Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 30 (2007) 2826–2841

computer communications

www.elsevier.com/locate/comcom

A survey on clustering algorithms for wireless sensor networks

Ameer Ahmed Abbasi [a,*], Mohamed Younis [b]

[a] Department of Computing, Al-Hussan Institute of Management and Computer Science, Dammam 31411, Saudi Arabia
[b] Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

Abstract

The past few years have witnessed increased interest in the potential use of wireless sensor networks (WSNs) in applications such as disaster management, combat field reconnaissance, border protection and security surveillance. Sensors in these applications are expected to be remotely deployed in large numbers and to operate autonomously in unattended environments. To support scalability, nodes are often grouped into disjoint and mostly non-overlapping clusters. In this paper, we present a taxonomy and general classification of published clustering schemes. We survey different clustering algorithms for WSNs; highlighting their objectives, features, complexity, etc. We also compare of these clustering algorithms based on metrics such as convergence rate, cluster stability, cluster overlapping, location-awareness and support for node mobility.

Keywords: Wireless sensor networks; Clustering algorithms; Scalability; Network architecture; Energy aware design

1. Introduction

Recent advances in miniaturization and low-power design have led to the development of small-sized battery-operated sensors that are capable of detecting ambient conditions such as temperature and sound. Sensors are generally equipped with data processing and communication capabilities. The sensing circuitry measures parameters from the environment surrounding the sensor and transforms them into an electric signal. Processing such a signal reveals some properties about objects located and/or events happening in the vicinity of the sensor. Each sensor has an onboard radio that can be used to send the collected data to interested parties. Such technological development has encouraged practitioners to envision aggregating the limited capabilities of the individual sensors in a large scale network that can operate unattended [1–7]. Numerous civil and military applications can be leveraged by networked sensors. A network of sensors can be employed to gather meteorological variables such as temperature and pressure. These measurements can be then used in preparing forecasts or detecting harsh natural phenomena. In disaster management situations such as earthquakes, sensor networks can be used to selectively map the affected regions directing emergency response units to survivors. In military situations (Fig. 1), sensor networks can be used in surveillance missions and can be used to detect moving targets, chemical gases, or the presence of micro-agents.

One of the advantages of wireless sensors networks (WSNs) is their ability to operate unattended in harsh environments in which contemporary human-in-the-loop monitoring schemes are risky, inefficient and sometimes infeasible. Therefore, sensors are expected to be deployed randomly in the area of interest by a relatively uncontrolled means, e.g. dropped by a helicopter, and to collectively form a network in an ad-hoc manner [8,9]. Given the vast area to be covered, the short lifespan of the battery-operated sensors and the possibility of having damaged nodes during deployment, large population of sensors are expected in most WSNs applications. It is envisioned that hundreds or even thousands of sensor nodes will be involved. Designing and operating such large size network would require scalable architectural and management strategies. In addition, sensors in such environments are energy

---

* Corresponding author.
E-mail addresses: ameer_abbasi@hussan.edu.sa (A.A. Abbasi), younis@csee.umbc.edu (M. Younis).

0140-3664/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.comcom.2007.05.024

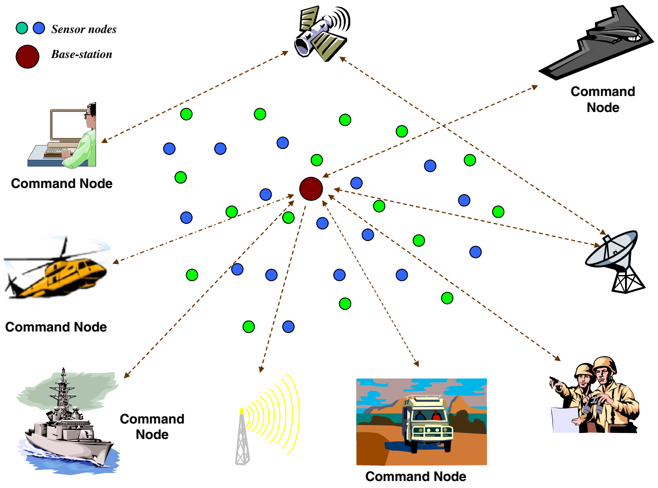

Fig. 1. An articulation of a sample WSN architecture for a military application.

constrained and their batteries cannot be recharged. Therefore, designing energy-aware algorithms becomes an important factor for extending the lifetime of sensors. Other application centric design objectives, e.g. high fidelity target detection and classification, are also considered [10].

Grouping sensor nodes into clusters has been widely pursued by the research community in order to achieve the network scalability objective. Every cluster would have a leader, often referred to as the cluster-head (CH). Although many clustering algorithms have been proposed in the literature for ad-hoc networks [11–15], the objective was mainly to generate stable clusters in environments with mobile nodes. Many of such techniques care mostly about node reachability and route stability, without much concern about critical design goals of WSNs such as network longevity and coverage. Recently, a number of clustering algorithms have been specifically designed for WSNs [16–20]. These proposed clustering techniques widely vary depending on the node deployment and bootstrapping schemes, the pursued network architecture, the characteristics of the CH nodes and the network operation model. A CH may be elected by the sensors in a cluster or pre-assigned by the network designer. A CH may also be just one of the sensors or a node that is richer in resources. The cluster membership may be fixed or variable. CHs may form a second tier network or may just ship the data to interested parties, e.g. a base-station or a command center.

In addition to supporting network scalability, clustering has numerous advantages. It can localize the route set up within the cluster and thus reduce the size of the routing table stored at the individual node [21]. Clustering can also conserve communication bandwidth since it limits the scope of inter-cluster interactions to CHs and avoids redundant exchange of messages among sensor nodes [22]. 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 [23]. 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 [22]. A CH can schedule activities in the cluster so that nodes can switch to the low-power sleep mode most of the time and reduce the rate of energy consumption. 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 [24–27]. Furthermore, a CH can aggregate the data collected by the sensors in its cluster and thus decrease the number of relayed packets [28].

In this paper, we opt to categorize clustering algorithms proposed in the literature for WSNs. We report on the state of the research and summarize a collection of published schemes stating their features and shortcomings. We also compare the different approaches and analyze their applicability. In the next section, we discuss the different classifications of clustering techniques and enumerate a set of attributes for categorizing published algorithms. In Section 3, we summarize a collection of clustering algorithms for WSNs and present classification of the various approaches pursued. Finally, Section 4 concludes the paper.

2. Taxonomy of clustering attributes

Clustering techniques for WSNs proposed in the literature can be generally classified based on the overall network architectural and operation model and the objective of the node grouping process including the desired count and properties of the generated clusters. In this section we discuss the different classifications and present taxonomy of a clustering attributes. We later use such attributes to categorize and compare the surveyed clustering algorithms.

2.1. Classifying clustering techniques

2.1.1. Network model

Different architectures and design goals/constraints have been considered for various applications of WSNs. The following enlists some the relevant architectural parameters and highlight their implications on network clustering.

- *Network dynamics:* Basically WSNs consist of three main components: sensor nodes, base-station and monitored events. Aside from the few setups that utilize mobile sensors [29,30], most of the network architectures assume that sensor nodes are stationary [19,31,32]. Sometimes it is deemed necessary to support the mobility of base-station or CHs. Node mobility would make clustering very challenging since the node membership will dynamically change, forcing clusters to evolve over time. On the other hand, the events monitored by a sensor can be either intermittent or continual depending on the application. For instance, in a target detection/tracking application, the event (phenomenon) is dynamic whereas forest monitoring for early fire prevention is an example of intermittent events. Monitoring intermittent events allows the network to work in a reactive mode, simply generating traffic when reporting. Continual events in most applications require periodic reporting and consequently generate significant traffic to be routed to the sink. Although continual events would mostly make the clusters stable, it may unevenly load CHs relative to the nodes in the cluster and a rotation of the CH role may be required if the CH is randomly picked from the sensor population. Intermittent events would favor adaptive clustering strategies if the number of events significantly fluctuates.

