Hierarchical Clustering-task Scheduling Policy in Cluster-based Wireless Sensor Networks

Article *in* IEEE Transactions on Industrial Informatics · September 2017

DOI: 10.1109/TII.2017.2757606

5 authors, including:

Peyman Neamatollahi
Ferdowsi University Of Mashhad

S. Abrishami
Ferdowsi University Of Mashhad

M. Naghibzadeh
Ferdowsi University Of Mashhad

Mohammah-H Yaghmaee
Ferdowsi University Of Mashhad

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2757606, IEEE Transactions on Industrial Informatics

Hierarchical Clustering-task Scheduling Policy in Cluster-based Wireless Sensor Networks

Peyman Neamatollahi, Saeid Abrishami, Mahmoud Naghibzadeh, *Senior Member, IEEE*, Mohammad-Hossein Yaghmaee, *Senior Member, IEEE*, and Ossama Younis

***Abstract*—Organizing sensor nodes into a clustered architecture is an effective method for load balancing and prolonging the network lifetime. However, a serious drawback of the clustering approach is the imposed energy overhead caused by the "global" clustering operations in every round of the Global Round-Based Policy (GRBP). To mitigate this problem, this work proposes a Hierarchical Clustering-task Scheduling Policy (HCSP), which triggers node-driven clustering as opposed to GRBP's time-driven clustering. Based on HCSP, each cluster is reconfigured only once at each local super round. Therefore, the cluster reconfiguration frequency varies on-demand and may differ from one cluster to another throughout the network lifetime. However, in order to refresh the entire network structure, global clustering is performed at the end of every global hyper round. Accordingly, HCSP aims to achieve a more flexible, energy-efficient, and scalable clustering-task scheduling than that of GRBP. This policy mitigates the clustering overhead which is the worst disadvantage of clustering approaches. Energy consumption calculations and extensive simulations show the effectiveness of HCSP in saving energy, and in prolonging the network lifetime.**

***Index Terms*—Wireless sensor networks, network lifetime, energy efficiency, clustering-task scheduling, super round.**

I. Introduction

To increase the longevity of the Wireless Sensor Network (WSN) (see Table I), many solutions [1]–[10] have been proposed, each of which provides different levels of efficiency. Among these solutions, clustering is considered advantageous because it reduces the amount of the raw data transmitted to the sink. In addition, it eliminates collision, need for idle listening, and overhearing in each cluster [11].

Clustering algorithms can be categorized as dynamic or static. In static protocols [12], the node's cluster membership is not modified during the network lifetime; however, the Cluster Head (CH) role rotation among cluster members is permitted. In this case, a high amount of energy is dissipated due to radio wave interference caused by CHs located within each other's cluster range. In contrast, the dynamic clustering protocols balance the load over the sensor nodes by splitting the network's operation into pre-specified time durations called rounds. The sensor nodes are organized in clusters for one round and then reorganized in new clusters for the next round (Global Round-based Policy (GRBP)). There are many proposed protocols in the literature [2]–[7], [11], [13], [14] that globally reconstruct the network in every round based on GRBP, even though only some clusters require reconfiguration. Since global clustering is costly, especially if periodically initiated, it leads to considerable dissipation of energy each time a clustering reconfiguration task is triggered.

TABLE I
Most Cited Acronym List

CH	Cluster Head
DMCC	Distributed Multi-Criteria Clustering
GDHRS	Global Dynamic Hyper Round Scheme
GHR	Global Hyper Round
GRBP	Global Round-based Policy
HCSP	Hierarchical Clustering-task Scheduling Policy
LDSRS	Local Dynamic Super Round Scheme
LSR	Local Super Round
WSN	Wireless Sensor Network

To maximize the longevity of WSN, this research aims at reducing the clustering energy cost by proposing a Hierarchical Clustering-task Scheduling Policy (HCSP). This policy employs two clustering-task scheduling schemes, namely the Local Dynamic Super Round Scheme (LDSRS) and the Global Dynamic Hyper Round Scheme (GDHRS). While GRBP is a "time-driven" scheduling policy, HCSP is a "node-driven" policy that schedules the clustering-task according to the network nodes' requirements. In other words, during the network lifetime, HCSP limits the clustering region by reducing the number of engaged nodes in clustering. HCSP also controls the reconfiguration frequency by postponing the clustering operations as much as possible. Consequently, the energy dissipation of the network nodes is significantly reduced, and therefore, the worst drawback of clustering approaches, i.e. clustering overhead, is mitigated. The main contributions in this paper are as follows:

- This is the first LDSRS-based work which decreases the overload caused by periodic global clustering in each global fixed-length round by employing local clustering in each Local Super Round (LSR). In other words, instead of triggering global clustering at fixed durations of time, LDSRS schedules local clustering only when necessary. According to this approach, the target cluster protects its

P. Neamatollahi, S. Abrishami, M. Naghibzadeh, and M. H. Yaghmaee are with the Department of Computer Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, PO Box 91775-1111, Mashhad, Iran. (e-mail: neamatollahi@mail.um.ac.ir, {s-abrishami, naghibzadeh, hy-aghmae}@um.ac.ir).

Ossama Younis, Ph.D., is Chief Scientist of Smart Streets, LLC (USA), a company that provides novel technology solutions for transportation systems (e-mail: ossyounis@yahoo.com).

structure from being changed during each LSR. Nevertheless, for load balancing goals, the nodes of each cluster participate in the reclustering process before the current LSR's termination. As a result, instead of reclustering the entire network, only a part is engaged in executing the clustering process. Global clustering, however, is carried out to refresh the clustered network at the end of every Global Hyper Round (GHR) according to GDHRS. A GHR is dynamically defined to include some LSRs. Therefore, the clustering-task is hierarchically scheduled by splitting the network lifetime into GHRs, LSRs, rounds, etc.

- GRBP schedules the clustering-task in a time-driven manner by assuming that the nodes are always consuming energy. Hence, with GRBP, clustering protocols are often implemented for continuous data delivery scenarios. In contrast, a clustering protocol utilizing HCSP can be employed for every data delivery scenario (data-driven, event-driven, or query-driven). The reason for this adaptability is that HCSP dynamically determines the time of re-clustering based on the condition of network nodes.
- As HCSP schedules clustering-task independent of the clustering algorithm, it is compatible with existing clustering protocols, such as LEACH [14] and HEED [11]. HCSP can thus improve their performance in terms of network lifetime and energy conservation. However, for more energy retention and higher compatibility with HCSP, a Distributed Multi-Criteria Clustering (DMCC) protocol is proposed which employs HCSP for its clustering-task scheduling. In combination with a score, DMCC uses residual energy to elect CHs. For each node, the score is obtained from a combination of the degree and the centrality factor (described in Section V) of that particular node among its adjacent nodes. In each area, the nodes with a greater residual energy and score have a higher chance of being picked as CH. Besides, in its proximity, every regular node finds the CH with the highest score to connect to.

The rest of the paper is organized as follows: Section II states the problem that we study in this work. In Section III, the related works are reviewed. Section IV describes HCSP, and presents a simple clustering-task scheduling algorithm and its analysis. In Section V, the DMCC algorithm is described. Section VI performs the energy consumption calculations by using an energy dissipation model. Section VII describes the simulation experiments and results. Finally, Section VIII concludes this paper and provides directions for future work.

