Chapter 12

Clustering in Wireless Sensor Networks

Basilis Mamalis, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou

CONTENTS

The use of wireless sensor networks (WSNs) has grown enormously in the last decade, pointing out the crucial need for scalable and energy-efficient routing and data gathering and aggregation protocols in corresponding large-scale environments. Hierarchical clustering protocols (as opposed to direct single-tier communication schemes) have extensively been used toward the above directions. Moreover, they can greatly contribute to overall system scalability, lifetime, and energy efficiency. In this chapter the state of the art in corresponding hierarchical clustering approaches for large-scale WSN environments is presented. The need for clustering in WSNs is first motivated and a brief description of the implied hierarchical network pattern is given. The basic advantages, objectives, and design challenges are also briefly explored. A set of appropriate taxonomy parameters as well as a global classification scheme is then introduced. In the main body of the chapter the most significant of the existing WSN clustering algorithms are concisely presented and commented according to the previously stated parameters and classification scheme. The chapter is concluded by stating some general remarks as well as some open research issues in the field.

12.1 Introduction

In most wireless sensor network (WSN) applications nowadays the entire network must have the ability to operate unattended in harsh environments in which pure human access and monitoring cannot be easily scheduled or efficiently managed or it's even not feasible at all [1]. Based on this critical expectation, in many significant WSN applications the sensor nodes are often deployed randomly in the area of interest by relatively uncontrolled means (i.e., dropped by a helicopter) and they form a network in an ad hoc manner [2,3]. Moreover, considering the entire area that has to be covered, the short duration of the battery energy of the sensors and the possibility of having damaged nodes during deployment, large populations of sensors are expected; it's a natural possibility that hundreds or even thousands of sensor nodes will be involved. In addition, sensors in such environments are energy constrained and their batteries usually cannot be recharged. Therefore, it's obvious that specialized energy-aware routing and data gathering protocols offering high scalability should be applied in order that network lifetime is preserved acceptably high in such environments.

Naturally, grouping sensor nodes into *clusters* has been widely adopted by the research community to satisfy the above scalability objective and generally achieve high energy efficiency and prolong network lifetime in large-scale WSN environments. The corresponding hierarchical routing and data gathering protocols imply cluster-based organization of the sensor nodes in order that data fusion and aggregation are possible, thus leading to significant energy savings. In the hierarchical network structure each cluster has a leader, which is also called the cluster head (CH) and usually performs the special tasks referred above (fusion and aggregation), and several common sensor nodes (SN) as members.

The cluster formation process eventually leads to a two-level hierarchy where the CH nodes form the higher level and the cluster-member nodes form the lower level. The sensor nodes periodically transmit their data to the corresponding CH nodes. The

CH nodes aggregate the data (thus decreasing the total number of relayed packets) and transmit them to the base station (BS) either directly or through the intermediate communication with other CH nodes. However, because the CH nodes send all the time data to higher distances than the common (member) nodes, they naturally spend energy at higher rates. A common solution in order balance the energy consumption among all the network nodes, is to periodically re-elect new CHs (thus rotating the CH role among all the nodes over time) in each cluster. A typical example of the implied hierarchical data communication within a clustered network (assuming single-hop intracluster communication and multi-hop intercluster communication) is further illustrated in Figure 12.1.

The BS is the data processing point for the data received from the sensor nodes, and where the data is accessed by the end user. It is generally considered fixed and at a far distance from the sensor nodes. The CH nodes actually act as gateways between the sensor nodes and the BS. The function of each CH, as already mentioned, is to perform common functions for all the nodes in the cluster, like aggregating the data before sending it to the BS. In some way, the CH is the sink for the cluster nodes, and the BS is the sink for the CHs. Moreover, this structure formed between the sensor nodes, the sink (CH), and the BS can be replicated as many times as it is needed, creating (if desired) multiple layers of the hierarchical WSN (multi-level cluster hierarchy).

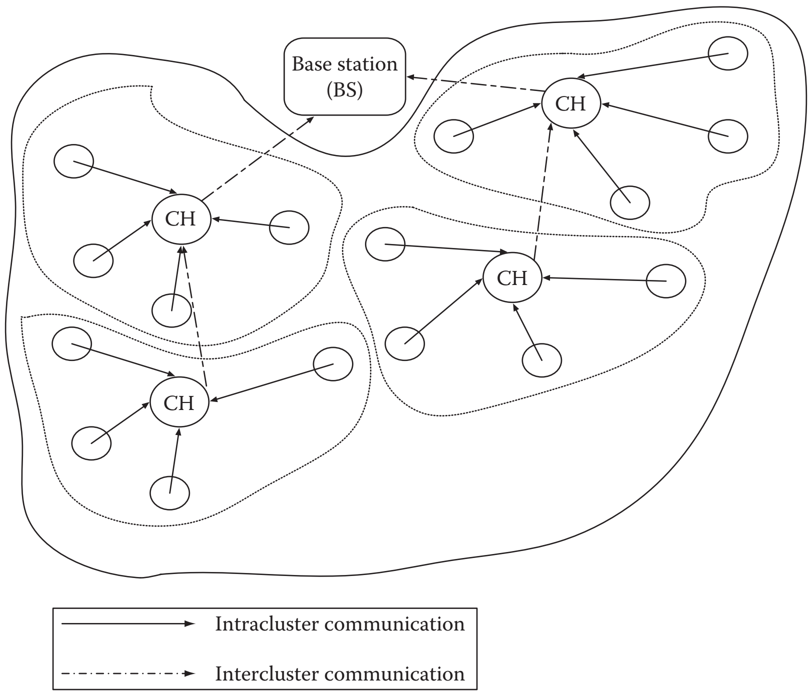

Figure 12.1 Data communication in a clustered network.

12.1.1 Main Objectives and Design Challenges of Clustering in WSNs

As was mentioned at the beginning, hierarchical clustering in WSNs can greatly contribute to overall system scalability, lifetime, and energy efficiency. Hierarchical routing is an efficient way to lower energy consumption within a cluster, performing data aggregation and fusion in order decrease the number of transmitted messages to the BS. On the contrary, a single-tier network can cause the gateway to overload with the increase in sensors density. Such overload might cause latency in communication and inadequate tracking of events. In addition, the single-tier architecture is not scalable for a larger set of sensors covering a wider area of interest because the sensors are typically not capable of long-haul communication. Hierarchical clustering is particularly useful for applications that require scalability to hundreds or thousands of nodes. Scalability in this context implies the need for load balancing and efficient resource utilization. Applications requiring efficient data aggregation (e.g., computing the maximum detected radiation around a large area) are also natural candidates for clustering. Routing protocols can also employ clustering [9,27]. In Ref. [50], clustering was also proposed as a useful tool for efficiently pinpointing object locations.

In addition to supporting network scalability and decreasing energy consumption through data aggregation, clustering has numerous other secondary advantages and corresponding objectives [1]. It can localize the route setup within the cluster and thus reduce the size of the routing table stored at the individual node. It can also conserve communication bandwidth because it limits the scope of intercluster interactions to CHs and avoids redundant exchange of messages among sensor nodes. Moreover, clustering can stabilize the network topology at the level of sensors and thus cuts on topology maintenance overhead. Sensors would care only for connecting with their CHs and would not be affected by changes at the level of inter-CH tier. The CH can also implement optimized management strategies to further enhance the network operation and prolong the battery life of the individual sensors and the network lifetime. A CH can schedule activities in the cluster so that nodes can switch to the low-power sleep mode and reduce the rate of energy consumption. Furthermore, sensors can be engaged in a round-robin order and the time for their transmission and reception can be determined so that the sensors reties are avoided, redundancy in coverage can be limited, and medium access collision is prevented.

WSNs also present several particular *challenges* in terms of design and implementation. Similar challenges and design goals have also been faced earlier in the field of mobile ad hoc networks (MANETs), and naturally a lot of related ideas (considering clustering protocols etc.) have been borrowed from that field. In WSNs, however (in which the support of mobility even if it's applicable, it's not critical), the limited capabilities (battery power, transmission range, processing hardware and memory used, etc.) of the sensor nodes combined with the special location-based conditions met (not easily accessed in order recharge the batteries or replace the entire sensors) make the energy efficiency and the scalability factors even more crucial. Moreover, the challenge of prolonging network lifetime under the above restrictions is difficult to be met by using only traditional techniques. Consequently, it becomes unavoidable to follow alternative techniques (i.e., see

Section 12.3) leading to more efficient protocols with a lot of differences compared to the ones designed for MANETs.

Beyond the typical (however vital) challenges mentioned above (limited energy, limited capabilities, network lifetime) some additional important considerations in the design process of clustering algorithms for WSNs should be the following: Cluster formation: The CH selection and cluster formation procedures should generate the best possible clusters (well balanced, etc.). However they should also preserve the number of exchanged messages low and the total time complexity should (if possible) remain constant and independent to the growth of the network. This yields a very challenging trade-off. *Application Dependency*: When designing clustering and routing protocols for WSNs, application robustness must be of high priority and the designed protocols should be able to adapt to a variety of application requirements. Secure communication: As in traditional networks, the security of data is naturally of equal importance in WSNs too. The ability of a WSN clustering scheme to preserve secure communication is ever more important when considering these networks for military applications. Synchronization: Slotted transmission schemes such as TDMA allow nodes to regularly schedule sleep intervals to minimize energy used. Such schemes require corresponding synchronization mechanisms and the effectiveness of this mechanisms must be considered. Data aggregation: Because this process makes energy optimization possible it remains a fundamental design challenge in many sensor network schemes nowadays. However its effective implementation in many applications is not a straightforward procedure and has to be further optimized according to specific application requirements.

12.2 Classification of Clustering Algorithms

12.2.1 Clustering Parameters

Before documenting on the possible classification options of WSNs clustering algorithms as well as on the algorithms themselves in more details, it is worth reporting on some important parameters with regard to the whole clustering procedure in WSNs. These parameters also serve as the basic means for further comparison and categorization of the presented clustering protocols throughout this chapter.