- *In-network data processing:* Since sensor nodes might generate significant redundant data, similar packets from multiple nodes can be aggregated so that the number of transmissions would be reduced. Data aggregation combines data from different sources by using functions such as *suppression* (eliminating duplicates), *min, max* and *average* [33]. Some of these functions can be performed either partially or fully in each sensor node, by allowing sensor nodes to conduct in-network data reduction. Recognizing that computation would be less energy consuming than communication, substantial energy savings can be obtained through data aggregation. This technique has been used to achieve energy efficiency and traffic optimization in a number of routing protocols. In some network architectures, all aggregation functions are assigned to more powerful and specialized nodes. Data aggregation is also feasible through signal processing techniques. In that case, it is referred as *data fusion* where a node is capable of producing a more accurate signal by reducing the noise and using some techniques such as *beamforming* to combine the signals [20]. It will be intuitive to expect CHs to perform such data aggregation/fusion which may restrict the choice of CH to only specialized node or require limiting the number of sensors per cluster in order to ensure that CHs are not overburdened [16]. It is worth noting that sometimes it may be necessary to assign backup CHs for a cluster or rotate the role of being CH among the sensors in the cluster [20,34]. Obviously, such design choices/constraints influence the clustering scheme.
- *Node deployment and capabilities:* Another consideration is the topological deployment of nodes. This is application dependent and affects the need and objective of the network clustering. The deployment is either deterministic or self-organizing. In deterministic situations, the sensors are manually placed and data is routed through pre-determined paths. Therefore, clustering is such setup is also preset or unnecessary. However in self-organizing systems, the sensor nodes are scattered randomly creating an infrastructure in an ad hoc manner [8,19,20]. In that infrastructure, the position of the base-station or the CH is also crucial in terms of energy efficiency and performance. When the distribution of nodes is not uniform, optimal clustering becomes a pressing issue to enable energy efficient network operation. In addition, in some setups different functionalities can be associated with the deployed nodes and the CH selection may be constrained. In networks of homogenous sensor nodes, i.e. all having equal capacity in terms of computation, communication and power, CHs are picked from the deployed sensors [20,35,36]. Often in that case, CHs are carefully tasked, e.g. excluded from sensing duties, in order to avoid depleting their energy rather quickly. In addition, the communication range and the relative CH's proximity to the base-station may also be constraints/issues that have to be considered. Sensors' communication range is usually limited and a CH may not be able to reach the base-station. Even if a node can directly communicate with the base-station, it is still better to pursue multi-hop routes. Therefore, inter-CH connectivity becomes an important factor that affects the clustering scheme [17,37]. On the other hand, heterogeneous WSNs may impose more constraints on the clustering process since some nodes may be designated for special tasks or empowered with distinct capabilities. It may then be required to either avoid such specific nodes to conserve their resources or limit the selection of CHs to a subset of these nodes.

2.1.2. Clustering objectives

Clustering algorithms in the literature varies in their objectives. Often the clustering objective is set in order to facilitate meeting the applications requirements. For example if the application is sensitive to data latency, intra and inter-cluster connectivity and the length of the data routing paths are usually considered as criteria for CH selection and node grouping. The following discussion highlights popular objectives for network clustering:

- *Load balancing:* Even distribution of sensors among the clusters is usually an objective for setups where CHs perform data processing or significant intra-cluster management duties [16]. Given the duties of CHs, it is intuitive to balance the load among them so that they can meet the expected performance goals [38]. Load balancing is a more pressing issue in WSNs where CHs are picked from the available sensors [19]. In such case, setting equal-sized clusters becomes crucial for extending the network lifetime since it prevents the exhaustion of the energy of a subset of CHs at high rate and prematurely making them dysfunctional. Even distribution of sensors can also leverage data delay [37]. When CHs perform data aggregation, it is imperative to have similar number of node in the clusters so that the combined data report becomes ready almost at the same time for further processing at the base-station or at the next tier in the network.
- *Fault-tolerance:* In many applications, WSNs will be operational in harsh environments and thus nodes are usually exposed to increased risk of malfunction and physical damage. Tolerating the failure of CHs is usually necessary in such applications in order to avoid the loss of important

sensors' data. The most intuitive way to recover from a CH failure is to re-cluster the network. However, re-clustering is not only a resource burden on the nodes, it is often very disruptive to the on-going operation. Therefore, contemporary fault-tolerance techniques would be more appropriate for that sake. Assigning backup CHs is the most notable scheme pursued in the literature for recovery from a CH failure. The selection of a backup and the role such spare CH will play during normal network operation varies. When CHs have long radio range, neighboring CHs can adapt the sensors in the failing cluster [34]. Rotating the role of CHs among nodes in the cluster can also be a means for fault-tolerance in addition to their load balancing advantage [20].

- *Increased connectivity and reduced delay:* Unless CHs have very long-haul communication capabilities, e.g. a satellite link, inter-CH connectivity is an important requirement in many applications. This is particularly true when CHs are picked from the sensors population. The connectivity goal can be just limited to ensuring the availability of a path from every CH to the base-station [17] or be more restrictive by imposing a bound on the length of the path [40]. When some of the sensors assume the CH role, the connectivity objective makes network clustering one of the many variant of the connected dominating set problem. On the other hand, when data latency is a concern, intra-cluster connectivity becomes a design objective or constraint. Delay is usually factored in by setting a maximum number of hops "K" allowed on a data path. K-hop clustering is K-dominating set problem [41–43].
- *Minimal cluster count:* This objective is particularly common when CHs are specialized resource-rich nodes [39]. The network designer often likes to employ the least number of these nodes since they tend to be more expensive and vulnerable than sensors. For example, if CHs are laptop computers, robots or a mobile vehicle there will be inherently some limitation on the number of nodes. The limitation can be due to the complexity of deploying these types of nodes, e.g. when the WSN is to operate in a combat zone or a forest. In addition, the size of these nodes tends to be significantly larger than sensors, which makes them easily detectable. Node visibility is highly undesirable in many WSNs applications such as border protection, military reconnaissance and infrastructure security.
- *Maximal network longevity:* Since sensor nodes are energy-constrained, the network's lifetime is a major concern; especially for applications of WSNs in harsh environments. When CHs are richer in resources than sensors, it is imperative to minimize the energy for intra-cluster communication [22]. If possible, CHs should be placed close to most of the sensors in its clusters [39,44]. On the other hand, when CHs are regular sensors, their lifetime can be extended by limiting their load as we mentioned earlier. Combined clustering and route setup has also been considered for maximizing network's lifetime [45]. Adaptive clustering is also a viable choice for achieving network longevity [46,47].

2.1.3. Taxonomy of clustering attributes

In this section we opt to enumerate the set of attributes that can be use to categorizes and differentiate clustering algorithms of WSNs. Based on the discussion above, we can identify the following attributes:

1. *Cluster properties:* Often clustering schemes strive to achieve some characteristics for the generated clusters. Such characteristics can be related to the internal structure of the cluster or how it relates to others. The following are the relevant attributes:
   - *Cluster count:* In some published approaches the set of CHs are predetermined and thus the number of clusters are preset. Randomly picking CHs from the deployed sensors usually yields variable number of clusters.
   - *Stability:* When the clusters count varies and the node's membership evolves overtime, the clustering scheme is said to be adaptive. Otherwise, it is considered fixed since sensors do not switch among clusters and the number of clusters stays the same throughout the network lifespan.
   - *Intra-cluster topology:* Some clustering schemes are based on direct communication between a sensor and its designated CH. However, multi-hop sensor-to-CH connectivity is sometimes required; especially when the sensor's communication range is limited and/or the CH count is bounded.
   - *Inter-CH connectivity:* When the CH does not have long haul communication capabilities, CHs connectivity to the base-station has to be provisioned. In that case, the clustering scheme has to ensure the feasibility of establishing an inter-CH route from every CH to the base-station. Some of the published work assumes that CH would be able to directly reach the base-station.

2. *Cluster-head capabilities:* As discussed earlier the network model influences the clustering approach; particularly the node capabilities and the scope of the in-network processing. The following attributes of the CH node are differentiating factors among clustering schemes:
   - *Mobility:* When a CH is mobile, sensor's membership dynamically changes and the clusters would need to be continuously maintained. On the other hand, stationary CH tends to yield stable clusters and facilitate intra- and inter-cluster network management. Sometimes, CHs can travel for limited distances to reposition itself for better network performance.
   - *Node types:* As indicated earlier, in some setups a subset of the deployed sensors are designated as CHs while in others CHs are equipped with significantly more computation and communication resources.
   - *Role:* A CH can simply act as a relay for the traffic generated by the sensors in its cluster or perform aggregation/fusion of collected sensors' data. Sometime, a CH acts as a sink or a base-station that takes actions based on the detected phenomena or targets.

3. *Clustering process:* The coordination of the entire clustering process and the characteristics of the algorithms vary significantly among published clustering schemes. The following attributes are deemed relevant:
   - *Methodology:* When CHs are just regular sensors nodes, clustering has to be performed in a distributed manner without coordination. In few approaches, a centralized authority partitions the nodes offline and controls the cluster membership. Hybrid schemes can also be found; especially when CHs are rich in resources. In the later case, inter-CHs coordination is performed in a distributed manner, while each individual CH takes charge of forming its own cluster.
   - *Objective of node grouping:* As discussed in the previous section, several objectives have been pursued for forming clusters. Examples include fault-tolerance, load balancing, network connectivity, etc.
   - *Cluster-head selection:* CHs can be pre-assigned or picked randomly from the deployed set of nodes.
   - *Algorithm complexity:* Depending on the objective and the methodology, numerous clustering algorithms have been proposed. The complexity and convergence rate of these algorithms can be constant or dependent on the number of CHs and/or sensors.