II. Problem Statement

In many WSN applications, such as habitat monitoring, sensor nodes are quasi-stationary and are densely deployed in the field. These location-unaware (without GPS) sensor nodes are left unattended after deployment; hence, their battery recharge is impossible. As a result, minimizing energy consumption to maximize the network lifetime is a priority.

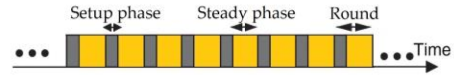


Fig. 1. The timeline (phases) of a round.

The cluster lifetime and clustering region are two concepts used in this study. The cluster lifetime is defined as the time interval from cluster creation until re-clustering is triggered (assuming this will lead to a change in cluster members). Furthermore, the definition of a clustering region is a set of clusters engaged in a triggered clustering operation.

In many applications, sensor nodes are considered homogeneous, i.e. all of them have similar processing and communication capabilities and are equally significant. In this case, every network node should play either the regular role or the CH role. Due to its heavier workload in comparison with that of a regular node, the CH node consumes significantly more energy. To lengthen the network life cycle, the CHs workload should be distributed among all homogeneous sensor nodes. Consequently, a rotating CH role among the sensor nodes is an acceptable choice for load balancing. In dynamic clustering approaches, CH role rotation is applied by re-clustering. To perform clustering, GRBP schedules a global clustering-task to be triggered once per round. As a result, according to this policy, the cluster life cycle is often bounded to one round. In GRBP, global clustering is performed at the beginning of a round, i.e. the setup phase (observe Fig. 1). Then, the clustered sensor nodes carry out the assigned tasks during the steady phase. Due to the short cluster lifetime in GRBP, the load is acceptably balanced on the network nodes. However, the dissipation of a significant amount of energy in the setup phase, i.e. clustering overhead, is considered as a serious disadvantage of GRBP usage. The reason is that the clustering region for GRBP expands to all clusters while it can be limited to only critical clusters whose CHs have been exhausted. In contrast, if cluster lifetime increases too much, the CHs will soon die from being heavily loaded. There is a trade-off between improving load balancing and reducing the clustering overhead, a subject which is addressed in this work.

III. Related Works

LEACH [14] is a well-known fully distributed clustering protocol that does not require global network knowledge for the selection of CHs. In this protocol, CH selection is performed by using a random function. The extreme simplicity and ease of employment are the main advantages of this protocol. However, when a node is picked as a CH, it informs all nodes by means of broadcasting a network-wide announcement message which is an energy consuming task. Many researches have been proposed based on LEACH (such as [15], [16]).

HEED [11] is different from LEACH in the way CHs are chosen and in the manner in which information is sent to the sink. The combination of the nodes' residual energy and the intra-cluster communication cost is used in HEED's cluster formation process. The data are sent to the sink by using multi-hop paths created on the CHs, which consume less energy in comparison with that of direct communication by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2757606, IEEE Transactions on Industrial Informatics

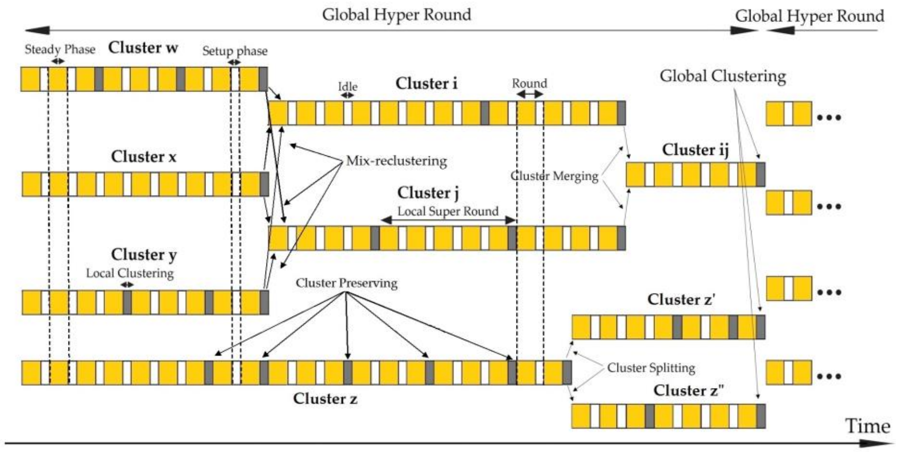


Fig. 2. An example for HCSP.

LEACH. EHEED [17] extends the HEED by multi-hop intra-cluster transmissions.

PEACH [18] is another clustering protocol that uses the overhearing characteristics of wireless communication to decrease the clustering overhead. Therefore, the method employed by PEACH is completely different from HCSP in which the clustering-task is scheduled. In PEACH, no criterion related to nodes' potentials is employed to select CHs. Besides, it seriously suffers from packet collisions due to cluster overlapping in practice.

The authors in [19] presented DHCR which is an energy-aware clustering protocol. In this protocol, the CHs are selected based on some criteria such as the node's residual energy and distance to the sink.

EDIT [6] presents a clustering algorithm that requires the minimum number of messages exchanged among sensor nodes. This low-cost clustering protocol mainly selects CHs according to the nodes' residual energy. In special cases, it also considers other metrics, such as the nodes' distance from the sink.

The common disadvantage of most clustering protocols is high energy consumption from performing global clustering in all rounds, according to GRBP. This issue is the current study's motivation.

IV. Hierarchical Clustering-task Scheduling Policy

For scheduling the clustering-task, this section presents a new policy, namely HCSP, which operates as a node-driven instead of a time-driven GRBP. A simple algorithm is proposed for HCSP implementation and the calculations of LSR length variation are also presented.

As explained earlier, LDSRS and GDHRS are used to apply HCSP. As shown in Fig. 2, the network lifecycle in HCSP is hierarchically divided into GHRs so that each GHR often includes some LSRs for each cluster. In turn, each LSR is split into some rounds. In addition, each round is further separated into two phases: the steady phase and setup phase. In contrast to GRBP, according to the LSR concept, HCSP places the setup phase at the end of the round. Clustering process, if any, are performed in the setup phase.

In comparison with global clustering, because less communication cost is involved in a local split, HCSP employs LDSRS for handling local clusterings and controlling LSRs during the lifetime of the sensor network. In each setup phase, it is possible that some clusters participate in local clustering while no clustering is carried out by the others according to LDSRS. Therefore, the clustering region is limited to the sensor nodes belonging to these participant clusters. In other words, after the setup phase, the new LSRs may be started for the nodes of some clusters while other clusters continue with their current LSRs. In each cluster, the clustering-task is only triggered at the end of the cluster's current LSR, i.e. the last round's setup phase of its current LSR. At each setup phase, in order to participate in data transmissions to the sink, the clusters that were not reconfigured are left idle until the time when the newly formed clusters are established. While in the steady phase, the selected CHs of the newly formed clusters participate with the other CHs in transmitting the sensed data to the data collection center. The LSR length can differ from one cluster to another. In addition, the LSR duration of a cluster can dynamically vary during the network life cycle depending on the node requirements of that cluster (e.g. see LSRs in Fig. 2). Note that some rounds include no local clustering in their setup phase and so the structure of the clusters is not reconstructed in these rounds. This property leads to further decrease in the clustering overhead.