- Number of clusters (cluster count). In most recent probabilistic and randomized clustering algorithms the CH election and formation process lead naturally to variable number of clusters. In some published approaches, however, the set of CHs are predetermined and thus the number of clusters are preset. The number of clusters is usually a critical parameter with regard to the efficiency of the total routing protocol.
- Intracluster communication. In some initial clustering approaches the communication between a sensor and its designated CH is assumed to be direct (one-hop communication). However, multi-hop intracluster communication is often (nowadays) required, i.e., when the communication range of the sensor nodes is limited or the number of sensor nodes is very large and the number of CHs is bounded.

- Nodes and CH mobility: If we assume stationary sensor nodes and stationary CHs we are normally led to stable clusters with facilitated intracluster and intercluster network management. On the contrary, if the CHs or the nodes themselves are assumed to be mobile, the cluster membership for each node should dynamically change, forcing clusters to evolve over time and probably need to be continuously maintained.
- Nodes types and roles: In some proposed network models (i.e., heterogeneous environments) the CHs are assumed to be equipped with significantly more computation and communication resources than others. In most usual network models (i.e., homogeneous environments) all nodes have the same capabilities and just a subset of the deployed sensors are designated as CHs.
- Cluster formation methodology: In most recent approaches, when CHs are just regular sensors nodes and time efficiency is a primary design criterion, clustering is being performed in a distributed manner without coordination. In few earlier approaches a centralized (or hybrid) approach is followed; one or more coordinator nodes are used to partition the whole network off-line and control the cluster membership.
- Cluster-head selection: The leader nodes of the clusters (CHs) in some proposed algorithms (mainly for heterogeneous environments) can be preassigned. In most cases however (i.e., in homogeneous environments), the CHs are picked from the deployed set of nodes either in a probabilistic or completely random way or based on other more specific criteria (residual energy, connectivity etc.).
- Algorithm complexity. In most recent algorithms the fast termination of the executed protocol is one of the primary design goals. Thus, the time complexity or convergence rate of most cluster formation procedures proposed nowadays is constant (or just dependent on the number of CHs or the number of hops). In some earlier protocols, however, the complexity time has been allowed to depend on the total number of sensors in the network, focusing in other criteria first.
- Multiple levels. In several published approaches the concept of a multi-level cluster hierarchy is introduced to achieve even better energy distribution and total energy consumption (instead of using only one cluster level). The improvements offered by multi-level clustering are to be further studied, especially when we have very large networks and inter-CH communication efficiency is of high importance.
- Overlapping. Several protocols give also high importance on the concept of node overlapping within different clusters (either for better routing efficiency or for faster cluster formation protocol execution or for other reasons). Most of the known protocols, however, still try to have minimum overlap only or do not support overlapping at all.

According to the above parameters, we then try to introduce and further compare most of the algorithms presented in this chapter. This brief initial presentation is given in Table 12.1. The reader should refer to this table in combination with the global classification scheme given in the next section to gain a more clear view of the presented algorithms.

Table 12.1 Comparison of the Presented Clustering Algorithms

Clustering Approaches	*Time Complexity*	*Node Mobility*	*Cluster Overlap*	*In-Cluster Topology*	*Cluster Count*	*Clustering Process*	*CHs Selection*	*CHs Rotation*	*Multi Level*
LBC [5]	N/A	No	No	1-hop	Fixed	Centralized	Preset	No	No
MSNDP [6]	N/A	No	No	1-hop	Variable	Centralized	Preset	No	No
LCA [7]	Variable	Possible	No	1-hop	Variable	Distributed	ID-based	No	No
AC [9]	Variable	Yes	No	1-hop	Variable	Distributed	ID-based	No	No
DCATT [10]	N/A	No	No	1-hop	Fixed	Manual	Preset	No	No
LEACH [11]	Constant	Limited	No	1-hop	Variable	Distributed	Prob/random	Yes	No
EEHC [13]	Variable	No	No	*k*-hop	Variable	Distributed	Prob/random	Yes	Yes
HEED [14]	Constant	Limited	No	1-hop	Variable	Distributed	Prob/energy	Yes	No
LEACHC [12]	N/A	Limited	No	1-hop	Variable	Centralized	Prob/random	Yes	No
TLEACH [15]	Constant	Limited	No	1-hop	Variable	Distributed	Prob/random	Yes	Yes
MOCA [16]	Constant	Limited	Yes	*k*-hop	Variable	Distributed	Prob/random	Yes	No
TCCA [17]	Variable	No	No	*k*-hop	Variable	Distributed	Prob/energy	Yes	No
EECS [18]	Constant	No	No	1-hop	Constant	Distributed	Prob/energy	Yes	No
EEMC [19]	Variable	No	No	*k*-hop	Variable	Distributed	Prob/energy	Yes	Yes
RCC [21]	Variable	Yes	No	*k*-hop	Variable	Hybrid	Random	No	No

(*continued*)

Table 12.1 (continued) Comparison of the Presented Clustering Algorithms

Clustering Approaches	*Time Complexity*	*Node Mobility*	*Cluster Overlap*	*In-Cluster Topology*	*Cluster Count*	*Clustering Process*	*CHs Selection*	*CHs Rotation*	*Multi Level*
CLUBS [22]	Variable	Possible	Yes	2-hop	Variable	Distributed	Random	No	No
FLOC [23]	Constant	Possible	No	2-hop	Variable	Distributed	Random	No	No
RECA [24]	Constant	No	No	1-hop	Variable	Distributed	Random	Yes	No
HCC [27]	Variable	Possible	Yes	*k*-hop	Variable	Distributed	Connectivity	No	Yes
HC [28]	Variable	Possible	No	1-hop	Variable	Distributed	Connectivity	No	No
MMDC [29]	Variable	Yes	No	*k*-hop	Variable	Distributed	Connectivity	No	No
EEDC [30]	Variable	No	No	1-hop	Variable	Centralized	Connectivity	No	No
CAWT [31]	Constant	No	No	2-hop	Variable	Distributed	Connectivity	No	No
EACLE [32]	Variable	No	No	2-hop	Variable	Distributed	Proximity	Yes	No
ACE [33]	Constant	Possible	Yes	*k*-hop	Variable	Distributed	Connectivity	No	No
WCA [38]	Variable	Yes	No	1-hop	Variable	Distributed	Weight-based	No	No
DWEHC [39]	Constant	No	No	*k*-hop	Variable	Distributed	Weight-based	No	No
TASC [40]	Variable	No	No	2-hop	Variable	Distributed	Weight-based	No	No
GS3 [25]	Variable	Possible	Yes	*k*-hop	Constant	Distributed	Preset	No	No
GROUP [26]	Variable	No	No	*k*-hop	Controlled	Hybrid	Proximity	No	No

12.2.2 Taxonomy of Clustering Protocols

There have been several different ways (based directly on the above-mentioned parameters or not) to initially distinguish and further classify the algorithms used for WSNs clustering, [4]. Two of the most early and common classifications in the bibliography are (i) clustering algorithms for homogeneous or heterogeneous networks and (ii) centralized or distributed clustering algorithms.

The first of the above classifications is based on the characteristics and functionality of the sensors in the cluster, whereas the other one is based on the method used to form the cluster. In *heterogeneous* sensor networks (i.e., [6,10]), there are generally two types of sensors, sensors with higher processing capabilities and complex hardware (used generally to create some sort of backbone inside the WSN—being preset as the CH nodes—and also serve as data collectors and processing centers for data gathered by other sensor nodes), and common sensors, with lower capabilities, used to actually sense the desired attributes in the field. In homogeneous networks, all nodes have the same characteristics, hardware and processing capabilities (i.e., this is the typical case when the sensors are deployed in battle fields). In this case (which is the most usual in nowadays applications) every sensor can become a CH. Moreover, the CH role can be periodically rotated among the nodes in order achieve better load balancing and more uniform energy consumption.

Also, when all the nodes have the same capabilities (homogeneous environments), a distributed CH election and formation process is the most appropriate technique to gain increased flexibility and fast execution-convergence times independent of the number of nodes of the WSN. There are also a few approaches using *centralized* or hybrid techniques (i.e., [5,6,12]—where one or more coordinator nodes or the BS is responsible to partition the whole network off-line and control the cluster membership), however they are naturally not suitable for practical general-purpose large-scale WSNs applications (they may be suitable only for special purpose limited-scale applications where high-quality connectivity and network partitioning is required). Here we mainly focus on distributed (which are the most efficient, especially for large networks) clustering protocols for homogeneous environments (which are the most general purpose and widely used nowadays).

Another common classification is between *static* and dynamic clustering. A cluster formation procedure is regarded as dynamic (otherwise as static) when it includes regular (periodic or event driven) CH reelection or cluster reorganization procedures, either to effectively react to network topology changes and adjust appropriately the cluster topology, or simply aiming at the appropriate rotation of the CH role among the nodes to gain in energy efficiency. Dynamic cluster architectures make a better use of the sensors in a WSN and naturally lead to improved energy consumption management and network lifetime.

Most of the known clustering algorithms for WSNs can be further distinguished into two main categories (as presented in details in Sections 12.3–12.4), depending on cluster formation criteria and parameters used for CH election:

- Probabilistic (random or hybrid) clustering algorithms
- Nonprobabilistic clustering algorithms

In the category of probabilistic selection clustering algorithms [11–24], a priori probability assigned to each sensor node is used to determine the initial CHs (or some other type random election procedure is scheduled). The probabilities initially assigned to each node often serve as the primary (random) criterion in order for the nodes to decide individually on their election as CHs (in a flexible, uniform, fast and completely distributed way); however other secondary criteria may also be considered either during CH election process (i.e., the residual energy) or during the cluster formation process (i.e., the proximity or the communication cost) in order achieve better energy consumption and network lifetime. Beyond the high energy efficiency (which is facilitated also from the periodic CH reelection scheme usually adopted), the clustering algorithms of this category usually achieve faster execution/convergence times and reduced volume of exchanged messages.