We would like to note that some of these attributes are mutually exclusive, e.g. preset or variable cluster count, and some are not. For example, a clustering process may have multiple objectives. It is also worth noting that network clustering can influence or be influenced by the planned network and link layer protocols. We plan to hint on the implications of routing and MAC protocols when we summarize the published clustering schemes. Fig. 2 summarizes the presented taxonomy of attributes. We use this set of attributes in categorizing the clustering algorithms summarized in the next section.

3. Clustering algorithms for WSNs

Generally, WSNs involve a large number of sensors ranging in the hundreds or even thousands. Clustering is an effective mean for managing such high population of nodes. In this section we present a literature survey of published distributed algorithms for clustering WSNs. Given that scalability is regarded as the main advantage of network clustering, the surveyed algorithms are grouped according to their convergence rate into two subsections for variable and constant convergence time algorithms, respectively.

3.1. Variable convergence time algorithms

Time is a significant factor in the convergence of clustering algorithms. Some of the proposed clustering algorithms such as LCA [48], RCC [52] and CLUBS [53], have $O(n)$ convergence time, where n represent the number of sensor nodes in the network. It is thus practical to implement these types of clustering algorithms to the networks having small number of nodes. However, convergence time has enhanced dramatically in some recent algorithms, e.g. [17], and showed their suitability for networks having large number of nodes. In general, variable convergence time algorithms enable more control of the cluster properties than the constant time ones.

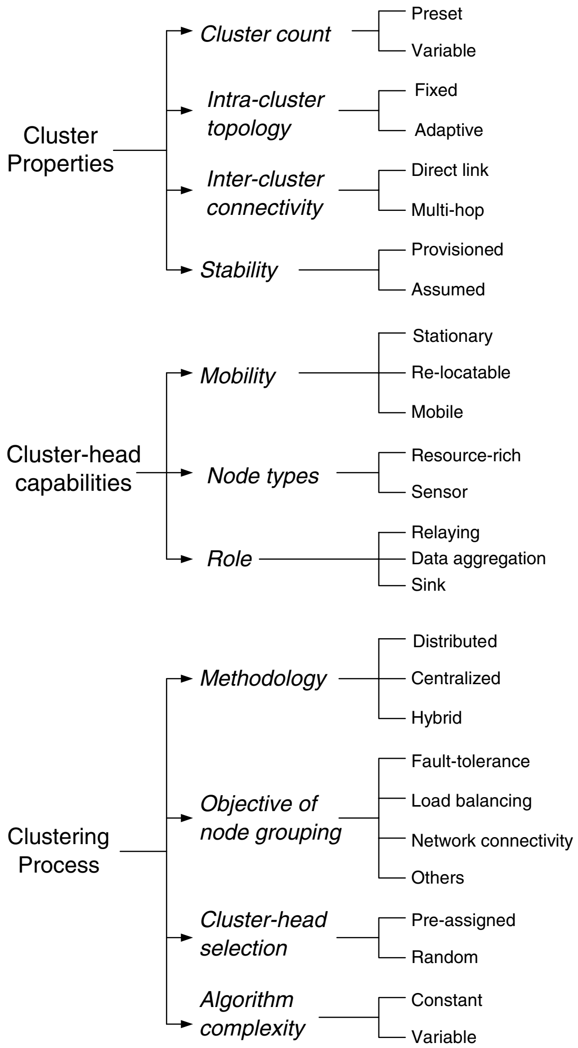

Fig. 2. Taxonomy of the different attributes of clustering of WSNs.

Linked cluster algorithm (LCA): The work of Baker and Ephremides [48,49] is among the early ones on clustering of wireless networks. The focus is mainly on forming an efficient network topology that can handle the mobility of nodes. By clustering, CHs are hoped to form a backbone network to which cluster members can connect while on the move. The Objective of the proposed distributed algorithm is to form clusters such that a CH is directly connected to all nodes in its cluster. LCA is thus geared for maximizing network connectivity. The algorithm assumes synchronized nodes and time-based medium access. A node is assigned the slot in the frame that matches its ID. First, each node broadcasts its ID and listens to transmission of other nodes. In the next round, a node broadcast the set of neighbors that it heard from and thus every node will eventually know its 1-hop and 2-hop neighbors. A node x becomes a CH if it has the highest ID among its neighbors *or* does not have the highest ID in its 1-hop neighborhood, but there exists at least one neighboring node y such that x is the highest ID node in y's 1-hop neighborhood. Since *LCA* is found to yield excessive number of clusters, the approach is refined in [50]. The idea is to pick a node x at random as the first CH and assign its neighbors to such first cluster. Then the node y with the lowest ID in the cluster is nominated as a CH. The neighbors of y that are not reachable to x would join the second cluster. The procedure is repeated for the third cluster and so on.

Adaptive clustering: In [51], Lin and Gerla studied the efficient support of multimedia applications in the general multi-hop mobile ad-hoc networks using CDMA based medium arbitration. To minimize the data delivery delay the network is clustered and distinct code is assigned to the cluster. Similar to [48] and [50], an ID-based cluster selection scheme is employed. Like LCA, a single-hop intra-cluster topology is established. A CH arbitrates the selection of communication codes with neighboring CHs. The algorithm strives to optimally control the cluster size by balancing the interest in the spatial reuse of channels, which is increased by having small clusters, and data delivery delay, which gets reduced by avoiding inter-cluster routing, i.e. large cluster sizes. Like LCA, TDMA is used for intra-cluster communication. However, every cluster would use a distinct code resulting is simplified implementation and great potential for meeting the QoS requirements often found in multimedia applications.

Random competition based clustering (RCC): Although RCC [52] is designed for mobile ad hoc networks, it is also applicable to WSNs. RCC mainly focuses at cluster stability in order to support mobile nodes. The RCC algorithm applies the *First Declaration Wins* rule, in which any node can "govern" the rest of the nodes in its radio coverage if it is the first to claim being a CH. After hearing the claim which is broadcasted by the first node, neighboring nodes join its cluster as member and give up their right to be a CH. To maintain clusters, every CH in the network broadcast a CH claim packet periodically. Since there is a time delay between broadcasting a claim packet and receiving it, concurrent broadcast can possibly create a conflict. Being unaware of on-going claims, many neighboring nodes may broadcast CH claim packets concurrently. To avoid such a problem RCC explicitly employs a random timer and uses the node ID for arbitration. Each node in the network reset its random time value, every time before broadcasting its CH claim packet. During this random time if it receives a broadcast message carrying CH claim packet from another node, it simply ceases the transmission of its CH claim. Since random timer is not a complete solution, RCC resolve further the concurrent broadcast problems by using the node ID. If the conflict persists, node having lower ID will become the CH. Although frequent node mobility still has direct effect, RCC is shown to be more stable than conventional clustering schemes such as [51]. A CH in adaptive clustering abandons its role when it hears a node with a lower ID, while, a CH in RCC only gives up its position when another CH moves near to it.

CLUBS: In [53], Nagpal and Coore proposed CLUBS, an algorithm that forms clusters through local broadcast and converge in a time proportional to the local density of nodes. Basically, cluster formation in CLUBS is based on the following three characteristics:

- Every node in the network must be connected to a cluster.
- Maximum diameter of all clusters in the network should be same.
- Clusters should support the intra-cluster communication, which means nodes in a cluster must be able to communicate with each others.

The algorithm forms clusters with a maximum of two hops. Each node in the network takes part in the cluster formation process by choosing a random number from a fixed integer range. Then it counts down from that number silently. If the count down was not interrupted from any other neighboring node and it reaches zero, it announces itself CH and broadcasts a "recruit" message. When a neighboring node receives the recruit message that comes within two-hop diameter boundary, it stops the count down, accepts the invitation and joins the cluster. A node that has joined a cluster is called "*follower*" is no longer allowed to compete for being a CH.

Since CLUBS allows overlapping, follower nodes keep listening to additional recruit messages and can be follower of more than on CH. If a node that is competing to become a CH detects a collision or received a garbled message, it becomes a follower node and assumes that multiple CHs attempted to recruit it at the same time. It can find out its CH later. The algorithm does not terminate unless all nodes in the network join some cluster as a CH or as a follower. Fig. 3, from [53], shows the final layout of the clustered network.