For the CH role rotation in local clustering, there are two possible solutions: (1) static clustering with only the CH role rotating among cluster members, and (2) dynamic clustering which reconfigures the structure of the clusters. To further explain, when a dynamic clustering solution is employed, the sensor nodes belonging to each contributing cluster freely participate in local clustering. Thus, as shown in Fig. 2, each of the following events may occur:

- **Cluster preserving:** In this case, local clustering only leads to changing the CH node while the cluster members remain the same and the cluster lifetime is extended to a new LSR.
- **Cluster splitting:** There is little possibility that a reconfigured cluster is split into two or more clusters during the setup phase.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2757606, IEEE Transactions on Industrial Informatics

- **Cluster merging:** It is also possible that several adjacent clusters concurrently finish their LSRs and hold local clustering. In this case, the nodes belonging to these clusters may be merged into a new cluster.
- **Mix-reclustering:** In the case when several neighboring clusters simultaneously trigger local clustering, the clusters' members, and even the number of clusters, may change.

After a duration of time, cluster splitting and mix-reclustering can increase the CHs' workload due to irregular growth in the number of the clusters and the data routes created on the CHs. To refresh the network organization, global clustering should be performed at the end of every GHR consisting of some LSRs for every cluster. This global clustering can be held after a constant number of rounds has elapsed or when either one of the CHs or the sink recognizes that the number of clusters exceeds the boundary of a pre-specified range. In another solution, the CHs monitor the variations of LSRs. When the LSR length in a region is very short in contrast to that of other areas, this problem may be solved by triggering a global clustering which is scheduled by the CHs. Besides, the nodes' mobility may be considered as a metric for determining GHR length. Each of these approaches can be utilized by negotiations among the CHs via the message passing. Nevertheless, the number of global clusterings should be limited to hold down the clustering and scheduling overhead. On the other hand, the length of the GHR can dynamically vary based on the network conditions. In doing so, if a CH node discovers that the global clustering condition has been satisfied, it then informs other CHs by propagating a GC-msg among its neighbor CHs. This message is forwarded to others by the receiver CHs. Then, the CHs announce to their cluster members that the global clustering will be triggered at the upcoming setup phase. By receiving this announcement message, every regular node participates in the global clustering process at the beginning of the next setup phase.

A. A Simple Algorithm

To implement HCSP, this section presents a simple distributed algorithm which only pays attention to the CH nodes' energy consumption for LDSRS. As the decision for triggering local clustering is made without any regard to the energy information of the cluster members, no message is exchanged among the relative CH and its members at this stage. Accordingly, the overhead of this algorithm is negligible.

After deployment of the sensor nodes, there is a global clustering to configure the nodes in a hierarchical structure. In the setup phase of a particular round, reclustering may or may not be triggered. If triggered, reclustering is mostly performed locally and only clusters with an exhausted CH (tired clusters) are reorganized to form fresh clusters. In contrast, the clusters with a CH having the adequate energy (half-bored clusters) remain without any modification. To achieve this, each CH node belonging to a cluster (say cluster c) saves its residual energy in a variable named $E^c_{CH\_init}$ at the end of the clustering process. In turn, each new CH node computes its residual energy threshold value, denoted by E^c_{th}, in order to hold local clustering at the necessary time:

$$E^c_{th} = LCF \times E^c_{CH\_init}. \quad (1)$$

In (1), LCF $(0 < LCF < 1)$ is the local clustering factor determined as a constant value with respect to the application's requirements. During the steady phase, every CH is vigilant about its residual energy. If the CH's residual energy falls below its E^c_{th}, that CH then sends a message, called LC_msg, to its members at the end of the current steady phase. By transmitting this message, the CH informs its members to perform reclustering during the setup phase of the current round, i.e. the end of current LSR. Therefore, both the CH election and cluster formation are accomplished locally.

Occasionally, the number of clusters increases due to cluster splitting and mix-reclustering events of LDSRS. Every CH transmits the summary messages obtained from its cluster members via a multi-hop path toward the sink. As a result, the increase in the number of clusters leads to the growth of data routes over some CHs. This reduces the longevity of these CHs, which is an issue that can be mitigated by a global clustering. Therefore, limiting the number of clusters is taken into account as an approach selected for implementing GDHRS. In doing so, each CH, in turn, computes the threshold of its received summary message number, denoted by N^c_{th}, as:

$$N^c_{th} = GCF \times N^c_{rsm\_init}, \quad (2)$$

in which $N^c_{rsm\_init}$ indicates the initial number of received summary messages from outer clusters and GCF $(GCF \geq 1)$ is the global clustering factor determined by an application similar to LCF. For example, if the summary message size in an application is large, the GCF must be a small value to avoid overloading some CHs with the forwarding of extra summary messages. Accordingly, whenever a CH finds that the number of its received summary messages has grown higher than its threshold, it sends a GC_msg to neighboring CHs for triggering an immediate global clustering. To inform all clusters, the receiver CHs (if needed) transmit this message to the others. Consequently, global clustering is performed at the upcoming setup phase which will create fresh clusters for the next GHR (observe Fig. 3).

B. LSR Calculations

The theoretical analysis provided in this section addresses the LSR length variation. First, some notations are defined. $E^c_{CH}(r)$ is defined as the residual energy of cluster c's CH node at the end of the steady phase of the r-th round. In addition, e^c_{CH} indicates cluster c's CH energy consumption during the steady phase of a round. Because the e^c_{CH} value largely depends on the structure of the network in cluster c, it can be assumed as fixed during the LSR of a cluster. Consider that a CH has the residual energy of $E^c_{CH}(r)$ and $E^c_{CH}(r+1)$ at the end of round r's steady phase, and that of immediate round $r+1$, respectively. If r and $r+1$ are in a LSR, then:

$$E^c_{CH}(r+1) = E^c_{CH}(r) - e^c_{CH}. \quad (3)$$

Now, it is vital to understand how long local clustering is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2757606, IEEE Transactions on Industrial Informatics

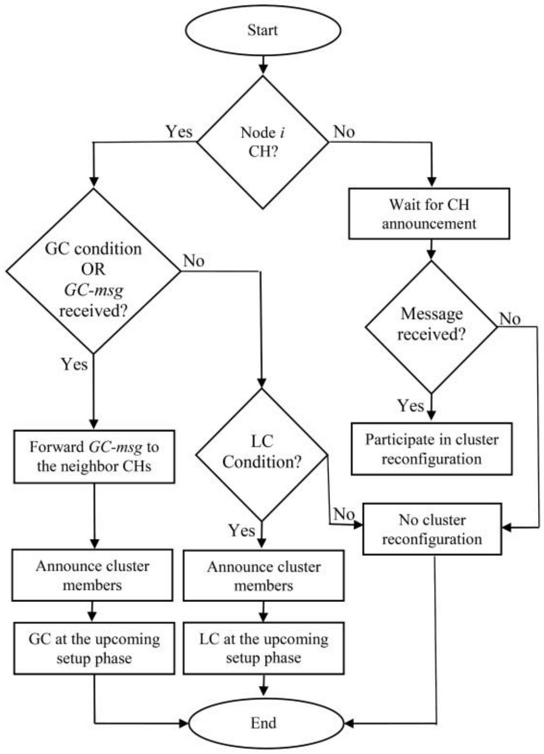


Fig. 3. HCSP overview.