In the category of nonprobabilistic clustering algorithms [25–43], more specific (deterministic) criteria for CH election and cluster formation are primarily considered, which are mainly based [25–36] on the nodes' proximity (connectivity, degree, etc.) and on the information received from other closely located nodes. The cluster formation procedure here is mainly based on the communication of nodes with their neighbors (one or multi-hop neighbors) and generally requires more intensive exchange of messages and probably graph traversing in some extent, thus leading sometimes to worse time complexity than probablistic/random clustering algorithms. On the contrary these algorithms are usually more reliable toward the direction of extracting robust and well-balanced clusters. In addition to node proximity, some algorithms [37–40] also use a combination of metrics such as the remaining energy, transmission power, mobility, etc. (forming corresponding combined weights) to achieve more generalized goals than single-criterion protocols. In the same category we also address a relatively new and quite challenging class of clustering algorithms for WSNs, namely, the biologically inspired protocols [41–43] (based on swarm intelligence) which are probably the most promising alternative approaches for clustering in WSNs nowadays.

Furthermore, in Section 12.5, we refer separately to a special-purpose class of clustering protocols, those that are suitable for Reactive Networks [44–49]. These protocols have clearly different objectives compared to the most common category of proactive clustering algorithms to which all the other above-mentioned protocols belong. They are specifically oriented to applications with timing restrictions and usually take advantage of user queries for the sensed data or of specific triggering events that occur in the WSN. It should also be noted that throughout this chapter the concept of sensor nodes or CHs mobility has not been considered in any particular way, as the number of applications that require mobile nodes are still considerably rare; also there is not much specialized work in the literature till now. The reader may find some relevant information and specific related work in Refs. [51] and [52].

Finally, before the detailed presentation of the main clustering categories introduced above, we will refer briefly to the former protocols used for clustering in WSNs (i.e., even before the last decade). The first clustering algorithms for WSNs were naturally inspired from (or entirely based on) corresponding algorithms already studied and used in the field of wired sensor networks or, later, in the field of mobile ad hoc networks.

Uniformly assigned unique identifiers were usually the key parameter for selecting CHs in those algorithms.

One of the first such clustering algorithms (initially developed for wired sensor networks) was the Linked Cluster Algorithm (LCA—[7]). LCA was a distributed ID-based, one-hop, static clustering algorithm, trying to maximize network connectivity. The main disadvantage of LCA was that usually led to excessive number of clusters. An improved LCA-based approach (generating smaller number of clusters) was given in [8] (LCA2). Both algorithms [7,8] had limited scope as clustering algorithms for WSNs because they did not consider the problem of limited energy of WSNs. Additionally, both protocols construct one-hop clusters and their time complexity is $O(n)$ which is rather unacceptable for large size WSNs. Similarly, an early example of clustering protocols initially developed for mobile ad hoc networks and then applied also to WSNs, was the adaptive clustering algorithm presented in [9]. Other classical paradigms of clustering algorithms designed initially for MANETs, were the MAX-MIN [29], HC [28], and WCA [38] algorithms; we will briefly refer to them later (due to their specific characteristics) in Section 12.4. Finally, some of the initial clustering schemes proposed for WSNs were based on some sort of manual formation of the clusters (mostly applicable to heterogeneous environments). Such a representative case can be found in Ref. [10] (DCATT). These manual-based clustering formation schemes are not applicable to general-purpose WSNs of our days, unless specific conditions are met.

12.3 Probabilistic Clustering Approaches

As the need for efficient use of WSNs on large regions increased in the last decade dramatically, more specific clustering protocols were developed to meet the additional requirements (increased network lifetime, reduced and evenly distributed energy consumption, scalability, etc.). The most significant and widely used representatives of these focused on WSN clustering protocols (LEACH, EEHC, and HEED) and their most valuable extensions are presented in the main part of this section. They are all probabilistic in nature and their main objective was to reduce the energy consumption and prolong the network lifetime. Some of them (such as LEACH, EEHC, and their extensions) follow a random approach for CH election (the initially assigned probabilities serve as the basis for the random election of the CHs), whereas others (like HEED and similar approaches) follow a hybrid probabilistic methodology (secondary criteria are also considered during CH election—i.e., the residual energy). Some additional energy-efficient random selection approaches with good performance (like RECA) are also examined at the end of the section.

12.3.1 Popular Probabilistic Clustering Protocols

12.3.1.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

One of the first and most popular clustering protocols proposed for WSNs was LEACH (Low Energy Adaptive Clustering Hierarchy) [11,12]. It is probably the first dynamic clustering protocol which addressed specifically the WSNs needs, using homogeneous stationary sensor nodes randomly deployed, and it still serves as the basis for other improved clustering protocols for WSNs. It's an hierarchical, probabilistic, distributed, one-hop protocol, with main objectives (a) to improve the lifetime of WSNs by trying to evenly distribute the energy consumption among all the nodes of the network and (b) to reduce the energy consumption in the network nodes (by performing data aggregation and thus reducing the number of communication messages). It forms clusters based on the received signal strength and also uses the CH nodes as routers to the BS. All the data processing such as data fusion and aggregation are local to the cluster.

LEACH forms clusters by using a distributed algorithm, where nodes make autonomous decisions without any centralized control. All nodes have a chance to become CHs to balance the energy spent per round by each sensor node. Initially a node decides to be a CH with a probability "p" and broadcasts its decision. Specifically, after its election, each CH broadcasts an advertisement message to the other nodes and each one of the other (non-CH) nodes determines a cluster to belong to, by choosing the CH that can be reached using the least communication energy (based on the signal strength of each CH message). In Figure 12.2 the cluster formation scheme is given in a more clear view.

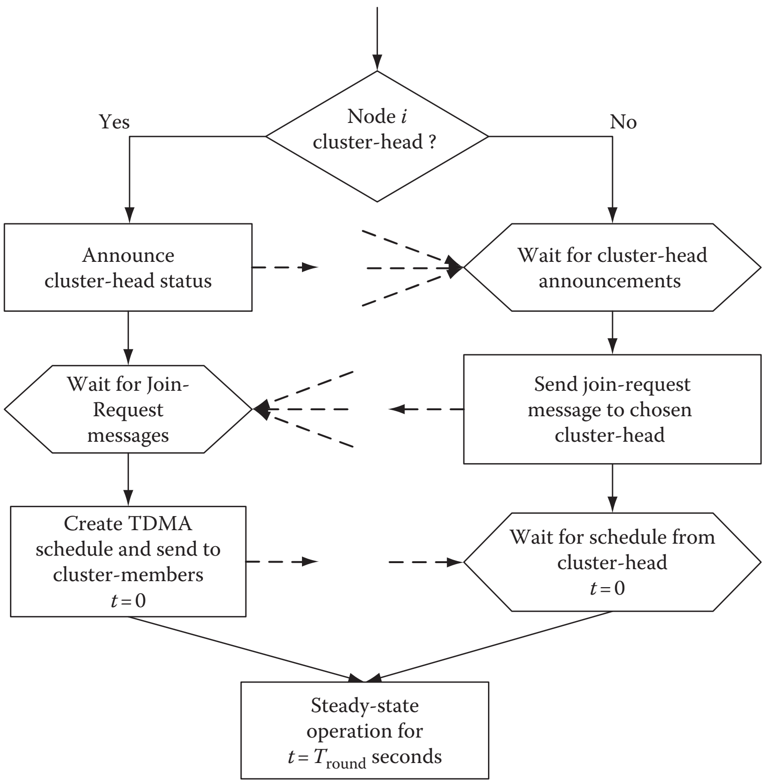

Figure 12.2 Flowchart of the cluster formation process of LEACH. (Redrawn from Heinzelman, W.B. et al., *IEEE Trans. Wireless Commun.*, 1, 660, 2002.)

The role of being a CH is rotated periodically among the nodes of the cluster to balance the load. The rotation is performed by getting each node to choose a random number "T" between 0 and 1. A node becomes a CH for the current rotation round if the number is less than the following threshold:

$$T(i) = \begin{cases} \dfrac{p}{1 - p * \left(r \bmod \frac{1}{p}\right)} & \textit{if } i \in G \\ 0 & \text{otherwise} \end{cases}$$

where
p is the desired percentage of CH nodes in the sensor population
r is the current round number
G is the set of nodes that have not been CHs in the last $1/p$ rounds

The clusters are formed dynamically in each round (through the process of Figure 12.2) and the time to perform the rounds are also selected randomly.

Generally, LEACH can provide a quite uniform load distribution in one-hop sensor networks. Moreover, it provides a good balancing of energy consumption by random rotation of CHs. Furthermore, the localized coordination scheme used in LEACH provides better scalability for cluster formation, whereas the better load balancing enhances the network lifetime. However, despite the generally good performance, LEACH has also some clear drawbacks. Because the decision on CH election and rotation is probabilistic, there is still a good chance that a node with very low energy gets selected as a CH. Due to the same reason, it is possible that the elected CHs will be concentrated in one part of the network (good CHs distribution cannot be guaranteed) and some nodes will not have any CH in their range. Also, the CHs are assumed to have a long communication range so that the data can reach the BS directly. This is not always a realistic assumption because the CHs are usually regular sensors and the BS is often not directly reachable to all nodes. Moreover, LEACH forms in general one-hop intracluster and intercluster topology where each node should transmit directly to the CHs and thereafter to the BS, thus normally it cannot be used effectively on networks deployed in large regions.

12.3.1.2 Energy-Efficient Hierarchical Clustering (EEHC)

Another significant probabilistic clustering algorithm was earlier proposed in Ref. [13] (Energy Efficient Hierarchical Clustering—EEHC). The main objective of this algorithm was to address the shortcomings of one-hop random selection algorithms such as LEACH by extending the cluster architecture to multiple hops. It is a distributed, k-hop hierarchical clustering algorithm aiming at the maximization of the network lifetime. Initially, each sensor node is elected as a CH with probability "p" and announces its election to the neighboring nodes within its communication range. The above CHs are now called the "volunteer" CHs. Next, all the nodes that are within "k"-hops distance from a "volunteer" CH, are supposed to receive the election message either directly or through intermediate forwarding. Consequently, any node that receives such CH election message and is not itself a CH, becomes a member of the closest cluster. Additionally, a number of 'forced' CHs are elected from nodes that are neither CHs nor belong to a cluster. Specifically, if the election messages do not reach a node within a preset time interval t, the node becomes a "forced" CH assuming that it is not within k hops of all volunteer CHs.