CLUBS can be implemented in the asynchronous environment without losing efficiency and simplicity. Furthermore, CLUBS satisfies many constraints that are common in other distributed environment such as limited/no topology knowledge or access to global IDs. The major problem of CLUBS algorithm is the clusters having their CHs within 1-hop range of each other. If this is the case, both clusters will collapse and CH election process will restart.

Hierarchical control clustering: Unlike most of the published schemes, the goal of Banerjee and Khuller is to form a multi-tier hierarchical clustering [37]. Fig. 4 illustrate the concept of hierarchy of clusters. A number of cluster's properties such as cluster size and the degree of overlap, which are useful for the management and scalability of

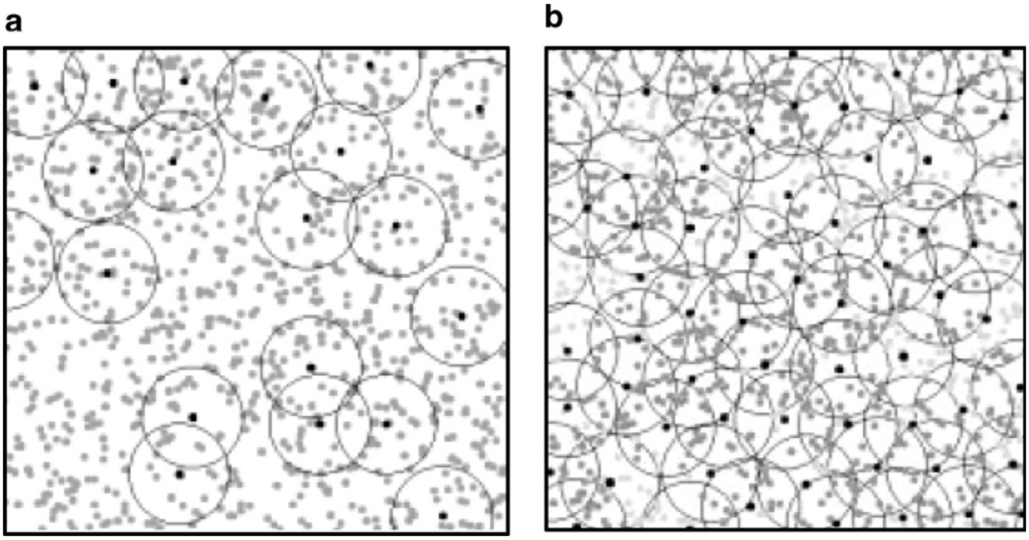

Fig. 3. (a) Leaders, dark spots, forms recruit nodes in their communication range (circles). (b) Final clusters formed. Nodes in the intersection of circles belong to more than one cluster.

the hierarchy, are also considered while grouping the nodes. In the proposed scheme, any node in the WSN can initiate the cluster formation process. Initiator with least node ID will take precedence, if multiple nodes started cluster formation process at the same time. 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*, broadcast 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. Broadcast signal carry the information such as source ID, parent ID, root ID, and 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 sub-tree size is $<2k$, or else, it will form multiple clusters. After the cluster creation phase, keeping cluster information is crucial for clusters while maintaining BFS tree is unimportant. This approach is shown to handle dynamic environments, e.g. presence of mobile nodes, very well.

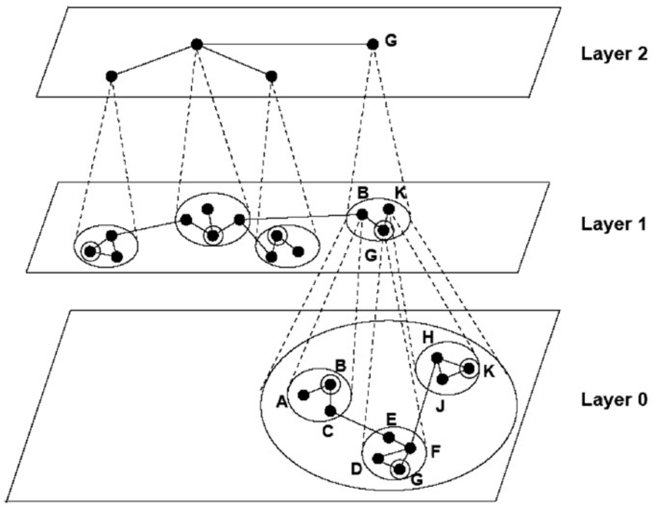

Fig. 4. An Example of a Three Layer Cluster Hierarchy (Figure is redrawn form [37]).

GS^3: In [54], Zhang and Arora present an algorithm, called GS^3, for self-configuring a wireless network into a cellular hexagon structure. The authors argue that ignoring the geographical boundary of clusters can be unwise; especially for very large network. They define the radius of the circle that contains all nodes in the cluster as a measure for the geometric size. A large cluster radius is said to increase energy consumption and reliability for intra-cell communication and limit the spatial reuse of radio frequencies in the network. Two kinds of nodes are assumed in the system: big and small. The big nodes are responsible for initiating the cluster formation process. In addition, they interface the small nodes to other cells and other network, e.g. the Internet and act as mediator. It should be noted that the cellular hexagon structure is virtual and is just used to guide the grouping and redistribution of nodes. To form the cellular hexagon structure, the area is divided into cells of equal radius R, as shown in Fig. 5. One of the big nodes starts the clustering process by selecting the heads of neighboring cells which select their neighbors and so on. Unselected members become cell members. Upon their selection cell heads relocate to the centers of their cells and start establishing their neighboring cells by selecting their heads. The process is repeated until no more cells could be added.

GS^3 differs from other distributed clustering by guaranteeing a predicable placement and number of CHs in a system. Unlike the hierarchical clustering algorithm of [37] in

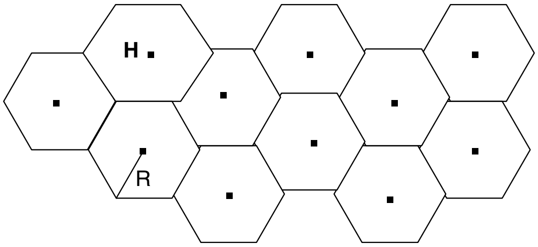

Fig. 5. The cellular hexagon virtual structure with a big node re-located to the center of a cell.

which convergence under perturbations requires multiple rounds of messages, GS^3 offers one-way diffusion within perturbed areas. Since, GS^3 uses geographic radius of cluster instead of logical radius, long intra-cluster links are possible and it guarantees the logical radius of clusters implicitly. GS^3 is a self-healing, and is thus applicable to static networks as well as to the networks with high degree of dynamics and mobility.

Energy Efficient Hierarchical Clustering (EEHC): Bandyopadhyay and Coyle [17] proposed EEHC; a distributed, randomized clustering algorithm for WSNs with the objective of maximizing the network lifetime. CHs collected the sensors' readings in their individual clusters and send an aggregated report to the base-station. Their technique is based on two stages; initial and extended. In the initial stage, also called single-level clustering, each sensor node announces itself as a CH with probability p to the neighboring nodes within its communication range. These CHs are named as the *volunteer CHs*. All nods that are within k hops range of a CH receive this announcement either by direct communication or by forwarding. Any node that receives such announcements and is not itself a CH becomes the member of the closest cluster. *Forced CHs* are nodes that are neither CHs nor belong to a cluster. If the announcement does not reach to a node within a preset time interval t that is calculated based on the duration for a packet to reach a node that is k hops away, the node will become a forced CH assuming that it is not within k hops of all volunteer CHs.

In the second stage, the process is extended to allow multi-level clustering, i.e. building h levels of cluster hierarchy. Like [37], the clustering process is recursively repeated at the level of CHs to form an additional tier. The algorithm opts to ensure h-hop connectivity between CHs and the base-station. Assumed that level h is highest, sensor nodes transmit the collected data to level-1 (lowest level) CHs. The CHs at the level-1 transmit the aggregated data to the level-2 CHs and so on. At the top level of the clustering hierarchy, CHs transmit the aggregated data report to the base station. EEHC has a time complexity of $O(k_1 + k_2 + \ldots + k_h)$, which is a significant improvement over the many $O(n)$ clustering algorithms such as LCA, and thus make it suitable for networks of large number of nodes. Energy consumption for network operations (e.g., sensor data collection, aggregated information transmission to base station) will depend on the parameters p and k of the algorithm. The authors derive mathematical expression the values of p and k that achieve minimal energy consumption. The derivation is based on periodic generation and transmission of sensors data and employs stochastic geometry to estimate communication energy. Simulation results confirmed that by using the optimal parameter values energy consumption in the network can be reduce significantly.