triggered in a cluster using LDSRS. As shown in Fig. 4, N_{LSR}^{c} denotes the length of the current LSR in cluster c according to the number of rounds. With respect to the LDSRS algorithm, consider that a sensor node has been selected as the CH of cluster c in round r. If this CH node satisfies the following condition in the steady phase of round $(r+N_{LSR}^{c}-1)$, then local clustering will be accomplished in the setup phase of this round. According to (1), this indicates:

$$E_{th}^{c}=LCF\times E_{CH\_init}^{c}(r)\geq E_{CH}^{c}(r+N_{LSR}^{c}-1). \quad (4)$$

Using (3) and (4), we conclude that:

$$LCF\times E_{CH\_init}^{c}(r)\geq E_{CH\_init}^{c}(r)-N_{LSR}^{c}\times e_{CH}^{c}. \quad (5)$$

Therefore,

$$N_{LSR}^{c}\geq\frac{(1-LCF)\times E_{CH\_init}^{c}(r)}{e_{CH}^{c}}. \quad (6)$$

Since N_{LSR}^{c} is the lowest integer number that satisfies (6), therefore:

$$N_{LSR}^{c}=\left\lceil\frac{(1-LCF)\times E_{CH\_init}^{c}(r)}{e_{CH}^{c}}\right\rceil. \quad (7)$$

Based on (7), the N_{LSR}^{c} value depends on the parameter value of LCF, $E_{CH\_init}^{c}(r)$, and also e_{CH}^{c}. The parameter value of LCF is considered constant during analysis. With respect to network density and the application's requirements, this parameter can be simply tuned to maximize energy resource usage. A large LCF decreases the LSRs' length and, as a result, the number of local clusterings increases. This situation is suitable for applications in which load balancing is a critical requirement. In contrast, a small value for LCF leads to an increase in the LSR length. This is appropriate for applications whose clustering overhead is high. As e_{CH}^{c} is also assumed to be fixed during a LSR, the amount of $E_{CH\_init}^{c}(r)$ directly affects the amount of N_{LSR}^{c}. The residual energy of sensor nodes decreases as time passes; thus, the amount of N_{LSR}^{c} also

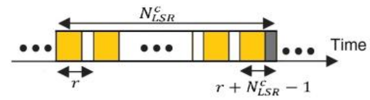


Fig. 4. LSR notations in cluster c.

decreases. In the worst case, N_{LSR}^{c} will be one, which leads to the clusters' reorganization in every round, similar to GRBP. This may occur in the final rounds of the network lifetime.

V. Distributed Multi-Criteria Clustering (DMCC) Protocol

This section presents DMCC, a distributed dynamic clustering protocol for a homogeneous WSN. In addition to making the performance evaluation of HCSP possible as the primary aim, the other goals of presenting this clustering protocol are to: (1) satisfy the application's requirements by splitting the network nodes into clusters, (2) reduce node energy dissipation via the selection of the appropriate CH nodes, and (3) prolong the network lifetime through lengthening LSRs and GHRs.

In contrast to most existing clustering protocols, the main novelty of DMCC is the use of an efficient multi-criteria scoring function (described in Section V. A) to select the CHs and to form the clusters. In addition, DMCC (in contrast to many popular clustering protocols) selects the CHs only based on their abilities without using any random function. The other benefit of the proposed protocol is the low communication overhead for cluster formation. DMCC does not make any particular assumptions about nodes capabilities, e.g. location awareness or the distribution of nodes. The focus here is on continuous monitoring applications, such as industrial monitoring, in which the user needs the most recent values sensed by the network nodes [20]. More specifically, the setup phase of DMCC includes preparation and clustering processes when the clustering is triggered. The duration of the DMCC protocol setup phase is fixed, which means it does not depend on node deployment. On the other hand, the steady phase is split into frames. In each frame, a regular node senses the environment and sends the sensed data to the related CH; then, the data is aggregated by that CH and forwarded to the sink. In DMCC, CHs can benefit from a routing protocol, such as power-aware routing [21], to find inter-cluster paths for accessing the sink in a multi-hop fashion. In the following sections, the preparation and clustering processes will be explained in detail.

A. Preparation Process

Neighbor discovery is performed at the beginning of the preparation process to obtain the latest information from adjacent sensor nodes about their new relative locations (if there is some mobility) and life status (if a node's battery is discharged). Afterward, each node computes its own score as:

$$score(v)\leftarrow\begin{cases}\sum_{i=1}^{n=num\ of\ neighbors}\frac{1}{1+\log(dist(v,i)+1)} & n>0\\ 0 & n=0\end{cases} \quad (8)$$

in which $dist(v,i)$ denotes the distance between node v and node i. Note that each node can compute its distance from neighbors by the received signal strength method, as in [14].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2757606, IEEE Transactions on Industrial Informatics

In (8), two metrics are implicitly combined in order to prolong the network lifetime via the reduction of energy consumption. These metrics are the node degree (the number of adjacent nodes in a cluster range) and node centrality (the centrality of the node among its neighbors). The lower the value of the neighbors' distance from node v, the higher the value of $score$-(v). This signifies that the location of node v is more central among its adjacent nodes. In addition to decreasing the interference of radio waves between CHs and the cost of communication of a CH node with its members, the use of this feature in CH election may prevent cluster splitting; hence, the GHRs' length may also be extended. In addition, the greater the number of neighbor nodes for node v, the higher the value of $score(v)$. In other words, node v is more eligible to become a CH and so dense clusters are organized. In this way, some benefits are easily attainable, such as using data correlation, eliminating data redundancy, and prolonging the GHRs' length (due to decreasing the number of clusters). Note that, to consider the network variations, the preparation process "may not" need to be performed whenever reclustering is triggered.

B. Clustering Process

The pseudo code of the clustering process is depicted in Algorithm 1. During this clustering process, a node can become a candidate CH, a deterministic CH, or a regular node joining another CH. For each node, S_{det_CH} includes all neighbor deterministic CHs while S_{can_CH} is defined as a set of all candidate CHs in the neighborhood, from the first iteration to the current iteration of *while loop*. Since prolonging network longevity is the principal goal of presenting the current paper, candidate CHs are primarily selected based on the nodes' residual energy. As a result, each node determines its waiting time (T_{wait}) proportional to the inverse of its residual energy at the beginning of this process. For a node, the lower the waiting time the higher the chance of being CH. $T_{clu-pro}$ is the time threshold which limits the number of iterations of the *while loop* to O(1). When either $T_{clu-pro}$ is expired or the status of the node which is executing the code is determined, i.e. regular node or deterministic CH, that node relinquishes executing the *loop*.