However, the most challenging feature of the EEHC algorithm is the direct extension to a corresponding multi-level clustering structure. The initial clustering process is recursively repeated at the level of CHs making it possible to build multiple levels of cluster hierarchy. Assuming that an "h"-level cluster hierarchy has been constructed in that way (with corresponding preset CH election probabilities $p_1, p_2, \cdots, p_h$ for each level), the algorithm ensures the efficient "h"-level communication between common sensor nodes and the BS, as follows (assuming that level h is the highest): Common sensor nodes transmit their collected data to the corresponding first-level (level # 1) CHs, the CHs of the first-level clusters transmit the aggregated data to the second-level CHs and so on, till the top (#h) level of the clustering hierarchy is reached; the CHs of those h-level clusters transmit their final aggregated data reports to the BS. This multi-level protocol has a time complexity of $O(k_1 + k_2 + \cdots + k_h)$, where k_i is the corresponding parameter (for each level) to the above-mentioned "k" parameter (number of hops in the basic-initial procedure). That was a significant improvement over the $O(n)$ time complexity that many of the existing algorithms till then (like LCA) had, and made this algorithm quite suitable for large networks.

Considering the overall performance of EEHC, the energy consumption for network operations (data gathering, aggregation, transmission to the BS, etc.) clearly depends on the parameters p and k of the algorithm. The authors derive mathematical expression for the values of p and k that achieve minimal energy consumption and they show via simulation results that by using the optimal parameter values energy consumption in the network can be reduced significantly. Also the simulation results validate the worth of use multiple levels (instead of single-level) of cluster hierarchy, as it is presented in Figure 12.3 for different values of communication radii r and spatial density λ.

12.3.1.3 Hybrid Energy-Efficient Distributed Clustering (HEED)

Another improved and very popular energy-efficient protocol is HEED (Hybrid Energy-Efficient Distributed Clustering [14]). HEED is a hierarchical, distributed, clustering scheme in which a single-hop communication pattern is retained within each cluster, whereas multi-hop communication is allowed among CHs and the BS. The CH nodes are chosen based on two basic parameters, residual energy and intracluster communication cost. Residual energy of each node is used to probabilistically choose the initial set of CHs. On the other hand, intracluster communication cost reflects the node degree or node's proximity to the neighbor and is used by the nodes in deciding to join a cluster or not. Thus, unlike LEACH, in HEED the CH nodes are not selected randomly. Only sensors that have a high residual energy are expected to become CH nodes. Also, the probability of two nodes within the transmission range of each other becoming CHs is small. Unlike LEACH, this means that CH nodes are well distributed in the network. Moreover, when choosing a cluster, a node will communicate with the CH that yields the lowest intracluster communication cost. In HEED, each node is mapped to exactly

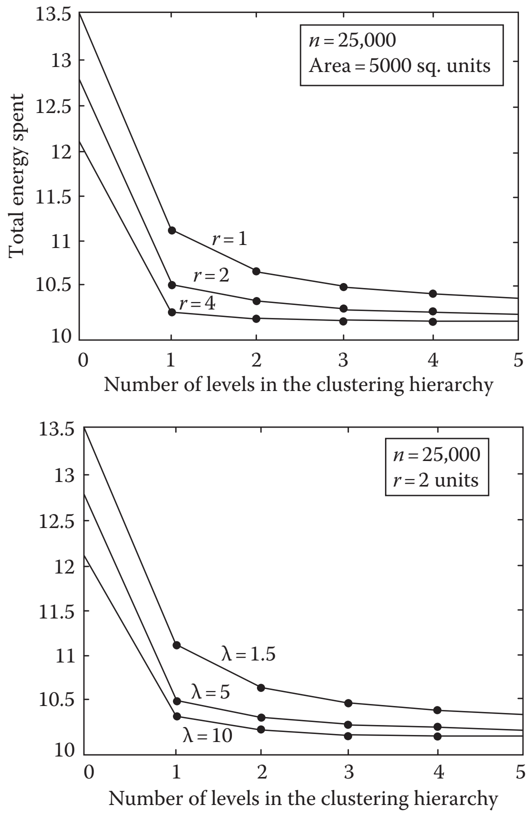

Figure 12.3 Energy consumption of multi-level EEHC. (Redrawn from Bandyopadhya, S. and Coyle, E., An energy efficient hierarchial clustering algorithm for wireless sensor network, in *22nd Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM 2003)*, San Francisco, CA, April 2003.)

one cluster and can directly communicate with its CH. Also, energy consumption is not assumed to be uniform for all the nodes. The algorithm is divided into three stages.

At the beginning, the algorithm sets an initial percentage of CHs among all sensors. This percentage value, C_{prob}, is used to limit the initial CHs announcements to the other sensors. Each sensor sets its probability of becoming a CH, CH_{prob}, as follows: $CH_{\text{prob}} = C_{\text{prob}} * E_{\text{residual}}/E_{\text{max}}$, where E_{residual} is the current energy in the sensor, and E_{max} is the maximum energy, which corresponds to a fully charged battery. CH_{prob} is not allowed to fall below a certain threshold p_{min}, which is selected to be inversely proportional to E_{max}.

The main body of the algorithm consists of a (constant) number of iterations. Every sensor goes through these iterations until it finds the CH that it can transmit to with the least transmission power (cost). If it hears from no CH, the sensor elects itself to be a CH and then sends an announcement message to its neighbors informing them about the change of status. Finally, each sensor doubles its CH_{prob} value and goes to the next iteration of this phase. It stops executing this phase when its CH_{prob} reaches 1. Therefore, there are two types of CH status that a sensor could announce to its neighbors: (a) The sensor becomes a 'tentative' CH if its CH_{prob} is less than 1 (it can change its status to a regular node at a later iteration if it finds a lower cost CH). (b) The sensor "permanently" becomes a CH if its CH_{prob} has reached 1.

At the end, each sensor makes a final decision on its status. It either picks the least cost CH or announces itself as CH. Note also that for a given sensor's transmission range, the probability of CH selection can be adjusted to ensure inter-CH connectivity.

Generally, HEED's mechanism to select the CHs and form the clusters produces a uniform distribution of cluster heads across the network through localized communications with little overhead. It also clearly outperforms LEACH with regard to the network lifetime and the desired distribution of energy consumption. However, synchronization is required and the energy consumed during data transmission for far away cluster heads is significant, especially in large-scale networks. Also, a knowledge of the entire network is normally needed to determine reliably the intracluster communication cost and configuration of those parameters might be difficult in practical world.

12.3.2 Extensions and Other Similar Approaches

On the basis of the probabilistic nature of LEACH, several other protocols were developed aiming at better energy consumption and overall performance. First, the LEACH-C and the LEACH-F protocols were proposed in Ref. [12], introducing slight modifications to the initial LEACH cluster formation procedure. LEACH-C is a centralized version of LEACH, in the sense that the responsibility of the cluster creation is transferred to the BS. Each node is initially obligated to perform a direct communication with the BS in order that a global view of the network is formed. As a result an improved cluster formation procedure is performed and a slightly better overall performance of the network is achieved. LEACH-F is also a centralized protocol and is based initially on the same global clustering scheme as in LEACH-C. The main difference lies on the fact that all clusters are fixed once when they are formed, thus reducing the overhead of cluster formation in the network. However, the above design directive prevents the use of the protocol in networks with any kind of mobility.

A valuable extension to LEACH has been proposed in Ref. [15] (two-level LEACH), where the key idea of probabilistic CH election is extended (similar to the EEHC protocol but keeping one-hop intracluster topology) to construct a two-level clustering scheme. The outer level consists of the "primary" CHs where as the inner level consists of the "secondary" CHs. The "primary" CHs in each outer-level cluster communicate directly with the corresponding "secondary" CHs and the "secondary" CHs in each inner-level cluster communicate directly with the corresponding nodes in that subcluster. Data fusion as well as communication within a cluster is performed like in LEACH, in TDMA schedules. The selection of the "primary" and the "secondary" CHs is performed also in the same way as in LEACH, by setting corresponding a priori probabilities for each node. The "primary" CHs are selected first and the "secondary" CHs are selected next from the remaining nodes. The probability to become a "primary" CH is normally less than the probability to become a "secondary" CH. Generally, the two-level clustering scheme of this algorithm achieves a significant reduction on the percentage of nodes that have to transmit data to the BS in each round. Thus, it is normally expected to reduce the total energy spent.

Also, most of the published probabilistic clustering algorithms construct "disjoint" clusters. On the contrary, in Ref. [16] the authors argue that allowing some degree of overlap among clusters can be quite effective for many tasks like intercluster routing, topology discovery and node localization, recovery from CH failure, etc. Specifically, they introduce a probabilistic (randomized), distributed Multi-hop Overlapping Clustering Algorithm (MOCA) for organizing the sensors into overlapping clusters. The goal of the clustering process is to ensure that each node is either a CH or within "k" hops from at least one CH, where k is a preset cluster radius. The algorithm initially assumes that each sensor in the network becomes a CH with probability "p." Each CH then advertises itself to the sensors within its radio range. This advertisement is forwarded to all sensors that are no more than k hops away from the CH. A node sends a request to all CHs that it heard from to join their clusters. In the join request, the node includes the ID of all CHs it heard from, which implicitly implies that it is a boundary node. The CH election probability (p) is used to control the number of clusters in the network and the degree of overlap among them. The authors also provide extensive simulation work to validate appropriate values of "p" to achieve particular cluster count and overlapping degree.