3.2. Constant convergence time algorithms

Clustering algorithms that converge completely in a fixed number of iterations, regardless of the size of the nodes population are called constant convergence time clustering algorithms. These algorithms usually pursue a localized strategy in which nodes execute the algorithm independently and base their cluster membership decisions on their own state and the state of their neighbors. In the balance of this section, we review a number of the published constant convergence time algorithms.

Low Energy Adaptive Clustering Hierarchy (LEACH): LEACH is one of the most popular clustering algorithms for WSNs [20]. It forms clusters based on the received signal strength and uses the CH nodes as routers to the base-station. 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. Initially a node decides to be a CH with a probability p and broadcasts its decision. Each non-CH node determines its cluster by choosing the CH that can be reached using the least communication energy. The role of being a CH is rotated periodically among the nodes of the cluster in order 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:

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

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

Since the decision to change the CH is probabilistic, there is a good chance that a node with very low energy gets selected as a CH. When this node dies, the whole cell becomes dysfunctional. Also, the CH is assumed to have a long communication range so that the data can reach the base-station from the CH directly. This is not always a realistic assumption since the CHs are regular sensors and the base-station is often not directly reachable to all nodes due to signal propagation problems, e.g., due to the presence of obstacles. LEACH also forms one-hop intra- and inter cluster topology where each node can transmit directly to the CH and thereafter to the base-station. Consequently, it is not applicable to networks deployed in large regions.

Fast Local Clustering service (FLOC): FLOC [30] is a distributed technique that produces approximately equal-sized clusters with minimum over-lap. The assumed radio model classifies nodes based on their proximity to the CH into inner (i-band) and outer (o-band). I-band nodes will suffer very little interference communicating with the CH, while message from o-band nodes may be lost. FLOC favors i-band membership in order to increase the robustness of the intra-cluster traffic. Fig. 6 summarizes the FLOC algorithm. A node stays *idle* waiting for some random duration to receive an invitation from any potential CH. If the node gets no invitation, it becomes a *candidate* CH and broadcasts a candidacy message (transition 1). Upon hearing the candidacy message a recipient node "k" that is already an i-band member of a cluster C_k, will reply back to inform the candidate CH about such membership. The candidate CH will then realize the conflict and join C_k as an o-band node (transition 3). If the candidate CH receives no conflict messages, it becomes a CH and starts inviting members to its cluster (transition 4). An idle node would join a cluster as an o-band node (transition 5) if it does not receive an invitation from a closer CH (transition 2). That decision can be changed, if the node later receives an invitation from a closer CH, i.e. the node switch its membership to a better cluster (transition 6).

FLOC scales very well converging in a constant time is $O(1)$, regardless the size of the network. It also exhibits self-healing capabilities since o-band nodes can switch to an i-band node in another cluster. In addition, new nodes can execute the algorithm and either joins one of the existing clusters or forms a new one that possibly would attract some of the current o-band nodes in neighboring clusters. It is unclear though, how the data are disseminated across clusters.

Algorithm for Cluster Establishment (ACE): Unlike other distributed clustering schemes, ACE employs an emergent algorithm [55]. Emergent algorithms much like artificial neural networks evolve to optimal solution through a mix of local optimization steps. The main idea of ACE is to allow a node to assess its potential as a CH before becoming one and stepping down if it is not the best CH at the moment. The algorithm works in iterations that do not have to be synchronized at the individual nodes. S*pawning* new clusters and *migration of* existing ones are the two functional components of ACE. A node spawns of new cluster when it decides to become a CH. It broadcasts an invitation message to recruit its neighbors. 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. However, the node can be a loyal follower, i.e. a member, of only a single cluster.

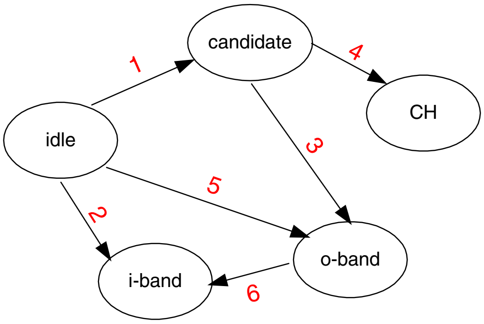

Fig. 6. State transition for the FLOC clustering algorithm (redrawn from [30]).

Migration is a process in which the best candidate for being CH is 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 existing clusters will be considered as the best candidate for CH. The overall effect would appear as clusters are applying a repulsive force to spread out and reduce their overlap. Fig. 7, which is taken from [55], shows the progression of the ACE algorithm after 3-iterations and compares ACE to a simple node-ID based clustering such as LCA [48]. It is worth noting that ACE covers the entire network just in three rounds and uses only intra-cluster communications. It converges in a constant time $O(d)$ where d is the node density per unit disk. An enhancement to the migration process of ACE was proposed in [56]. The idea is to further iterate in order to increase the regularity of cluster layout. In addition to the repulsion effect, an attraction between clusters that far apart is provisioned by factoring in the degree of overlap between neighboring clusters.

Like GS^3 [54], ACE increased the spatial coverage in the network since it increases the separation among clusters in areas where the degree of nodes is high, while allows cluster overlap where degree of nodes is low. Such an approach allows spreading the clusters according to the node density throughout the area of interest. Experimental validation of ACE indicated that it achieves low variance and high average of cluster sizes when compared to node-ID based schemes like LCA [48] and [51]. In addition, ACE can easily repair structure damage in the network caused by node failure and can also integrate new nodes.

Hybrid Energy-Efficient Distributed Clustering (HEED): HEED [19] is a distributed clustering scheme in which CH nodes are picked from the deployed sensors. HEED considers a hybrid of energy and communication cost when selecting CHs. Unlike LEACH, it does not select cell-head nodes randomly. Only sensors that have a high residual energy can become cell-head nodes. HEED has three main characteristics:

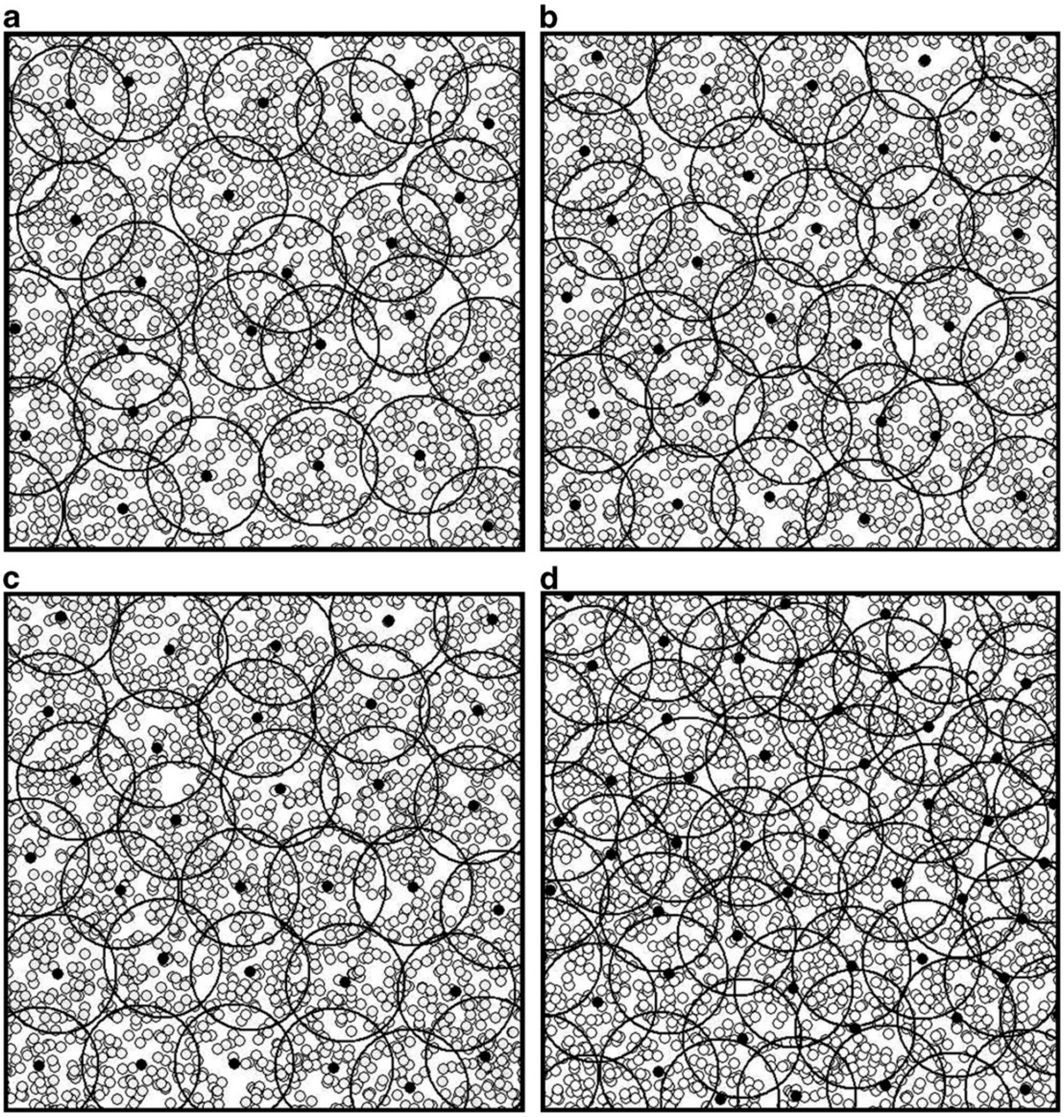

Fig. 7. Illustration of the topological layout of clusters progressively achieved by ACE as compared to Node-ID passed clustering algorithms.