Firstly, a node with a greater amount of energy (with short T_{wait}) has a higher chance of becoming a candidate CH. If a node becomes a candidate CH, it broadcasts its new status to the nodes in its cluster range (Lines 11-13). In the next iterations, if this node has the highest score among the other candidate CHs in its vicinity, its status will turn to a deterministic CH. Afterwards, this node broadcasts a CH_msg to the adjacent nodes in its cluster range (Lines 8-10) and abandons the process. Accordingly, nodes with a higher amount of energy can terminate this process earlier. Whenever a node receives a CH_msg from a deterministic CH, it is turned into a regular node and can no longer become a CH in the current reclustering (Lines 6-7). Therefore, this regular node has to select one of the deterministic CHs in its vicinity based on the *score* of that deterministic CH via Lines 19-21. If a node has executed the *while loop* several times but has not yet received a CH_msg from a deterministic CH, it will finally leave the execution of *while loop* while its status is still unknown. This node then introduces itself as a deterministic CH via the execution of Lines 22-25. It is worth mentioning that each deterministic or candidate CH can only send one announcement message in this process.

Algorithm 1. Clustering process at node i.

```
1.  T_wait ∝ 1/E_res
2.  status ← UNKNOWN
3.  S_det_CH ← Ø
4.  S_can_CH ← Ø
5.  WHILE (status = UNKNOWN AND t < T_clu−pro) DO
6.    IF (S_det_CH ≠ Ø)
7.    |    status ← regular_node
8.    ELSE IF (S_can_CH ≠ Ø AND t > T_wait AND ID = MOST_SCORE (S_can_CH))
9.    |    CH_msg (ID, det_CH, score)
10.   |    status ← deterministic_CH
11.   ELSE IF (t > T_wait AND ID ∉ S_can_CH)
12.   |    CH_msg (ID, can_CH, score)
13.   END_IF
14.   Upon Receiving CH_msg (ID, det_CH, score)
15.   |    S_det_CH = S_det_CH ∪ {(ID, score)}
16.   Upon Receiving CH_msg (ID, can_CH, score)
17.   |    S_can_CH = S_can_CH ∪ {(ID, score)}
18. END_WHILE
19. IF (status = regular_node)
20. |    my_CH ← MOST_SCORE (S_det_CH)
21. |    JOIN_CLUSTER (my_CH, ID)
22. ELSE IF (status = UNKNOWN)
23. |    CH_msg (ID, det_CH, score)
24. |    status ← deterministic_CH
25. END_IF
```

VI. Energy Consumption Analysis

In this section, the energy consumption model is first presented and then the average amount of energy consumed for every sensor node during a GHR by DMCC is calculated.

A. Sensor Node Energy Consumption Model

The same model in [2]–[5], [8] is utilized here for the energy dissipation of radio hardware. The energy spent to send a message, denoted as $E_{tx}(l,d)$, is computed with both the message length l and the transmitter node's distance to receiver d. That is:

$$E_{tx}(l,d) = \begin{cases} l \times E_{elec} + l \times \epsilon_{fs} \times d^2, & if\ d < d_0 \\ l \times E_{elec} + l \times \epsilon_{mp} \times d^4, & if\ d \geq d_0 \end{cases}, \quad (9)$$

where E_{elec} is the electronic energy, $\epsilon_{fs} \times d^2$ and $\epsilon_{mp} \times d^4$ are the amplifier energies, and d_0 is a distance threshold between the sending and receiving nodes, over which multipath fading (i.e., d^4 power loss) is used as a channel model; otherwise, the free-space model (i.e., d^2 power loss) is assumed. However, the consumed energy for receiving a message, denoted as $E_{rx}(l)$, depends only on the message size. Then:

$$E_{rx}(l) = l \times E_{elec}. \quad (10)$$

B. Energy Consumption in a GHR duration

To compute the amount of energy expended in an HCSP-enabled WSN, this section presents a simple mathematical model. Assume N nodes have been uniformly scattered in a $S \times S$ area [11], [20] and the total number of regular nodes is N_{re}, whereas the number of CHs is denoted as N_{CH}. In each cluster, there are N/N_{CH} number of nodes on the average, including one CH node and $N/N_{CH}-1$ regular nodes. Considering these assumptions, the average amount of node energy consumption during a GHR, including the steady phase, the clustering message propagation stage, and the setup phase, will be computed as follows.

Energy Consumption in the Steady Phase: The computations of this phase are performed based on the continuous data delivery scenarios. Therefore, in every Time Division Multiple Access (TDMA) frame's time length, denoted as T_{fr}, each node senses its area once and periodically relays the sensed data to the associated CH. Then, each CH aggregates the raw data received from its cluster members in order to transfer them to the sink. To explain more in detail, $T_{fr} = (N/N_{CH})T_{dt}$, where T_{dt} represents the time slot duration for sensing the environment and transmitting a data message of the size l_{dm} by a sensor node. Each regular node is only awake during its particular time slot; thus, the value of energy dissipated by a specific regular node during T_{fr} is:

$$E_{re} = \left(T_{fr} - T_{dt}\right)E_{sl} + E_{tx}\left(l_{dm}, d_{re\_CH}\right), \qquad (11)$$

where d_{re_CH} is the distance between the member node and its associated CH ($d_{re\_CH} \leq R_c$) and E_{sl} is the average amount of energy used by a sensor node per unit time in the sleep state. The respective CH of each cluster generates a summary message at the end of each frame (the energy spent is E_{da} per bit). Then, this message is forwarded to the next CH along with a multi-hop route. In addition to its own message, each CH node relays the number of $N_{in\_sum\_msg}$ incoming summary messages from farther clusters. Accordingly, the total number of $N_{in\_sum\_msg}+1$ messages is forwarded by a CH which is denoted by $N_{out\_sum\_msg}$. Consequently, during a TDMA frame, the energy wasted by a CH node is:

$$E_{CH} = \left(\frac{N}{N_{CH}} - 1\right)E_{rx}(l_{dm}) + \left(\frac{N}{N_{CH}}\right)l_{dm}E_{da} + N_{in\_sum\_msg}E_{rx}(l_{dm}) + N_{out\_sum\_msg}E_{tx}\left(l_{dm}, d_{CH\_CH}\right), (12)$$

in which d_{CH_CH} demonstrates the distance between one CH to the next CH in the multi-hop path toward the sink. As a consequence, the total energy spent for the network's operation in a steady phase is:

$$E_{steady} = N_{fr}(N_{re}E_{re} + N_{CH}E_{CH}), \quad (13)$$

where N_{fr} is the number of frames in the steady phase of a round.

Energy Consumption in the Clustering Message Propagation Stage: The amount of energy spent to propagate an *LC_msg* in a cluster, denoted by E_{LC_msg}, is computed as:

$$E_{LC\_msg} = E_{tx}(l_{cm}, R_c) + \left(\frac{N}{N_{CH}} - 1\right)E_{rx}(l_{cm}), \qquad (14)$$

where l_{cm} denotes the control message length. The first element of (14) is the energy consumed by the CH for distributing the *LC_msg* and the second one is the amount of energy dissipated for receiving this message by the other member nodes.

As explained in Section IV, the GHR length is determined on-demand and online in a special event. In the final round of the GHR, by propagating *GC_msg* along multi-hop routes, the critical cluster's CH informs the other CHs that the global clustering conditions have been met. Then, each CH makes an announcement to its member nodes. Therefore:

$$E_{GC\_msg} = N_{CH}E_{tx}\left(l_{cm}, d_{CH\_CH}\right) + (N_{CH}-1)E_{rx}(l_{cm}) + N_{CH}E_{tx}(l_{cm}, R_c) + N_{re}E_{rx}(l_{cm}). \quad (15)$$

Energy Consumption in the Setup Phase: In the clustering event, the setup phase is divided into three stages: 1) preparation; 2) clustering; and 3) CH schedules. In the following, the energy expended in each stage will be separately computed.