Beyond the pure use of a priori probabilities to elect the initial CHs, another significant parameter additionally used (like in HEED) is the residual energy of each node. Two such recent algorithms (similar also to LEACH with regard to the overall clustering process) were proposed in Refs. [17,18]. In Ref. [17] (Time Controlled Clustering Algorithm—TCCA), the whole operation is divided (similar to LEACH) into rounds trying to achieve better load distribution among sensor nodes. In each round initially the CH selection procedure takes place and overall cluster formation process follows. Each node decides to elect itself as a CH or not based on the suitable combination of two basic criteria, its residual energy and a preset probability "p." Actually in this step TCCA applies a direct combination of LEACH and HEED algorithms by having the (HEED inspired) energy fraction $E_{\text{residual}}/E_{\max}$ participating directly in the computation of the (LEACH inspired) CH-election threshold T_i in each round. When a CH is selected, it announces its selection to the neighboring nodes by sending a message which includes its node id, initial time-to-live, its residual energy, and a time stamp. The time-to-live parameter is selected according to the residual energy and it is used to restrict the size of the clusters that are formed. On the other hand, in Ref. [18] (Energy Efficient Clustering Scheme—EECS), a constant number of CHs are elected (i) based on their residual energy (as the main criterion) and (ii) using localized competition process without iteration to complete the cluster formation process. Specifically, the candidate CHs compete for their chance to be elected at any given round by broadcasting their residual energy to neighboring candidates. If a given node does not find a node with more residual energy, it becomes the CH. Additionally, clusters are then formed by retaining variable sizes dynamically, mainly depending on the distance of each cluster from the BS. As a result, the corresponding algorithm can effectively lead to better energy consumption and uniform load distribution (having a clearly better behavior compared to LEACH in simulated experiments), based on the fact that clusters at a greater distance from the BS require more energy for transmission than those that are closer.

Also, considering the HEED algorithm a slight (however effective) modification was also proposed in Ref. [20]. Specifically, the difference here is the treatment of nodes that eventually did not hear from any CH (orphaned nodes); during the finalization phase of the initial protocol all these nodes become CHs themselves. On the contrary, in Ref. [20] the authors claim that re-executing the algorithm for just those orphaned nodes could lead to significant improvements. Furthermore, this slight modification was shown to significantly decrease the CHs' count which then leads to reduced size (shorter paths) of the routing tree needed during inter-CH communication which finally results in faster data gathering procedures.

Similarly, considering the multi-level EEHC algorithm, a valuable extension (that includes additional CH election criteria) is proposed in Ref. [19] (EEMC), where the expected number of CHs at each level is previously determined by analytical formulas. The authors generalize the analysis given in Ref. [13] and present results about the optimal number of CHs at a certain level. Considering the formation process, they follow a top-down approach starting from the formation of level-1 clusters. The CHs at each level are randomly selected according to a certain probability. The probability of a node becoming a CH is proportional to the residual energy of the node as well as the distance of this node to the sink node (or to the CH it belongs to at lower levels). The distance is taken into account as each CH should transmit the aggregated data on behalf of its member nodes to its next level CH and a large distance between these two nodes contributes to fast energy consumption in the transmitting CH. The probabilities are also normalized so that the expected number of CHs at each level is according to the optimal values determined in their analysis. Extensive simulation work is also provided, in which the EEMC protocol is shown to achieve longer network lifetime and less latency compared to LEACH and EEHC protocols.

Finally, some random selection protocols have also been developed that follow an even more clear random CH election procedure (i.e., by randomly waiting or by generating a random competition, etc.). Such an early proposed algorithm was RCC [21], which was initially designed for MANETs and applies the 'First Declaration Wins' rule. In Ref. [22], another completely randomized clustering algorithm (CLUBS) was proposed, where each node participates in the election procedure by choosing a random number from a fixed integer range and then it counts down from that number silently. Two more recent and quite efficient (converging in constant time) completely randomized protocols were proposed in Refs. [23] and [24]. In Ref. [23] (Fast Local Clustering service—FLOC), a distributed protocol that produces approximately equal sized clusters with minimum overlap is presented.

On the other hand, in Ref. [24] (Ring-structured Energy-efficient Clustering Algorithm—RECA) sensors are grouped into one-hop bridgeable clusters during the initial cluster formation phase. Firstly, the expected number of nodes in one cluster is estimated a priori as $\gamma = N \times \pi \times R^2/A$, where N is the total number of nodes in the network, A is the area that the network covers, and R is the minimum transmission range. A node may elect itself to be a CH or a cluster member. In each slot, each node, which has neither elected itself a CH nor associated itself with a cluster, generates a random number in the [0,1] interval and compares the generated number to a threshold $h = \min(2^r \times 1/\gamma, 1)$, where r is the current slot number. If the generated number is less than h, the node becomes a CH and announces this information to nodes which reside within its area, otherwise the node waits and listens to other CH announcements. Upon receiving CH announcements, the node associates itself with the best signal-to-noise ratio (SNR) cluster. After approximately $\log_2 \gamma$ time slots, each node in the network is either a CH or a cluster member. RECA uses also a deterministic algorithm to rotate the role of CH within a cluster. In each round, cluster nodes take turns to be CHs, based on their position in the logical ring. The simulation results provided (comparing to LEACH and HEED protocols) show that RECA achieves even distribution of energy consumption among the nodes of each cluster and outperforms LEACH and HEED in terms of expected network lifetime.

In conclusion, the probabilistic (clearly random or hybrid) protocols can be regarded as the leading class of clustering algorithms for WSNs due to their simplicity and their high energy efficiency. Simple protocols like LEACH, EEHC, and HEED introduced such alternative techniques with low complexity time and improved energy efficiency, whereas later many other extensions and similar probabilistic approaches were developed (mostly by extending or combining the advantages of the above basic protocols) that presented very satisfactory total performance. The EEMC, EECS, MOCA, TCCA, and RECA protocols can be regarded as the most valuable of such recent probabilistic WSN clustering approaches, in terms of limited and balanced energy consumption and increased network lifetime. The basic disadvantage (however not critical in large scale environments) of these protocols is that due to their probabilistic nature, the CHs are not always distributed well and the CH role is not always rotated uniformly, which sometimes influences the distribution of energy consumption. The hybrid probabilistic protocols (like HEED and its extensions) behave better with regard to this aspect, however they usually lead (due to the extra processing needed) to increased total time compared to clearly random protocols.

12.4 Nonprobabilistic Clustering Approaches

Alternatively to the probabilistic (randomized or not) algorithms presented in the previous section, another basic class of clustering algorithms for WSNs primarily adopt more specific (deterministic) criteria for CHs election and cluster formation, which are mainly based on the nodes' proximity (connectivity, degree, etc.) and on the information received from other closely located nodes. The cluster formation procedure here is mainly based on the communication of nodes with their neighbors (one or multiple hops neighbors) and generally requires more intensive exchange of messages and probably graph traversing in some extent. The use of additional metrics (including the remaining energy, transmission power, mobility, etc.) in the form of combined weighted values is also a quite promising technique followed to achieve more generalized goals than other single-metric protocols. Furthermore, an even more challenging and promising nonprobabilistic clustering approach is based on the use of swarm intelligence and has led to the construction of corresponding biologically inspired clustering protocols that already have been shown to extend network lifetime in WSNs. Finally, in a few other approaches (not described here in details), beyond the application of typical proximity–connectivity criteria, the proposed protocols are primarily guided by specific (however virtual) structure-based sensors organizations and then by progressive cluster formation steps which normally lead to clusters with more controlled/predictable characteristics. Such examples can be found in Refs. [25,26].

12.4.1 Node Proximity and Graph-Based Clustering Protocols

Such a proximity-traversing-based algorithm was earlier proposed in Ref. [27] (Hierarchical Control Clustering—HCC). It is a distributed multi-hop hierarchical clustering algorithm which also efficiently extends to form a multi-level cluster hierarchy. Any node in the WSN can initiate the cluster formation process. The algorithm proceeds in two phases, "Tree Discovery" and "Cluster Formation." The tree discovery phase is basically a distributed formation of a Breadth-First-Search (BFS) tree rooted at the initiator node. Each node, u, broadcasts a signal once every p units of time, carrying the information about its shortest hop distance to the root, r. A node v that is neighbor of u will choose u to be its parent and will update its hop distance to the root, if the route through u is shorter. The broadcast signal carries the parent ID, the root ID, and the sub tree size. Every node updates its sub tree size when its children sub tree size change. The cluster formation phase starts when a sub tree on a node crosses the size parameter, k. The node initiates cluster formation on its sub tree. It will form a single cluster for the entire sub tree if the sub tree size is less than $2k$, or else, it will form multiple clusters. The cluster size and the degree of overlap are also considered. In Figure 12.4 the proposed multi-level

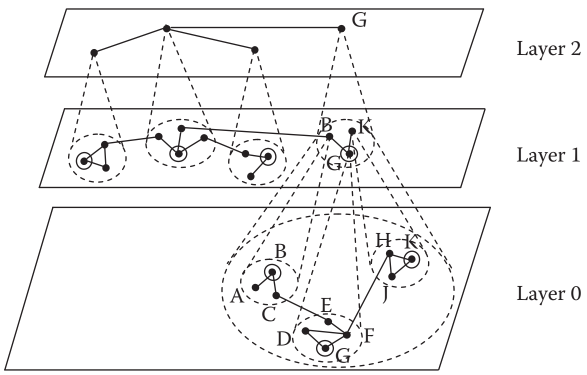

Figure 12.4 HCC Three-layer cluster hierarchy. (Redrawn from Banerjee, S. and Khuller, S., A clustering scheme for hierarchial control in multi-hop wireless networks, in *Proceedings of the 20th Joint Conference of the IEEE Computer and Communication Societies (INFOCOM 01)*, Anchorage, AK, April 2001.)

hierarchy is further illustrated. This approach has a time complexity of $O(n)$, however it has been shown to achieve quite balanced clustering as well as to handle dynamic environments very well.