- The probability that two nodes within each other's transmission range becoming CHs is small. Unlike LEACH, this means that CHs are well distributed in the network.
- Energy consumption is not assumed to be uniform for all the nodes.
- For a given sensor's transmission range, the probability of CH selection can be adjusted to ensure inter-CH connectivity.

In HEED, each node is mapped to exactly one cluster and can directly communicate with its CH. The algorithm is divided into three phases:

1. *Initialization phase:* The algorithm first sets an initial percentage of CHs among all sensors. This percentage value, C_{prob}, is used to limit the initial CH announcements to the other sensors. Each sensor sets its probability of becoming a cluster-head, CH_{prob}, as follows: $CH_{\text{prob}} = C_{\text{prob}} * E_{\text{residual}}/E_{\max}$, 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 $E_{\max}$.
2. *Repetition phase:* During this phase, every sensor goes through several 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 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 2 types of cell-head status that a sensor could announce to its neighbors:

- *Tentative status:* 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.
- *Final status:* The sensor permanently becomes a CH if its CH_{prob} has reached 1.

3. *Finalization phase:* During this phase, each sensor makes a final decision on its status. It either picks the least cost CH or pronounces itself as CH.

Huang and Wu extended the basic HEED algorithm [57] by not giving up on nodes that did not hear from any CH (orphaned nodes). Per the finalization phase above, these nodes become CHs themselves. Instead, the modified version opts to re-execute the algorithm for just those orphaned nodes. Such slight modification is shown to significantly decrease the CH count which would reduce the size of the routing tree needed during inter-CH communication and thus limit the data collection latency.

Distributed Weight-Based Energy-Efficient Hierarchical Clustering (DWEHC): Ding et al. [58] have proposed DWEHC to achieve more aggressive goals than those of HEED. Basically, generating balanced cluster sizes and optimizing the intra-cluster topology. DWEHC proceeds in a distributed manner and has $O(1)$ time complexity. Each sensor calculates its weight after locating the neighboring nodes in its area. The weight is a function of the sensor's energy reserve 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 since they have a direct link to the CH. A node progressively adjusts such membership in order 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. It is worth noting that by doing so the node may switch to a CH other than its original one. The process continues until nodes settles on the most energy efficient intra-cluster topology. To limit the number of levels, every cluster is assigned a range within which member nodes should lay. Fig. 8, redrawn from [58], illustrates the structure of the intra-cluster topology.

Both DWEHC and HEED are similar in many ways; every node in the network participates in the clustering process, they do not make any assumption about the network size and consider energy reserve in CH selection. Despite such resemblances, there are many performance differences between DWEHC and HEED. For example, clusters generated by DWEHC are more well-balanced than HEED. DWEHC also achieves significantly lower energy consumption in intra-cluster and inter-cluster communication than HEED.

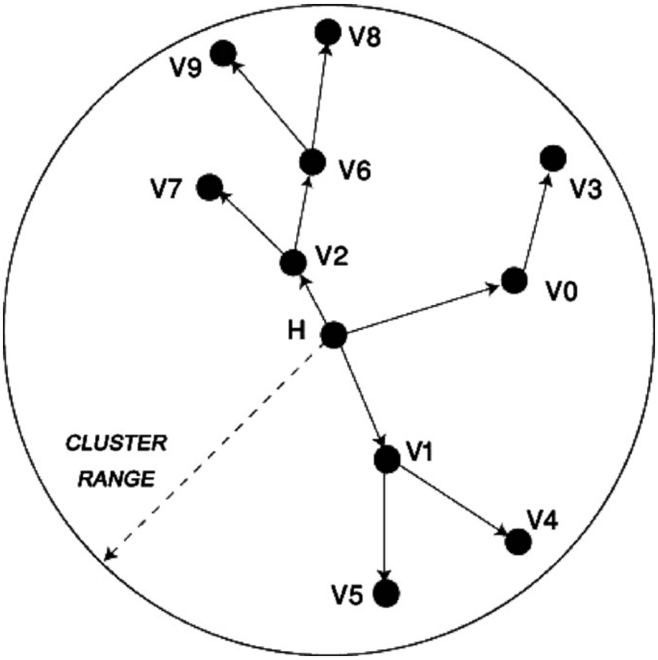

Fig. 8. DWEHC generates a multi-hop intra-cluster topology with the CH at the root and member nodes are ordered in breadth-first order.

MOCA: Most of the published clustering algorithms strive to generate the minimum number of disjoint clusters. However, Youssef et al. [59] argued that guaranteeing some degree of overlap among clusters can facilitate many applications like inter-cluster routing, topology discovery and node localization and recovery from cluster head failure, etc. They proposed MOCA, a randomized, distributed Multi-hop Overlapping Clustering Algorithm 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 in order 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 nomination probability (p) is used to control the number of clusters in the network and the degree of overlap among them. The authors conducted extensive simulation to guide the selection of appropriate value of p in order to achieve particular cluster count and overlapping degree.

Attribute-based clustering: Wang et al. [60] promoted the idea of clustering the WSN based on the queries and attributes of the data. The main motive is to achieve efficient dissemination of data in the network. The concept resembles the data-centric design model of WSNs. The clustering would be established by mapping a hierarchy of data attri-

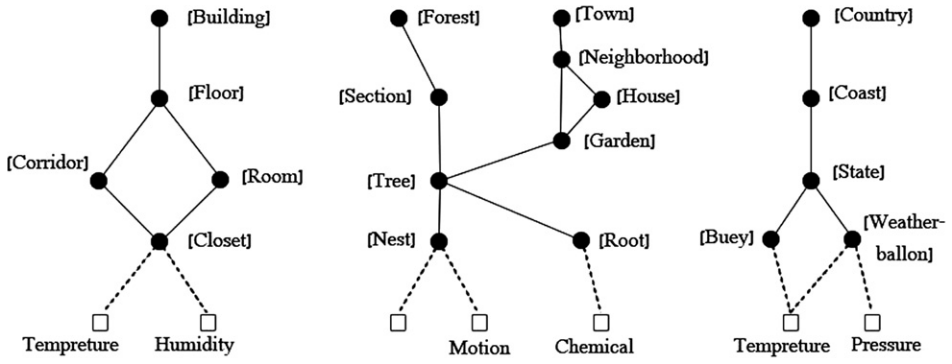

Fig. 9. An example of attributes hierarchy.

butes to the network topology. Fig. 9, redrawn from [60], shows an example of possible attribute hierarchies. The approach is based on the well know leader election algorithm. The base-station starts the process by asking nodes to form clusters. Those nodes that hear the request decide whether they should nominate themselves as CHs based on their energy. After receiving the base-station request, sensor nodes having intention to become CH wait for a random time period that is based on the remaining battery supply. Nodes with more energy wait longer. If a node nominates itself, it broadcasts an announcement that further gets spread from node to node. A node later joins the CH that can reach over the least number of hops. During the wait time, if a node hears a CH claim packet from a neighboring node it drops its CH bid and resends the received packet after incrementing the hop count field in the packet by one. Upon hearing a CH announcement from a node whose attribute is different, the recipient node establishes a new cluster for that attribute and becomes a CH.