1) *Preparation Stage:* In this stage, every contributor sensor node transmits a *Hello* message within its cluster radius to other nodes in its vicinity while it receives similar messages from its neighbors. Hence,

$$E_{pre}(N_{CN}) = N_{CN}\left(E_{tx}(l_{cm}, R_c) + N_{CN}^{R_c}E_{rx}(l_{cm})\right), (16)$$

where N_{CN} represents the total number of contributor nodes and $N_{CN}^{R_c}$ is the average number of contributor sensor nodes located in the range of the *Hello* message's sender. The N_{CN} is limited to the number of nodes in the clustering region for local clustering while this number is enlarged to the total number of network nodes in global clustering event.

2) *Clustering Stage*: At the beginning of the clustering algorithm, some candidate CHs are selected. Then, all chosen candidate CHs announce the sensor nodes located in their cluster radius. In the following, some of the candidate CHs turn into deterministic CHs and announce to their neighbors again. Accordingly,

$$E_{ann}(N_{CN}) = (N_{CCA} + N_{CCH})\left(E_{tx}(l_{cm}, R_c) + N_{CN}^{R_c}E_{rx}(l_{cm})\right), \qquad (17)$$

in which N_{CCA} and N_{CCH} denote the number of contributor candidate and deterministic CHs, respectively. It is worth mentioning that $N_{CCH} \leq N_{CH}$, but only whenever a global clustering is triggered: $N_{CCH} = N_{CH}$. The first element in (17) is related to the energy wasted by all the contributor candidate and deterministic CHs when sending all advertisements. Also, the second element of this equation is the energy dissipated for receiving the messages by all participant nodes in the sender's cluster range. On the other hand, by using a *join_cluster* message, the N_{cre} number of contributor regular nodes can join to their new clusters. Therefore, the average value of energy expended in the cluster join step is expressed as:

$$E_{join}(N_{CN}) = N_{cre}E_{tx}\left(l_{cm}, d_{re\_CH}\right) + N_{CCH}E_{rx}(l_{cm}). \quad (18)$$

Hence, the total amount of energy spent at this stage is:

$$E_{clu}(N_{CN}) = E_{ann}(N_{CN}) + E_{join}(N_{CN}). \quad (19)$$

3) *Node Scheduling Stage*: In this stage, each CH transmits a message to schedule its assigned regular nodes. Consequently, the average amount of energy spent in the node scheduling stage is calculated as:

$$E_{sch}(N_{CN}) = N_{CCH}E_{tx}(l_{cm}, R_c) + N_{cre}E_{rx}(l_{cm}). \quad (20)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2757606, IEEE Transactions on Industrial Informatics

TABLE II
SIMULATION PARAMETERS

PARAMETER	VALUE
ϵ_{fs}	10 pJ/bit/m^2
ϵ_{mp}	0.0013 pJ/bit/m^4
E_{elec}	50 nJ/bit
E_{DA}	5 nJ/bit/signal
Idle power	13.5 mW
Sleep power	15 µW
Threshold distance (d_0)	87.7 m
Initial energy per node	[2, 4] Jules
Round time	20 sec
Round	5 frame
Data packet size	100 byte
Control packet size	25 byte
C_{min}	10^{-4}
LCF	0.8
GCF	1.5

At last, using (16) to (20), the average amount of energy wasted for clustering is:

$$E_{setup}(N_{CN}) = E_{pre}(N_{CN}) + E_{clu}(N_{CN}) + E_{sch}(N_{CN}). \quad (21)$$

As a consequence, this value in each local clustering is calculated as:

$$E_{setup}^{LC} = E_{setup}\left(\frac{N}{N_{CH}}\right). \quad (22)$$

However, in the global clustering event, it is obtained as:

$$E_{setup}^{GC} = E_{setup}(N). \quad (23)$$

In other words, according to the first assumption:

$$E_{setup}^{LC} = \frac{E_{setup}^{GC}}{N_{CH}}. \quad (24)$$

Total Energy Consumption*:* The total amount of energy expended by the network nodes during a GHR, E_{GHR}, is calculated as:

$$E_{GHR} = N_{GHR}E_{steady} + N_{LC}^{GHR}\left(E_{LC\_msg} + E_{setup}^{LC}\right) + E_{GC\_msg} + E_{setup}^{GC}, \quad (25)$$

in which N_{GHR} is the average number of rounds in every GHR and N_{LC}^{GHR} denotes the average number of local clusterings in a GHR. Then, the average amount of energy dissipated for each round with HCSP is:

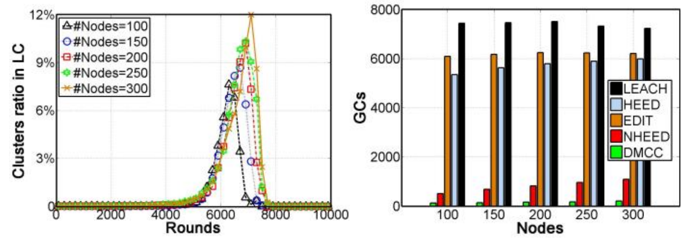
$$E_{round}^{HCSP} = \frac{E_{GHR}}{N_{GHR}}. \quad (26)$$

Finally, we compute the average energy consumed by every sensor node for every round in the WSN when HCSP is adopted, i.e.:

$$E_{sensor}^{HCSP} = \frac{E_{round}^{HCSP}}{N}. \quad (27)$$

VII. PERFORMANCE EVALUATION

In this section, the experimental setup is first explained. This section then compares the efficiency of DMCC with that of LEACH [14], HEED [11], EDIT [6], and NHEED by a simulation in MATLAB [2]–[4], [7] for all five protocols. To evaluate the performance of DMCC in comparison to other protocols, in addition to simulating the three clustering protocols of LEACH, HEED, and EDIT, a new version of the HEED protocol (proposed in this paper), called NHEED, has been simulated. This protocol is the same as HEED, with the exception that its clustering-task scheduling operation is not based on GRBP. To reduce the clustering overhead, NHEED performs global clustering when the residual energy of a CH falls below its power threshold. More precisely, when LDSRS recommends local clustering, the network nodes are globally reconfigured by NHEED. In other words, NHEED substitutes the local clusterings used in LDSRS with global clusterings. As a result, global clustering is not performed in every round which decreases the clustering energy overhead. However, it takes only one CH with inadequate energy to force all nodes in the network to perform global clustering.



(a) (b)

Fig. 5. (a) Ratio of clusters involved in local clustering during DMCC operation. (b) Total number of global clusterings.

A. Experimental Setup

In the simulations, the total number of nodes is set at 100, 150, 200, 250, and 300. A network with nodes deployed randomly over a square area the size of 100×100 m^2 (between (0, 0) and (100, 100)) is considered. The sink is located outside the supervised area at coordinates (50 m, 175 m), similar to [2], [20]. The parameters for the simulation and node energy model are summarized in Table II. It is worth mentioning that simulations were run 50 times and the average was reported for each different setup in order to consider the randomness of the sensor nodes' placement.