Two other early proposed algorithms of this category (designed primarily for MANETs, however applicable in WSNs too) can be found in Ref. [28] (Highest Connectivity–HC) and [29] (Max-Min D-Cluster algorithm). In Ref. [28] a connectivity-based heuristic is proposed, in which the sensor node with maximum number of one-hop neighbors is elected as a CH in its neighborhood. The formation of one-hop clusters and the clock synchronization requirement limit the practical usage of this algorithm nowadays. On the other hand, in Ref. [29], a distributed algorithm is proposed, in which the clusters consist of nodes that are no more than d-hops away from the CH. The algorithm has complexity $O(d)$, it does not require clock synchronization and it provides a better load balancing compared to LCA and HC algorithms.

Other more recent examples of proximity-connectivity and neighbors' information-based algorithms have been proposed in Refs. [30–32]. In Ref. [30] a typical centralized, graph-based clustering approach (EEDC) is presented. To minimize the number of clusters and therefore maximize the energy saving, EEDC models the cluster creation process as a clique-covering problem and uses the minimum number of cliques to cover all vertices in the graph. The sink also dynamically adjusts the clusters based on spatial correlation and the received data from the sensors. The algorithm produces robust and well-balanced clusters, however it is centralized and thus not suitable for large-scale WSNs.

In Ref. [31] (Clustering Algorithm via Waiting Timer—CAWT), a distributed proximity-connectivity-based algorithm for constructing cluster hierarchy has been proposed for homogeneous sensors with the same transmission range. Once sensors are deployed, each sensor broadcasts a "hello" message to show its presence to the neighbors while listening to the others. The sensors that hear a significant number of "hello" messages (meaning that are nodes with high connectivity) organize into clusters while others are waiting to form clusters. The performance of the algorithm was evaluated using simplified simulations leading to quite good results with regard to network lifetime. However, as it is clearly observed, the generalization of the algorithm is subject to detailed evaluation with respect to load balancing, CH reelection, and energy usage across the network.

Similarly, in Ref. [32] (EACLE) a distributed clustering procedure, which beyond the proximity takes also in account the residual energy of each node, is followed. It is mainly based on the information of 2-hop neighbors with a practical transmission power control scheme, and then builds a broadcast tree only by cluster heads. Initially, each sensor is in a 'waiting' state and waits for time T_1 which is a monotonous decreasing function on the residual energy of the node. When the timer expires, the waiting node becomes a CH and broadcasts two packets with different (power-high and power-low) transmission power each, which contain the list of the neighbor-IDs received before broadcasting. When a waiting node receives a power-low packet it becomes a member node, whereas when it receives a power-high packet, it compares its own neighbor list with the list of IDs in the receiving packet, to decide if it should continue waiting or become a CH. Also, each node executes the clustering process periodically. Once a node becomes a CH in a specific round, its timer is then set to a longer value to avoid becoming a CH again in the next round.

Also, a quite valuable alternative approach was given in Ref. [33] (Algorithm for Cluster Establishment, ACE). Unlike other distributed clustering schemes, ACE employs an emergent algorithm. Emergent algorithms much like artificial neural networks evolve to optimal solution through a mix of local optimization steps. Initially, a node decides to become a "candidate" CH, and then it broadcasts an invitation message. Upon getting the invitation, a neighboring sensor joins the new cluster and becomes a follower of the new CH. At any moment, a node can be a follower of more than one cluster. Next, the migration phase takes place in order the best candidate for being CH to be selected. Each CH periodically checks the ability of its neighbors for being a CH and decides to step down if one of these neighbors has more followers than it does. A node that has the largest number of followers and the least overlap with other clusters will be considered as the best final candidate for CH. The algorithm converges in time $O(d)$ where d is the node density per unit disk. Experimental validation of ACE indicated that it achieves low variance and high average of cluster sizes when compared to node-ID-based schemes like [7] and [8].

Finally, toward the direction of efficient data gathering and aggregation, some other alternative solutions, without direct clustering, have also been proposed [34,35]. These approaches are mainly based on graph traversing heuristics and they have shown worth telling improvements compared to typical cluster-based implementations like LEACH. Similarly, in Ref. [36] the authors propose a corresponding "hybrid" clustering protocol (PEACH) which builds an adaptive clustering hierarchy without incurring the clustering formation overhead commonly met in other direct clustering algorithms. Instead, by overhearing the packets transmitted and received by neighboring nodes, each node can determine its role (CH or not) and possibly join the cluster of other node, thereby creating a clustering hierarchy. In PEACH protocol, a node becomes a CH when it hears a packet destined for the node itself. Otherwise, when the packet is destined for a different node, the node that overheard the packet joins the cluster of the destination node. By means of simulation, the PEACH protocol was compared to other competitive approaches, such as LEACH, HEED, and PEGASIS, and showed lower energy consumption and higher network lifetime mainly because the clustering hierarchy is created on the fly based on the information overheard by nodes.

In conclusion, the node-proximity and graph traversing clustering protocols achieve quite balanced and stable clusters (quite uniform distribution of CHs in the entire area, low intracluster communication cost, etc.), however they present several disadvantages. They usually lead to increased complexity time to satisfy more qualitative criteria, whereas important parameters like the number of clusters and the size of each cluster cannot easily be controlled without qualitative cost. Additionally, most of these protocols do not apply effective CHs rotation procedures (or they do not apply such procedures at all) leading to reduced energy efficiency and worse network lifetime than probabilistic protocols. The HCC, ACE, CAWT, and EACLE protocols can be regarded as the most valuable approaches in this category, however they do not present (in terms of energy consumption and network lifetime) good results as the corresponding probabilistic clustering protocols. On the contrary, special attention has to be paid to some recent hybrid (without direct clustering) clustering protocols, like the PEACH protocol, that seem to achieve lower energy consumption by effectively avoiding part of the cluster formation overhead.

12.4.2 Weight-Based Clustering Protocols

In addition to node proximity, some other known algorithms use a combination of metrics such as the remaining energy, transmission power, etc., (thus forming corresponding combined weights) to achieve more generalized goals than single-criterion protocols. Several algorithms following this directive were initially borrowed from the field of mobile ad hoc networks, i.e., [37] and [38]. As a typical example, in Ref. [38] (WCA), a corresponding weight-based protocol was proposed where the CH election process is based on the computation of a "combined weight" W_v for each node, which takes into account several system parameters such as the node degree, the transmission power, mobility, and the remaining energy of the node: $W_v = w1T_v + w2D_v + w3M_v + w4P_v$. The combined weight is calculated and broadcasted by each node. The node with the smallest weight in its neighborhood is chosen as a CH. This is a nonperiodic procedure for CH election; it is invoked on demand every time a reconfiguration of the network's topology is unavoidable. This algorithm attempted to provide better load balancing through reduced number of sensors in a cluster but the requirement of clock synchronization limits its applications.

Two more recent weight-based protocols were proposed in Refs. [39,40]. In Ref. [39] [Distributed Weight-Based Energy-Efficient Hierarchical Clustering (DWEHC)] a corresponding distributed algorithm is given, which aims at high energy efficiency by generating balanced cluster sizes and optimizing the intracluster topology. Each sensor calculates its weight after locating the neighboring nodes in its area. The weight is a function of the sensors residual energy and the proximity to the neighbors. In a neighborhood, the node with largest weight would be elected as a CH and the remaining nodes become members. At this stage the nodes are considered as first-level members because they have a direct link to the CH. A node progressively adjusts such membership to reach a CH using the least amount of energy. Basically, a node checks with its non-CH neighbors to find out their minimal cost for reaching a CH. Given the node's knowledge of the distance to its neighbors, it can assess whether it is better to stay a first-level member or become a second-level one reaching the CH over a two-hop path. Figure 12.5, illustrates the structure of the intracluster topology. Compared to HEED, the DWEHC algorithm has been shown to generate more well-balanced clusters as well as to achieve significantly lower energy consumption in intracluster and intercluster communication.

Similarly, in Ref. [40] (Topology Adaptive Spatial Clustering—TASC), the authors propose another distributed algorithm that partitions the network into a set of locally isotropic, nonoverlapping clusters without prior knowledge of the number of clusters, cluster size, and node coordinates. This is achieved by deriving a set of weights that include distance, connectivity, and density information within the locality of each node. The derived weights form the terrain for holding a coordinated leader election procedure in which each node selects the node closer to the center of mass of its neighborhood to become its leader.

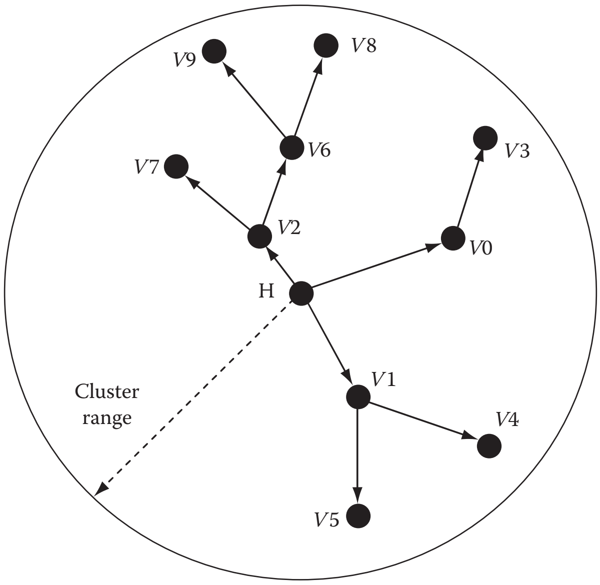

Figure 12.5 DWEHC multi-hop intracluster topology. (Redrawn from Ding, P. et al., Distributed energy efficient hierarchical clustering for wireless sensor networks, in *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems(DCOSS05)*, Marina Del Rey, CA, June 2005.)

Generally, the weight-based clustering protocols have been shown to produce well-balanced and stable clusters like the node-proximity and graph-based protocols, in a more systematic and deterministic way. Additionally they can achieve better distribution of energy consumption because most of them consider the residual energy as part of the computed weights during the CH election process. However, they normally suffer from the same disadvantages (increased communication time, no CHs rotation, etc.) as the node-proximity and graph-based protocols that were examined in the previous paragraph.

