireless sensor networks utilizing an application-based routing cost, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'04), 2004.

[2] M. Qin, R. Zimmermann, An energy-efficient voting-based clustering algorithm for sensor networks, in: Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN), 2005.

[3] M. Chatterjee, S.K. Das, D. Turgut, WCA: a weighted clustering algorithm for mobile ad hoc networks, Cluster Computing 5 (2) (2002) 193–204.

[4] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1 (4) (2002).

[5] M.J. Handy, M. Haase, D. Timmermann, Low energy clustering hierarchy with deterministic cluster head selection, in: Proceedings of IEEE MWCN, 2002.

[6] S. Bandyopadhyay, E.J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2003.

[7] O. Younis, S. Fahmy, Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach, IEEE Transactions on Mobile Computing 3 (4) (2004).

[8] T. Shu, M. Krunz, S. Vrudhula, Power balanced coverage-time optimization for clustered wireless sensor networks, in: Proceedings of MobiHoc'05, 2005.

[9] S. Soro, W. Heinzelman, Prolonging the lifetime of wireless sensor networks via unequal clustering, in: Proceedings of the Fifth International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (IEEE WMAN'05), 2005.

[10] H. Chan, A. Perrig, ACE: an emergent algorithm for highly uniform cluster formation, in: Proceedings of EWSN, 2004.

[11] V. Kawadia, P.R. Kumar, Power control and clustering in ad hoc networks, in: Proceedings of INFOCOM, 2003.

[12] R. Virrankoski, A. Savvides, TASC: topology adaptive spatial clustering for sensor networks, in: Proceedings of the 2nd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS'05), 2005.

[13] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking, 2001.

[14] Y. Xu, J. Heidemann, D. Estrin, Energy conservation by adaptive clustering for ad-hoc networks, in: Proceedings of MobiHoc, 2002.

[15] J. Deng, Y. Han, W. Heinzelman, P. Varshney, Scheduling sleeping nodes in high density cluster-based sensor networks, ACM/Kluwer MONET Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks, vol. 10, No. 6, December 2005, pp. 825–835.

[16] J. Deng, Y.S. Han, W. Heinzelman, P. Varsney, Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks, Elsevier's Computer Communications Journal (2005).

[17] K. Bae, H. Yoon, Autonomous clustering scheme for wireless sensor networks using coverage estimation self-pruning, IEICE Transaction on Communications E88-B (3) (2005).

[18] W. Choi, S.K. Das, A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks, in: Proceedings of MobiQuitous Networking Conference, 2004.

[19] M.T. Thai, F. Wang, D. Du, Coverage problems in wireless sensor networks – designs and analysis, International Journal of Sensor Networks, Special Issue on Coverage Problems in Sensor Networks, 2005.

[20] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M.B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, in: Proceedings of IEEE Infocom, 2001.

[21] M. Cardei, J. Wu, Energy-efficient coverage problems in wireless ad-hoc sensor networks, Computer Communications Journal (Elsevier) 29 (4) (2006) 413–420.

[22] M. Cardei, D.Z. Du, Improving wireless sensor network lifetime through power aware organization, ACM Wireless Networks 11 (3) (2005).

[23] F. Ye, G. Zhong, J. Cheng, S. Lu, L. Zhang, PEAS: a robust energy conserving protocol for long-lived sensor networks, in: Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS 2003), 2003.

[24] D. Tian, N.D. Georganas, A coverage-preserving node-scheduling scheme for large wireless sensor networks, in: Processing of ACM Wireless Sensor Network and Application Workshop, 2002.

[25] H. Zhang, J. Hou, Maintaining coverage and connectivity in large sensor networks, in: Proceedings of International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless and Peer-to-Peer Networks, 2004.

[26] S. Soro, W.Heinzelman, Camera selection in visual sensor networks, in: Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2007.

Stanislava Soro received the Dipl.-Ing. degree in Electrical and Computer Engineering in 2000 from the Faculty of Technical Sciences at the University of Novi Sad, Serbia. She received the M.S. and Ph.D. degrees in Electrical and Computer Enginering in 2004 and 2007 from the University of Rochester, Rochester, New York. Currently she is a postdoctoral researcher at the University of Rochester. Her current research interests lie in the areas of wireless communication and networking, multimedia communication, ad hoc and sensor networks and visual sensor networks.

Wendi B. Heinzelman is an associate professor in the Department of Electrical and Computer Engineering at the University of Rochester, and she holds a secondary appointment as an associate professor in the Department of Computer Science. Dr. Heinzelman has also served as Dean of Graduate Studies for the Faculty of Arts, Sciences and Engineering at the University of Rochester since July, 2008. Dr. Heinzelman received a B.S. degree in Electrical Engineering from Cornell University in 1995 and M.S. and Ph.D. degrees in Electrical Engineering and Computer Science from MIT in 1997 and 2000, respectively. Her current research interests lie in the areas of wireless communications and networking, mobile computing, and multimedia communication. Dr. Heinzelman received the NSF CAREER award in 2005 for her research on cross-layer architectures for wireless sensor networks, and she received the ONR Young Investigator Award in 2005 for her work on balancing resource utilization in wireless sensor networks. She is an Associate Editor for the IEEE Transactions on Mobile Computing, and she is an Area Editor for ACM Mobile Computing and Communications Review (MC2R). Dr. Heinzelman is a member of Sigma Xi and the ACM, she is a senior member of the IEEE, and she is co-founder of the N^2 Women (Networking Networking Women) group.