B. Simulation Results

Fig. 5(a) illustrates the ratio of clusters involved in local clustering for the total network lifetime. As illustrated in this figure, the average ratio of clusters involved in local clustering increases as the network lifetime expands. Because the nodes' residual energy diminishes as time passes, the CHs reach their energy threshold more quickly. Consequently, more clusters should be reorganized during a fixed time interval. However, at the end of the network lifetime, the average ratio of clusters in local clustering reduces due to a decrease in the number of live nodes. Thus, this reduction in the number of local clusterings continues until all nodes in the network die. This situation is remarkable in other figures. As shown in Fig. 5, the greatest average is less than 12%. This signifies that, in the worst case, an average 12% of clusters in each round are involved in local clustering; this is more energy-efficient than when all clusters are reorganized, as happens in the LEACH, HEED, EDIT, and NHEED protocols.

Fig. 5(b) shows the number of global clusterings triggered during the network lifetime by the simulated clustering protocols. As illustrated in this figure, the number of global clusterings in HEED, EDIT, and LEACH is much more than that of NHEED and DMCC, since those three protocols perform global clustering in every round. In contrast, NHEED does not perform global clustering at every round and DMCC requires a lower number of global clusterings than NHEED

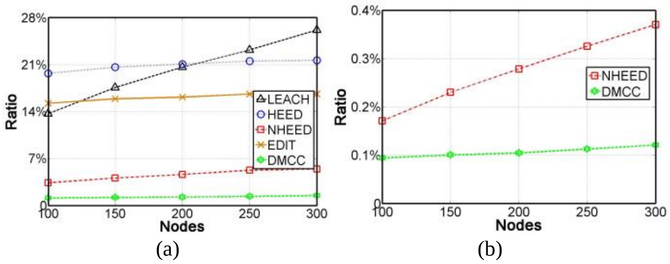


Fig. 6. (a) The ratio of energy consumed in the setup phases to total energy consumption. (b) The ratio of energy consumed in the clustering-task scheduling to total energy consumption.

does, because DMCC performs local clustering based on LDSRS which reduces the need for global clustering.

Fig. 6(a) illustrates the ratio of the energy consumed in the setup phases to the total energy dissipation (i.e. clustering overhead) in different simulated protocols. Because NHEED triggers global clustering based on LDSRS, there is less clustering energy overhead imposed on the nodes than with the LEACH, HEED, and EDIT protocols. Nevertheless, the HCSP-based DMCC often takes advantage of local clustering in contrast to NHEED. This results in DMCC's having the least energy consumption overhead from the setup phase when compared to other clustering protocols. Therefore, the worst drawback of the clustering approaches applied in WSNs, i.e. clustering overhead, is mitigated by using HCSP.

Fig. 6(b) depicts the clustering-task scheduling overhead. As clustering is statically scheduled in LEACH, HEED, and EDIT, these protocols have been removed from this experiment. In DMCC, a clustering-task scheduling overhead is imposed on the sensor nodes from the *GC_msg*s and *LC_msg*s propagation based on HCSP. Nevertheless, as shown in Fig. 6(b), the cost is negligible (less than 0.15% of the total energy consumption) because HCSP tremendously reduces the number of reclustering. However, NHEED bears a higher scheduling energy overhead because it propagates more number of *GC_msg*s.

The network lifetime of the simulated protocols is evaluated by the two metrics introduced in [2], [6], [22]: FND (First Node Dies) and HNA (Half of Nodes Alive). Fig. 7 plots the protocols' network longevity in different numbers of nodes. The round number for each metric is summarized in this figure. This figure indicates that the presented protocol typically outperforms the other simulated clustering protocols in every definition of the network lifetime mainly because of employing HCSP. Besides, the comparison of the metrics for different number of nodes shows that, with the enhancement in the number of nodes, the DMCC's network longevity does not fall into a descending order. Consequently, DMCC is a scalable clustering protocol.

Fig. 8 illustrates lifetime (FND) in relation to the distance of the sink node from the center of the sensing region (the total number of nodes is 100 for this experiment). When this distance is 0, the location of the sink is at the center of the sensing region. In this case, the amount of energy spent in the steady phase is lower than that of other sink locations because the energy dissipation of transmitter nodes depends on their distance from the sink. Therefore, the ratio of energy consumption in the steady phase to total energy consumption is reduced. As a result, the effectiveness of the setup phase, in

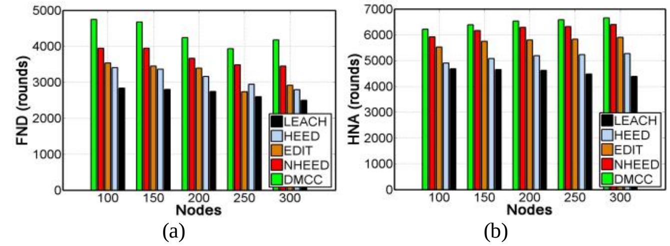


Fig. 7. Network lifetime. (a) FND. (b) HNA.

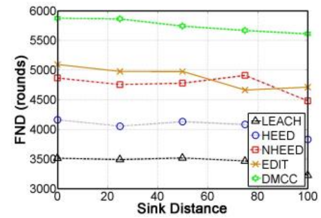


Fig. 8. The sink distance from the center of sensing area.

terms of energy consumption, is greater. In this case, due to the energy conservation features of HCSP in the setup phase, DMCC is the most efficient and operates much better than the other simulated protocols. As shown in Fig. 8, even with an increase in the sink distance from the center, the DMCC performance still surpasses the others.

VIII. Conclusion

This paper introduces a clustering-task scheduling policy, HCSP, which achieves remarkable energy savings while extending the network life cycle of clustering protocols. This policy controls the cluster lifetime and clustering region during the network lifetime. Although HCSP is compatible with different types of data delivery models and clustering protocols, the current study also introduces a clustering protocol, DMCC, which is suitable for continuous data delivery scenarios and improves upon the energy conservation properties of HCSP. Comparing to a HCSP-based protocol (such as DMCC), the GRBP-based protocols (such as LEACH, HEED, and EDIT) are simpler, however, the GRBP is not a perfect scheduling policy. Simulation results demonstrate that the predominance of DMCC over the other simulated clustering protocols is significant in regard to the network life cycle and energy savings. In contrast to when using the costly GRBP for scheduling the clustering-task, the data collection protocol of choice is rendered more flexible, energy-efficient, and scalable when it employs HCSP. Although HCSP is proposed for quasi-stationary sensor nodes, future work can explore the extension of this policy for handling mobile sensor nodes via a variation in clustering-task scheduling metrics.

References

[1] M. M. Afsar and M.-H. Tayarani-N, "Clustering in sensor networks: A literature survey," *J. Netw. Comput. Appl.*, vol. 46, pp. 198–226, Nov. 2014.

[2] M. Tarhani, Y. S. Kavian, and S. Siavoshi, "SEECH: Scalable Energy Efficient Clustering Hierarchy Protocol in Wireless Sensor Networks," *IEEE Sens. J.*, vol. 14, no. 11, pp. 3944–3954, Nov. 2014.