12.4.3 Biologically Inspired Clustering Approaches

Finally, the last few years some new algorithms have also been proposed based on swarm intelligence techniques which model the collective behavior of social insects such as ants. They have shown very promising results in simulated experiments (compared to protocols like LEACH and HEED) with regard to network lifetime. In Ref. [41] the authors propose such a swarm intelligence-based clustering algorithm based on the ANTCLUST method. ANTCLUST is a model of an ant colonial closure to solve clustering problems. In colonial closure model, when two objects meet together they recognize whether they belong to the same group by exchanging and comparing information about them. In the case of a WSN, initially the sensor nodes with more residual energy become CHs independently. Then, randomly chosen nodes meet each other, exchange information, and clusters are created, merged, and discarded through these local meetings and comparison of their information. Each node with less residual energy chooses a cluster based on specific criteria, like the residual energy of the CH, its distance to the CH, and an estimation of the cluster size. Eventually, energy efficient clusters are formed that result in an extension of the lifetime of the WSN.

Another related approach that ensures the good distribution of CHs and high energy efficiency, can be found in Ref. [42]. Also, in Ref. [43], a protocol that has the objective of minimizing the intracluster distance and optimizing the energy consumption of the network using Particle Swarm Optimization (PSO) is presented and evaluated via simulations. Generally, biologically inspired clustering algorithms show that they can dynamically control the CH selection while achieving quite uniform distribution of CHs and energy consumption. However, they have to be studied further as it is pointed out in the literature.

12.5 Clustering Algorithms for Reactive Networks

All the algorithms presented in the previous sections (having been considered as "proactive" clustering algorithms) are based on the assumption that the sensors always have data to send, hence, they should all be considered during the cluster formation. In contrast, "reactive" algorithms take advantage of user queries for the sensed data or of specific triggering events that occur in the WSN. Namely, nodes may react instantly to sudden and drastic changes in the value of a sensed attribute. This approach is useful for time-critical applications, but not particularly suited for applications where data retrieval is required on a regular basis.

The Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [44] forms a hierarchical clustered structure, grouping nearby nodes within the same cluster. The protocol focuses on information aggregation rather than on cluster formation, which is very similar to LEACH. The protocol defines two thresholds: the hard threshold is a threshold (absolute) value for the sensed attribute, while the soft threshold is a threshold (small change) value of the sensed attribute. The concept of threshold is highly significant in a variety of WSN applications, such as fire alarm, temperature monitoring, etc. The nodes transmit sensor readings only when they fall above the hard threshold and change by given amount (soft threshold). In TEEN, sensor nodes sense the medium continuously, but data transmission is done less frequently which favors the energy saving. However, if the thresholds are not crossed, the nodes will never communicate, namely TEEN does not support periodic reports. AQ1

The Adaptive Periodic-TEEN (APTEEN) [45] is a variation of TEEN which addresses the latter's main shortcoming. It is a hybrid routing protocol wherein the nodes still react to time-critical situations, but also give an overall picture of the network at periodic intervals in an energy efficient manner. The CH selection in APTEEN is based on the mechanism used in LEACH-C. The clusters are valid for an interval called the cluster period. At the end of this period, the BS performs reclustering. Assuming that adjacent nodes register similar data, APTEEN forms pairs of nodes where only one of them responds to queries. These two nodes can alternate in the role of handling queries, thereby saving resources. Simulations have indicated that APTEEN's performance lies in between TEEN and LEACH in terms of average energy dissipation events and network lifetime. That is a reasonable conclusion that derives from APTEEN's hybrid proactive-reactive nature. The main drawbacks of TEEN and APTEEN lie on the control overhead associated with the formation of multiple-level clusters, the method of implementing threshold functions, and none of them exploits spatial and temporal data correlation to improve efficiency.

Decentralized Reactive Clustering (DRC) has been proposed in Ref. [48]. Similarly to other reactive algorithms, the clustering procedure is initiated only in the case of events detection. Four different operation phases are defined: the postdeployment phase, followed by a cluster-forming phase which is when clusters are constructed, then an intra-cluster data processing phase and finally a CH-to-processing center phase. DRC uses power control technique to minimize energy usage in cluster formation. Unfortunately, simulations only compare DRC against LEACH and therefore do not highlight its performance gains or disadvantages against other reactive clustering protocols.

More recently, the Clustered AGgregation (CAG) [46], mechanism was proposed, which utilizes the spatial correlation of sensory data to further reduce the number of transmissions by providing approximate results to aggregate queries. CAG guarantees the result to be within a user-specified error-tolerance threshold. Cluster formation is performed while queries are disseminated to the network (query phase), where clusters group nodes sensing similar values. Subsequently, CAG enters the response phase wherein only one aggregated value per cluster is transmitted up the aggregation tree. In effect, CAG is a lossy clustering algorithm (most sensory readings are never reported) which trades a lower result precision for a significant energy, storage, computation, and communication saving.

The Updated CAG algorithm [47] extends CAG defining two operation modes, depending on the dynamics of the environment. In the interactive mode, users issue a one-shot query and the network generates a single response. This is appropriate for scenarios where the environment changes dynamically, or when users desire to change the approximation granularity or query attributes interactively. On the other hand, in the streaming mode, the CHs transmit a stream of response for a query that is issued just once. This mode of operation is well suited for static environments where sensor readings do not change frequently and the query remains valid for a certain period of time. Note that the interactive mode only exploits the spatial correlation of the sensor data to form clusters, whereas the streaming mode leverages both temporal and spatial correlations. The latter adjusts clusters locally as the data and topology change over time. Overall, CAG and Updated CAG approaches provide efficient data aggregation and energy saving solutions at the expense of a precision error bounded by a user-provided threshold.

In an even more recent approach, Guo and Li proposed Dynamic-Clustering Reactive Routing (DCRR) algorithm [49]. DCRR borrows ideas from biological neuron networks, following the observation that the latter also employ a many-to-one (neurons-to-brain) communication paradigm, similarly to the nodes of a WSN. In DCRR, once an incident emerges, the CH is dynamic selected in the incident region according to the nodes' residual energy. DCRR defines a TEEN-inspired "action threshold" for firing data to the sink. That threshold is dynamically adjusted to trace the changing speed of the incident. The action threshold is also intentionally fluctuated outside the incident region to enable all network nodes to send data in periodic basis (so that they are not misconceived as failed ones).

In conclusion, reactive clustering algorithms relax the network from the control overhead and the time required to perform the clustering process in the WSN. The use of attributes thresholds and the exploitation of the spatial and temporal correlation of sensory data, inherent in many networking environments, compress the magnitude of event-driven data transmissions. These algorithms are also highly responsive to radical changes in the monitored environment and are particularly useful for time-sensitive applications. However, they do not suit applications which require periodic retrieval of sensory readings, wherein the construction of stable and energy-efficient cluster structures is of critical importance.

12.6 Conclusion

Generally, clustering in WSNs has been of high interest in the last decade and there is already a large number of related published works. Throughout this chapter we tried to present the main characteristics of the most significant protocols that were proposed till now in the literature. As it was pointed out, grouping nodes into clusters, thus leading to hierarchical routing and data gathering protocols, has been regarded as the most efficient approach to support scalability in WSNs. The hierarchical cluster structures facilitate the efficient data gathering and aggregation independent to the growth of the WSN, and generally reduce the total amount of communications as well as the energy spent.

The main objective of most of the existing protocols lies on how to prolong the lifetime of the network and how to make a more efficient use of the critical resources, such as buttery power. Furthermore, the combined need for fast convergence time and minimum energy consumption (with regard to the cluster formation procedure) led to appropriate fast distributed probabilistic (clearly random or hybrid) clustering algorithms which quickly became the most popular and widely used in the field. In these algorithms the nodes are assumed to make fast decisions (i.e., to become CHs or not) based on some probability or other local information only (i.e., on their residual energy) and usually the desired quality of the final cluster output is considered as a secondary parameter only. Another critical feature of most of these algorithms (leading to more uniform distribution of the energy consumption) is the periodic reelection of CHs (rotation of the CH role) among all the nodes of the network. Clustering algorithms that adopt as primary election criteria other classical parameters like connectivity, nodes' proximity, distance, etc., have also been developed and relevant protocols are still being used, leading probably to more qualitative output (well balanced clusters, etc). However the time complexity of these algorithms is difficult to be kept low as in leading probabilistic/random clustering algorithms.

Moreover, because the size of the WSNs (number of sensors, area covered, etc.) used in real applications become larger and larger, the extension in multi-hop communication patterns (with regard to intracluster and/or intercluster communication) is unavoidable, whereas the multi-level cluster hierarchies have also been regarded as a promising option preserving energy efficiency independent of the growth of the network. Significant progress has also been noted in specialized protocols for timely critical applications, where nodes should react instantly to sudden and drastic changes in the value of a sensed attribute.

Finally, several additional issues should be further studied in future research. Some of the most challenging of these issues include the development of a generic method for finding the optimal number of clusters in order maximize the energy efficiency, the estimation of the optimal frequency of CH rotation/reelection to gain better energy distribution, however, keeping the total overhead low, the efficient support of nodes and CHs mobility as well as the support of mobile sinks, the incorporation of several security aspects (i.e., enhanced protection needed in hostile environments when cluster-based protocols are used), the further development of efficient recovery protocols in case of CHs failure, etc.