The approach also promotes the rotation of the CH role among the nodes of the cluster in order to prolong the node's lifespan. Failure of CHs can also be easily detected. A CH periodically sends a heartbeat message to members. If neighbors do not get the periodic update, they will assume that CH is dysfunctional and one of them will assume the role of CH. If the network is dense enough, CHs can monitor each other and perform the recovery at the level of CHs. The periodic heartbeat can also facilitate the integration of new nodes. Table 1 compares the

Table 1
Comparison of the presented clustering algorithms

Clustering approaches	Convergence time	Node mobility	Cluster overlapping	Location awareness	Energy efficient	Failure recovery	Balanced clustering	Cluster stability
LCA	Variable $O(n)$	Possible	No	Required	No	Yes	OK	Moderate
Adaptive clustering	Variable $O(n)$	Yes	No	Required	N/A	Yes	OK	Low
CLUBS	Variable $O(n)$	Possible	High	Not required	N/A	Yes	OK	Moderate
Hierarchical control clustering	Variable $O(n)$	Possible	Low	Not required	N/A	Yes	Good	Moderate
RCC	Variable $O(n)$	Yes	No	Required	N/A	Yes	Good	Moderate
GS^3	Variable $O(n)$	Possible	Low	Required	N/A	Yes	Good	Moderate
EEHC	Variable $O(k1+k2+\ldots+kh)$	No	No	Required	Yes	N/A	OK	N/A
LEACH	Constant $O(1)$	Fixed BS	No	Not required	No	Yes	OK	Moderate
FLOC	Constant $O(1)$	Possible	No	Not required	N/A	Yes	Good	High
ACE	Constant $O(d)$	Possible	Very Low	Not required	N/A	Yes	Good	High
HEED	Constant $O(1)$	Stationary	No	Not required	Yes	N/A	Good	High
Extended HEED	Constant $O(1)$	Stationary	No	Not required	Yes	N/A	Very good	High
DWEHC	Constant $O(1)$	Stationary	No	Required	Yes	N/A	Very good	High
MOCA	Constant $O(1)$	Stationary	Yes	Not required	Yes	N/A	Good	High
Attribute-based clustering	Constant $O(1)$	No	No	Required	Yes	Yes	Very good	High

Table 2
Classification of survey algorithms based on clustering attributes

Clustering approaches	Cluster properties				Cluster head capabilities			Clustering process			
	Cluster count	Intra-cluster topology	Inter-cluster connectivity	Stability	Mobility	Node type	Role	Methodology	Objective of node grouping	CH selection	Algorithm complexity
[16]	Fixed	Fixed (1-Hop)	Direct link/multi-hop	Assumed	Stationary	Resource-rich	Sink	Centralized	Load balancing	Pre-assigned	Constant
[34]	Fixed	Fixed (1-Hop)	Direct link/multi-hop	Provisioned	Stationary	Resource-rich	Sink	Centralized	Fault-tolerance	Pre-assigned	Constant
[39]	Variable	Fixed (1-Hop)	Direct link/multi-hop	Assumed	Stationary	Resource-rich	Sink	Centralized	Minimal cluster count & energy	Pre-assigned	Variable
LCA	Variable	Fixed (1-Hop)	Direct link/multi-hop	Provisioned	Mobile	Sensor	Aggregation	Distributed	Connectivity	Random	Variable
Adaptive clustering	Variable	Fixed (1-Hop)	Direct link/multi-hop	Provisioned	Mobile	Sensor	Relaying	Distributed	Bandwidth gain & QoS	Random	Variable
RCC	Variable	Adaptive	Direct link	Provisioned	Mobile	Sensor	Relaying	Hybrid	Stability & simplicity	Random	Variable
CLUBS	Variable	Fixed (2-Hop)	Multi-hop	Assumed	Re-locatable	Sensor	Aggregation & relaying	Distributed	Management & scalability	Random	Variable
Hierarchical control clustering	Variable	Adaptive	Multi-hop (hierarchical)	Provisioned	Re-locatable	Sensor	Relaying	Distributed	Management & scalability	Random	Variable
GS^3	Preset	Adaptive	Direct link	Provisioned	Re-locatable	Resource-rich	Relaying	Distributed	Scalability & fault tolerance	Pre-assigned	Variable
EEHC	Variable	Adaptive	Direct link/multi-hop (hierarchical)	Assumed	Stationary	Sensor	Aggregation & relaying	Distributed	Save energy	Random	Variable
LEACH	Variable	Fixed (1-Hop)	Direct link	Provisioned	Stationary	Sensor	Relaying	Distributed	Save energy	Random	Constant
FLOC	Variable	Fixed (2-Units)	Direct link	Provisioned	Re-locatable	Sensor	Aggregation & relaying	Distributed	Scalability & fault tolerance	Random	Constant
ACE	Variable	Adaptive	Direct link	Provisioned	Re-locatable	Sensor	Aggregation & relaying	Distributed	Scalability & load balancing	Random	Constant
HEED	Variable	Fixed (1-Hop)	Direct link/multi-hop	Assumed	Stationary	Sensor	Aggregation & relaying	Distributed	Save energy	Random	Constant
Extended HEED	Variable	Fixed (1-Hop)	Direct link/multi-hop	Assumed	Stationary	Sensor	Aggregation & relaying	Distributed	Save energy	Random	Constant
DWEHC	Variable	Adaptive (Multi-level)	Direct link	Provisioned	Stationary	Sensor	Aggregation & relaying	Distributed	Save energy	Random	Constant
MOCA	Variable	Fixed (k-Hop)	Direct link/multi-hop	Assumed	Stationary	Sensor	Aggregation & relaying	Distributed	Overlapping & connectivity	Random	Constant
Attribute-based clustering	Variable	Fixed	Direct link/multi-hop	Distributed	Stationary	Sensor	Aggregation & relaying	Distributed	Bandwidth gain	Random	Constant

algorithms discussed in this section. Table 2 classifies these algorithms based on the taxonomy of attributes discussed in section 2.

4. Conclusion

Wireless sensor networks (WSNs) have attracted significant attention over the past few years. A growing list of civil and military applications can employ WSNs for increased effectiveness; especially in hostile and remote areas. Examples include disaster management, border protection, combat field surveillance. In these applications a large number of sensors are expected, requiring careful architecture and management of the network. Grouping nodes into clusters has been the most popular approach for support scalability in WSNs. Significant attention has been paid to clustering strategies and algorithms yielding a large number of publications. In this paper, we surveyed the state of the research and classified the different schemes. We developed taxonomy of relevant attributes. We categorized the different schemes according the objectives, the desired cluster properties and clustering process. We highlighted the effect of the network model on the pursued approaches and summarized a number of schemes, stating their strength and limitations.

References

[1] I.F. Akyildiz et al., Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.

[2] C-Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges, Proceedings of the IEEE 91 (8) (2003) 1247–1256.

[3] D. Estrin, et al., Next century challenges: scalable coordination in sensor networks, in: Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99), Seattle, Washington, August 1999.

[4] G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors, Communications of the ACM 43 (5) (2000) 51–58.

[5] H. Wang, et al., Target classification and localization in habitat monitoring, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), Hong Kong, China. April 2003.

[6] J.M. Rabaey et al., PicoRadio supports ad hoc ultra low power wireless networking, IEEE Computer 33 (2000) 42–48.

[7] R.H. Katz, J.M. Kahn, K.S.J. Pister, Mobile networking for smart dust, in: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, WA, August 1999.

[8] K. Sohrabi et al., Protocols for self-organization of a wireless sensor network, IEEE Personal Communications 7 (5) (2000) 16–27.

[9] R. Min, et al., Low power wireless sensor networks, in: Proceedings of International Conference on VLSI Design, Bangalore, India, January 2001.

[10] R. Burne, et. al., A self-organizing, cooperative UGS network for target tracking, in: Proceedings of the SPIE Conference on Unattended Ground Sensor Technologies and Applications II, Orlando Florida, April 2000.

[11] V. Kawadia, P.R. Kumar, Power control and clustering in Ad Hoc networks, in: Proceedings of IEEE INFOCOM, San Francisco, CA, March 2003.

[12] M. Chatterjee, S.K. Das, D. Turgut, WCA: a Weighted Clustering Algorithm for mobile Ad Hoc networks, Cluster Computing 5 (2) (2002) 193–204.

[13] A.D. Amis, R. Prakash, T.H.P. Vuong, D.T. Huynh, Max-Min D-cluster formation in wireless Ad Hoc networks, in: Proceedings of IEEE INFOCOM, March 2000.

[14] A.B. McDonald, T. Znati, A mobility based framework for adaptive clustering in wireless ad-hoc networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1466–1487.

[15] S. Basagni, Distributed clustering algorithm for ad-hoc networks, in: Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN), Fremantle, Australia, June 1999.

[16] G. Gupta, M. Younis, Load-balanced clustering in wireless sensor networks, in: Proceedings of the International Conference on Communication (ICC 2003), Anchorage, Alaska, May 2003.