[3] M. Zhao, Y. Yang, and C. Wang, "Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks," *IEEE Trans. Mob. Comput.*, vol. 14, no. 4, pp. 770–785,

2015.
[4] S. H. Kang and T. Nguyen, "Distance Based Thresholds for Cluster Head Selection in Wireless Sensor Networks," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1396–1399, Sep. 2012.
[5] M. Elhoseny, X. Yuan, Z. Yu, C. Mao, H. K. El-Minir, and A. M. Riad, "Balancing Energy Consumption in Heterogeneous Wireless Sensor Networks using Genetic Algorithm," *IEEE Commun. Lett.*, vol. 7798, no. c, pp. 1–1, 2014.
[6] A. Thakkar and K. Kotecha, "Cluster head election for energy and delay constraint applications of wireless sensor network," *IEEE Sens. J.*, vol. 14, no. 8, pp. 2658–2664, 2014.
[7] V. Pal, G. Singh, and R. P. Yadav, "Balanced Cluster Size Solution to Extend Lifetime of Wireless Sensor Networks," *IEEE Internet Things J.*, vol. 2, no. 5, pp. 399–401, 2015.
[8] A. Y. Zomaya, W. Li, F. C. Delicato, and A. Y. Zomaya, "Adaptive energy-efficient scheduling for hierarchical wireless sensor networks," *ACM Trans. Sens. Networks*, vol. 9, no. 3, pp. 1–34, 2013.
[9] J. Ren, Y. Zhang, K. Zhang, A. Liu, J. Chen, and X. S. Shen, "Lifetime and Energy Hole Evolution Analysis in Data-Gathering Wireless Sensor Networks," *IEEE Trans. Ind. Informatics*, vol. 12, no. 2, pp. 788–800, Apr. 2016.
[10] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, no. 14–15, pp. 2826–2841, Oct. 2007.
[11] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mob. Comput.*, vol. 3, no. 4, pp. 366–379, Oct. 2004.
[12] X. Zhu, L. Shen, and T.-S. Yum, "Hausdorff Clustering and Minimum Energy Routing for Wireless Sensor Networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 990–997, Feb. 2009.
[13] Ruitao Xie and Xiaohua Jia, "Transmission-Efficient Clustering Method for Wireless Sensor Networks Using Compressive Sensing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 806–815, Mar. 2014.
[14] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wirel. Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.
[15] I. Sohn, J.-H. Lee, and S. H. Lee, "Low-Energy Adaptive Clustering Hierarchy Using Affinity Propagation for Wireless Sensor Networks," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 558–561, Mar. 2016.
[16] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003, vol. 3, pp. 1713–1723.
[17] M. R. Senouci, A. Mellouk, H. Senouci, and A. Aissani, "Performance evaluation of network lifetime spatial-temporal distribution for WSN routing protocols," *J. Netw. Comput. Appl.*, vol. 35, no. 4, pp. 1317–1328, 2012.
[18] S. Yi, J. Heo, Y. Cho, and J. Hong, "PEACH: Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks," *Comput. Commun.*, vol. 30, no. 14–15, pp. 2842–2852, Oct. 2007.
[19] M. Sabet and H. R. Naji, "A decentralized energy efficient hierarchical cluster-based routing algorithm for wireless sensor networks," *AEU - Int. J. Electron. Commun.*, vol. 69, no. 5, pp. 790–799, May 2015.
[20] N. Bouabdallah, M. E. Rivero-Angeles, and B. Sericola, "Continuous Monitoring Using Event-Driven Reporting for Cluster-Based Wireless Sensor Networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3460–3479, Sep. 2009.
[21] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 1998, pp. 181–190.
[22] P. Neamatollahi, H. Taheri, E. Toreini, M. Naghibzadeh, and M. H. Yaghmaee, "A novel fuzzy metric to evaluate clusters for prolonging lifetime in wireless sensor networks," in *2011 International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 2011, pp. 118–123.

Peyman Neamatollahi received the BS and MS degrees in computer engineering, with a concentration in parallel and distributed systems, in 2007 and 2011, respectively. He received the PhD degree in computer engineering from the Ferdowsi University of Mashhad in 2017. He has published several conference and journal papers. His research interests are in Wireless Sensor Network (WSN) protocols and applications, Internet of Things (IoT), job scheduling in distributed environments, cloud computing, and fuzzy logic control. He also serves as a reviewer of the Ad Hoc Networks, the Journal of Networks and Computer Applications, Computers and Electrical Engineering, Wireless Networks, and the Journal of Supercomputing.

Saeid Abrishami received the MSc and PhD degrees in computer engineering from Ferdowsi University of Mashhad, in 1999 and 2011, respectively. Since 2003, he has been with the Computer Engineering Department, Ferdowsi University of Mashhad, where he is currently an assistant professor. During the spring and summer of 2009 he was a visiting researcher at the Delft University of Technology. His research interests focus on resource management and scheduling in the distributed systems, especially in Grids and Clouds.

Mahmoud Naghibzadeh (SM'12) received MS and PhD degrees in Computer Science and Computer Engineering, respectively, from University of Southern California, USC-USA. He has taught Undergraduate and Graduate courses at USC and University of South Florida, USA. He is now a full professor at Ferdowsi University of Mashhad, Iran. He was a visiting professor at University of California-Irvine, UCI-USA, in 1991 and a visiting professor at Monash University, Australia, in 2003-2004. His research interests include scheduling aspects of real-time systems, Grid, Cloud, Multiprocessors, Multicores, and GPGPUs and also Bioinformatics algorithms, especially Genomics and Proteomics. He has published numerous papers as well as eight books and has been the chairman of two international conferences and technical chair of many others. He is the reviewer of many journals and a member of many computer societies and a senior member of IEEE. He is the recipient of many awards including MS and PhD study scholarship.

Mohammad Hossein Yaghmaee Moghaddam (SM'12) received his B.S. degree from Sharif University of Technology, Tehran, Iran in 1993, and M.S. and Ph.D degrees in communication engineering from Tehran Polytechnic (Amirkabir) University of Technology in 1995 and 2000, respectively. November 1998 to July1999, he was with Network

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2757606, IEEE Transactions on Industrial Informatics

Technology Group (NTG), C&C Media research labs. NEC corporation, Tokyo, Japan, as visiting research scholar. September 2007 to August 2008, he was with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, USA as the visiting associate professor. July 2015 to September 2016, he was with the electrical and computer engineering department of the University of Toronto (UoT) as the visiting professor. Currently, he is a full professor at the Computer Engineering Department, Ferdowsi University of Mashhad (FUM). His research interests are in Smart Grid Communication, Software Defined Networking (SDN), Cloud Computing and Internet of Things (IoT).

Ossama Younis, Ph.D., is Chief Scientist of Smart Streets, LLC (USA), a company that provides novel technology solutions for transportation systems (since May 2016). Dr. Younis has received his M.S. and Ph.D. degrees from Purdue University (USA) in 2003 and 2005, respectively. He has worked as a research scientist at the University of Arizona (USA) 2005-2007. From 2007 to 2014, he was a senior research scientist at Applied Communication Science (AppComSci). In 2013, he worked as a professor of information technology at Berkeley College, NJ (USA). He was also a computer scientist at the National Institute of Standards and Technology (NIST) in USA (2014-2016). Dr. Younis has twenty years of experience in computer systems and sensor networks application research (more recently in cyber-physical systems). He also has about forty-four publications in prestigious journals and conferences. He has lead U.S. Navy SBIR research projects (2011-2013).