AQ2

References

1. A.A. Abbasi and M. Younis, A survey on clustering algorithms for wireless sensor networks, *Computer Communications*, 30, 2826–2841, 2007.
2. K. Sohrabi et al., Protocols for self-organization of a wireless sensor network, *IEEE Personal Communications*, 7(5), 16–27, 2000.
3. R. Min et al., Low power wireless sensor networks, in *Proceedings of International Conference on VLSI Design*, Bangalore, India, January 2001.
4. M. Liliana, C. Arboleda, N. Nidal, Comparison of clustering algorithms and protocols for wireless sensor networks, in *Proceedings of IEEE CCECE/CCGEI Conference*, Ottawa, Ontario, Canada, pp. 1787–1792, May 2006.
5. G. Gupta and M. Younis, Load-balanced clustering in wireless sensor networks, in *Proceedings of the International Conference on Communication (ICC 2003)*, Anchorage, AK, May 2003.
6. E.I. Oyman and C. Ersoy, Multiple sink network design problem in large scale wireless sensor networks, in *Proceedings of the IEEE International Conference on Communications (ICC 2004)*, Paris, June 2004.
7. D.J. Baker and A. Ephremides, The architectural organization of a mobile radio network via a distributed algorithm, *IEEE Transactions on Communications*, 29(11), 1694–1701, 1981.
8. A. Ephremides, J.E. Wieselthier, and D.J. Baker, A design concept for reliable mobile radio networks with frequency hopping signalling, *Proceedings of IEEE*, 75(1), 56–73, 1987.
9. C.R. Lin and M. Gerla, Adaptive clustering for mobile wireless networks, *IEEE Journal on Selected Areas Communications*, 15(7), 1265–1275, 1997.
10. W.P. Chen, J.C. Hou, and L. Sha, Dynamic clustering for acoustic target tracking in wireless sensor networks, in *11th IEEE International Conference on Network Protocols (ICNP'03)*, pp. 284–294, 2003.

11. W.R. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, Energy efficient communication protocol for wireless microsensor networks, in *Proceedings of the 33rd Hawaaian Interantional Conference on System Sciences*, January 2000.
12. W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, An application specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications*, 1(4), 660–670, 2002.
13. S. Bandyopadhyay and E. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in *22nd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2003)*, San Francisco, CA, April 2003.
14. O. Younis and S. Fahmy, HEED: A hybrid, energy-efficient, distributed clustering approach for Ad Hoc sensor networks, *IEEE Transactions on Mobile Computing*, 3(4), 366–379, 2004.
15. V. Loscri, G. Morabito, and S. Marano, A two-level hierarchy for low-energy adaptive clustering hierarchy, *Proceedings of IEEE VTC Conference 2005*, Vol. 3, pp. 1809–1813, 2005.
16. A. Youssef, M. Younis, M. Youssef, and A. Agrawala, Distributed formation of overlapping multi-hop clusters in wireless sensor networks, in *Proceedings of the 49th Annual IEEE Global Communication Conference (Globecom06)*, San Francisco, CA, Nov. 2006.
17. S. Selvakennedy and S. Sinnappan, An adaptive data dissemination strategy for wireless sensor networks, *International Journal of Distributed Sensor Networks*, 3(1), 2007. AQ3
18. M. Ye, C. Li, G. Chen, and J. Wu, EECS: An energy efficient clustering scheme in wireless sensor networks, in *Proceedings of IEEE Intl. Performance Computing and Communications Conference (IPCCC)*, pp. 535–540, 2005.
19. Y. Jin, L. Wang, Y. Kim, and X. Yang, EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks, Elsevier, *Computer Networks Journal*, 52, 542–562, 2008.
20. H. Huang and J. Wu, A probabilistic clustering algorithm in wireless sensor networks, in *Proceedings of IEEE 62nd VTC Conference*, Dallas, TX, September 2005.
21. K. Xu and M. Gerla, A heterogeneous routing protocol based on a new stable clustering scheme, in *Proceeding of IEEE Military Communications Conference* (MILCOM 2002), Anaheim, CA, October 2002.
22. R. Nagpal and D. Coore, An algorithm for group formation in an amorphous computer, in *Proceedings of the 10th International Conference on Parallel and Distributed Systems* (PDCS98), Las Vegas, NV, October 1998.
23. M. Demirbas, A. Arora and V. Mittal, FLOC: A fast local clustering service for wireless sensor networks, in *Proc. of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS04)*, Florence, Italy, June 2004.
24. G. Li and T. Znati, RECA: A ring-structured energy-efficient clustering architecture for robust communication in wireless sensor networks, *International Journal Sensor Networks*, 2(1/2), 2007.
25. H. Zhang and A. Arora, GS3: Scalable self-configuration and self-healing in wireless networks, in *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, Monterey, CA, July 2002.

26. L. Yu, N. Wang, W. Zhang, and C. Zheng, GROUP: A grid-clustering routing protocol for wireless sensor networks, in *Proceedings of International Conference on Wireless Communications*, Networking and Mobile Computing, 2006.
27. S. Banerjee and S. Khuller, A clustering scheme for hierarchical control in multi-hop wireless networks, in *Proceedings of 20th Joint Conference of the IEEE Computer and Communications Societies (INFOCOMŠ 01)*, Anchorage, AK, April 2001.
28. A.K. Parekh, Selecting routers in ad-hoc wireless networks, in *Proceedings of SBT/IEEE International Telecommunications Symposium*, ITS, 1994.
29. A. Amis, R. Prakash, T. Vuong, and D. Huynh, Max-Min D-Cluster formation in wireless Ad Hoc networks, *IEEE INFOCOM*, March 2000.
30. C. Liu, K. Wu, and J. Pei, A dynamic clustering and scheduling approach to energy saving in data collection from wireless sensor networks, in *2nd Annual IEEE Conf. on Sensor and Ad Hoc Communications and Networks (SECON'05)*, September 2005.
31. C. Wen and W. Sethares, Automatic decentralized clustering for WSNs, *EURASIP Journal on Wireless Communications and Networking*, 5(5), 686–697, 2005.
32. K. Yanagihara, J. Taketsugu, K. Fukui, S. Fukunaga, S. Hara and K.I. Kitayama, EACLE: Energy-aware clustering scheme with transmission power control for sensor networks, Springer, *Wireless Personal Communications*, 40(3), 401–415, 2007.
33. H. Chan and A. Perrig, ACE: an emergent algorithm for highly uniform cluster formation, in *Proceedings of the 1st European Workshop on Sensor Networks (EWSN)*, Berlin, Germany, January 2004.
34. S. Lindsey and C.S. Raghavendra, PEGASIS: Power-efficient gathering in sensor information networks, Computer Systems Research Department, the Aerospace Corporation, 3, pp. 3-1125–3-1130, Los Angeles, CA, 2002.
35. H.O. Tan and I. Korpeoglou, Power efficient data gathering and aggregation in wireless sensor networks, *Issue on Sensor Networks Technology, ACM SIGMOD Record*, 32(4), 66–71, 2003.
36. S. Yi, J. Heo, Y. Cho, and J. Hong, PEACH: Power-efficient and adaptive clustering Hierarchy Protocol for WSNs, Elsevier, *Computer Networks*, 30, 2842–2852, 2007.
37. S. Basagni, Distributed clustering for Ad Hoc networks, in *Proceedings of Intl. Symp. on Parallel Architectures, Algorithms & Networks (ISPAN)*, pp. 310–315, 1999.
38. M. Chatterjee, S.K. Das, and D. Turgut, WCA: A weighted clustering algorithm for mobile ad hoc networks, *Clustering Computing*, 5, 193–204, 2002.
39. P. Ding, J. Holliday, and A. Celik, Distributed energy efficient hierarchical clustering for wireless sensor networks, in *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems(DCOSS05)*, Marina Del Rey, CA, June 2005.
40. R. Virrankoski, A. Savvides, TASC: Topology adaptive clustering for sensor networks, in *Proceedings of the Second IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, (MASS 2005)*, Washington, DC, November 2005.
41. J. Kamimura, N. Wakamiya, and M. Murata, A distributed clustering method for energy-efficient data gathering in sensor networks, *Int. J. Wireless and Mobile Computing*, 1(2), 113–120, 2006.

42. S. Selvakennedy, S. Sinnappan, and Y. Shang, A biologically inspired clustering protocol for wireless sensor networks, *Computer Communications* 30, 2786–2801, 2007.
43. N.M. Abdul Latiff, C.C. Tsimenidis, and B.S. Sharif, Energy-aware clustering for wireless sensor networks using particle swarm optimization, in *IEEE Intl. Symposium PIMRC'07*, pp. 1–5, Athens, Greece, Sept. 2007.
44. A. Manjeshwar and D.P. Agrawal, TEEN: A routing protocol for enhanced efficiency in wireless sensor networks, in *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, San Francisco, CA, 2001.
45. A. Manjeshwar and D.P. Agrawal, APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in *Proceedings of the International Parallel and Distributed Processing Symposium*, 2002.
46. S. Yoon and C. Shahabi, Exploiting spatial correlation towards an energy efficient Clustered AGgregation technique (CAG), in *IEEE International Conference on Communications*, pp. 82–98, 2005.
47. S. Yoon and C. Shahabi, The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks, *ACM Transactions on Sensor Networks*, 3(1), March 2007.
48. Y. Xu and H. Qi, Decentralized reactive clustering for collaborative processing in sensor networks, in *Proceedings of the 10th International Conference on Parallel and Distributed Systems (ICPADS'2004)*, pp. 54–61, July 2004.
49. B. Guo and Z. Li, A dynamic-clustering reactive routing algorithm for wireless sensor networks, *Wireless Networks Journal*, Springer, ISSN 1572-8196 (Online Publication), September 2007. AQ4
50. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, Next century challenges: scalable coordination in sensor networks, in *Proceedings of the ACM/IEEE MOBICOM Intl. Conference*, Boston, MA, pp. 6–11, August 2000.
51. C.M. Liu and C.H. Lee, Power efficient communication protocols for data gathering on mobile sensor networks, in *IEEE 60th VTC*, Los Angeles, CA, pp. 4635–4639, 2004.
52. X. Zhang, H. Wang, and A. Khokhar, An energy-efficient data collection protocol for mobile sensor networks, in *IEEE 64th VTC*, Montreal, Quebec, Canada, pp.1–5, 2006.

AUTHOR QUERY

[AQ1] Please check if the edit made to the sentence “The protocol defines . . . sensed attribute” is correct.
[AQ2] Please provide location for Refs. [10,11,18,26,28–30,37,45,46,48].
[AQ3] Please provide page range for Refs. [17,24,47].
[AQ4] Please provide complete details for Ref. [49].