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

[18] S. Ghiasi, A. Srivastava, X. Yang, M. Sarrafzadeh, Optimal energy aware clustering in sensor networks, Sensors Magazine MDPI 1 (1) (2004) 258–269.

[19] O. Younis, S. Fahmy, HEED: A Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks, IEEE Transactions on Mobile Computing 3 (4) (2004) 366–379.

[20] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, Application specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Networking (2002).

[21] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, Elsevier Journal of Ad Hoc Networks 3 (3) (2005) 325–349.

[22] M. Younis, M. Youssef, K. Arisha, Energy-aware management in cluster-based sensor networks, Computer Networks 43 (5) (2003) 649–668.

[23] Y.T. Hou, Y. Shi, H.D. Sherali, On energy provisioning and relay node placement for wireless sensor networks, IEEE Transactions on Wireless Communications 4 (5) (2005) 2579–2590.

[24] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01), Rome, Italy, July 2001.

[25] M. Adamou, I. Lee, I. Shin, An energy efficient real-time medium access control protocol for wireless ad-hoc networks, in: WIP Session of IEEE Real-time Systems Symposium (RTSS'01), London, UK, December 2001.

[26] T. Wu, S. Biswas, A self-reorganizing slot allocation protocol for multi-cluster sensor networks, in: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN 2005), April 2005.

[27] G. Jolly, M. Younis, An energy efficient, scalable and collision less MAC layer protocol for wireless sensor networks, Wireless Communications and Mobile Computing 5 (3) (2005) 285–304.

[28] K. Dasgupta, K. Kalpakis, P. Namjoshi, An efficient clustering–based heuristic for data gathering and aggregation in sensor networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC, 2003), New Orleans, LA, March 2003.

[29] L. Subramanian, R.H. Katz, An architecture for building self configurable systems, in: Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA, August 2000.

[30] M. Demirbas, A. Arora, V. Mittal, FLOC: a fast local clustering service for wireless sensor networks, in: Proceedings of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS'04), Palazzo dei Congressi, Florence, Italy, June 2004.

[31] P. Ding, J. Holliday, A. Celik., Distributed energy efficient hierarchical clustering for wireless sensor networks, in: Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'05), Marina Del Rey, CA, June 2005.
[32] K. Wang, S. Abu Ayyash, T.D.C. Little, P. Basu, Attribute-based clustering for information dissemination in wireless sensor networks, in: Proceeding of 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2005), Santa Clara, CA, September 2005.
[33] B. Krishnamachari, D. Estrin, S. Wicker, Modeling data centric routing in wireless sensor networks, in: Proceedings of IEEE INFOCOM, New York, NY, June 2002.
[34] G. Gupta, M. Younis, Fault-tolerant clustering of wireless sensor networks, in: Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2003), New Orleans, Louisiana, March 2003.
[35] S. Lindsey, C.S. Raghavendra, PEGASIS: power efficient gathering in sensor information systems, in: Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 2002.
[36] S. Lindsey, C.S. Raghavendra, K. Sivalingam, Data gathering in sensor networks using the energy*delay metric, in: Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.
[37] S. Banerjee, 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 (INFOCOM'01), Anchorage, AK, April 2001.
[38] M. Younis, K. Akkaya, A. Kunjithapatham, Optimization of task allocation in a cluster–based sensor network, in: Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC'2003), Antalya, Turkey, June 2003.
[39] E. Ilker Oyman, Cem 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.
[40] F. Dai, J. Wu, On constructing k-connected k-dominating set in wireless networks, in: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), Denver, Colorado, April 2005.
[41] F. Garcia, J. Solano, I. Stojmenovic, Connectivity based k-hop clustering in wireless networks, Telecommunication Systems 22 (1) (2003) 205–220.
[42] Y. Fernandess, D. Malkhi,. K-clustering in wireless ad hoc networks, in: Proceedings of the 2nd ACM international Workshop on Principles of Mobile Computing (POMC '02), Toulouse, France, October 2002.
[43] A.D. Amis, R. Prakash, T.H.P. Vuong, D.T. Huynh, Max-Min D-cluster formation in wireless ad hoc networks, in: Proceedings of 20th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2000), March 2000.
[44] Y.T. Hou, Y. Shi, H.D. Sherali, On energy provisioning and relay node placement for wireless sensor networks, in: IEEE Transactions on Wireless Communications, vol. 4, Mo. 5, September 2005, pp. 2579–2590.
[45] K. Dasgupta, M. Kukreja, K. Kalpakis, Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks, in: Proceedings of 8th IEEE Symposium on Computers and Communication (ISCC'03), Kemer-Antalya, Turkey, July 2003.
[46] T. Moscibroda, R. Wattenhofer, Maximizing the lifetime of dominating sets, in: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), Denver, Colorado, April 2005.
[47] R. Khanna, H. Liu, H.H. Chen, Self-organization of sensor networks using genetic algorithms, in: Proceedings of the 32nd IEEE International Conference on Communications (ICC'06), Istanbul, Turkey, Jun. 2006.
[48] D.J. Baker, A. Ephremides, The architectural organization of a mobile radio network via a distributed algorithm, IEEE Transactions on Communications, COM-29 (11) (1981) 1694–1701.
[49] D.J. Baker, A. Ephremides, J.A. Flynn, The design and simulation of a mobile radio network with distributed control, IEEE Journal on Selected Areas in Communications (1984) 226–237.
[50] A. Ephremides, J.E. Wieselthier, D.J. Baker, A design concept for reliable mobile radio networks with frequency hopping signaling, Proceedings of IEEE 75 (1) (1987) 56–73.
[51] C.R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, IEEE Journal on Selected Areas Communications 15 (7) (1997) 1265–1275.
[52] K. Xu, 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.
[53] R. Nagpal, D. Coore, An algorithm for group formation in an amorphous computer, in: Proceedings of the 10th International Conference on Parallel and Distributed Systems (PDCS'98), Las Vegas, NV, October 1998.
[54] H. Zhang, A. Arora, GS^3: 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.
[55] H. Chan, 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.
[56] H. Chan, M. Luk, A. Perrig, Using clustering information for sensor network localization, in: Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS'05), Marina del Rey, CA, USA, June 2005.
[57] H. Huang, J. Wu, A probabilistic clustering algorithm in wireless sensor networks, in: Proceeding of IEEE 62nd Semiannual Vehicular Technology Conference (VTC), Dallas, TX September 2005.
[58] P. Ding, J. Holliday, A. Celik, Distributed energy efficient hierarchical clustering for wireless sensor networks, in: Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems(DCOSS'05), Marina Del Rey, CA, June 2005.
[59] A. Youssef, M. Younis, M. Youssef, A. Agrawala, Distributed formation of overlapping multi-hop clusters in wireless sensor networks, in: Proceedings of the 49th Annual IEEE Global Communication Conference (Globecom'06), San Francisco, CA, November 2006.
[60] K. Wang, S. Abu Ayyash, T.D.C. Little, P. Basu, Attribute-based clustering for information dissemination in wireless sensor networks, in: Proceeding of 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'05), Santa Clara, CA, September 2005.

Ameer Ahmed Abbasi was born in Karachi, Pakistan. He went to the Sheikh Zayed Research Centre, University of Karachi, where he studied IS with computer technology and obtained his Master's degree in 2000. He worked one and half years for the Usman Business Solution Inc. Karachi, Pakistan as a software engineer. In 2001 he moved to the Al-Hussan Institute of Management & Computer Science, Dammam, Saudi Arabia. He is teaching computer science courses and participating in research in the fields of mobile computing, ad-hoc and wireless sensor networks. His e-mail address is: ameer_abbasi@hussan.edu.sas.

Mohamed Younis received B.S. degree in computer science and M.S. in engineering mathematics from Alexandria University in Egypt in 1987 and 1992, respectively. In 1996, he received his Ph.D. in computer science from New Jersey Institute of Technology. He is currently an associate professor in the department of computer science and electrical engineering at the university of Maryland Baltimore County (UMBC). Before joining UMBC, he was with the Advanced Systems Technology Group, an Aerospace Electronic Systems R&D organization of Honeywell International Inc. While at Honeywell he led multiple projects for building integrated fault tolerant avionics, in which a novel architecture and an operating system were developed. This new technology has been incorporated by Honeywell in multiple products and has received worldwide recognition by both the research and the engineering communities. He also participated in the development the Redundancy Management System, which is a key component of the Vehicle and Mission Computer for NASA's X-33 space launch vehicle. Dr. Younis' technical interest includes network architectures and protocols, embedded systems, fault tolerant computing and distributed real-time systems. Dr. Younis has four granted and three pending patents. He served on multiple technical committees and published over 85 technical papers in refereed conferences and journals.