A Survey of Clustering Algorithms for Wireless Sensor Networks

D. J. Dechene, A. El Jardali, M. Luccini, and A. Sauer.
Department of Electrical and Computer Engineering
The University Of Western Ontario
London, Ontario, Canada
{ddechene, aeljarda, mluccini, asauer2}@uwo.ca

Abstract— **In this paper, we examine currently proposed clustering algorithms for Wireless Sensor Networks. We will briefly discuss the operations of these algorithms, as well as draw comparisons on the performance between the various schemes. Specifically, we will examine the performance in terms of the power and quality aspects of these schemes. We also discuss improvements to be made for future proposed clustering schemes. This paper should provide the reader with a basis for research in clustering schemes for Wireless Sensor Networks.**

I. Introduction

With the continued advances in Micro-Electro-Mechanical Systems (MEMS), Wireless Sensor Networks (*WSNs*) have and will play a vital role in our daily lives. Humans have relied on wired sensors for years, for simple tasks such as temperature monitoring, to complex tasks such as monitoring life-signs in hospital patients.

Wireless Sensor Networks provide unforseen applications in this new field of design [1]. From military applications such as battlefield mapping and target surveillance, to creating context-aware homes [2] where sensors can monitor safety and provide automated services tailored to the individual user; the number of applications are endless. Smart Dust is an example of one such application [3], [4], [5]. However this new technology poses many design goals, [1] that up until recently, have not been considered feasible for these applications.

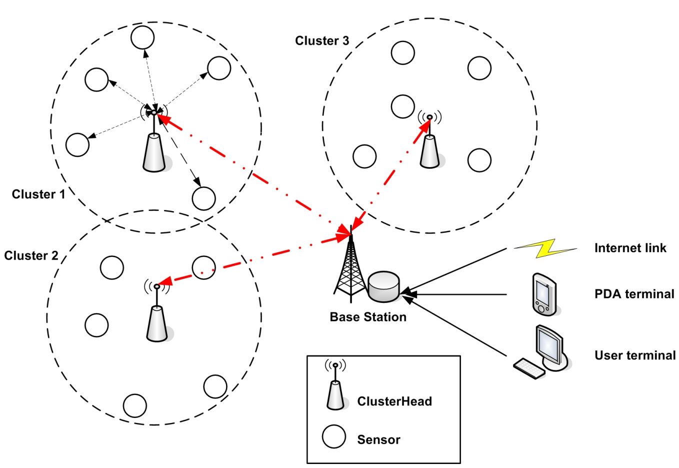

Fig. 1. General Sensor Network Architecture

One such problem is how to create an organizational structure amongst these nodes [8]. Since the fundamental advantage of WSNs is the ability to deploy them in an ad hoc manner, as it is not feasible to organize these nodes into groups pre-deployment. For this reason, there has been an large amount of research into ways of creating these organizational structures (or *clusters*) [6], [7], [8]. Looking at *Fig. 1*, we can see the architecture of a generic Wireless Sensor Network [1], and examine how the clustering phenomenon is an essential part of the organizational structure.

- *Sensor Node:* A sensor node is the core component of a WSN. Sensor nodes can take on multiple roles in a network, such as simple sensing; data storage; routing; and data processing.
- *Clusters:* Clusters are the organizational unit for WSNs. The dense nature of these networks require the need for them to be broken down into clusters to simplify tasks such a communication.
- *Clusterheads:* Clusterheads are the organization leader of a cluster. They often are required to organize activities in the cluster. These tasks include but are not limited to data-aggregation and organizating the communication schedule of a cluster.
- *Base Station:* The base station is at the upper level of the hierarchical WSN. It provides the communication link between the sensor network and the end-user.
- *End User:* The data in a sensor network can be used for a wide-range of applications. [1] Therefore, a particular application may make use of the network data over the internet, using a PDA, or even a desktop computer. In a queried sensor network (where the required data is gathered from a query sent through the network). This query is generated by the end user.

The clustering phenomenon as we can see, plays an important role in not just organization of the network, but can dramatically affect network performance. There are several key limitations in WSNs, that clustering schemes must consider.

- *Limited Energy:* Unlike wired designs, wireless sensor nodes are ”off-grid”, meaning that they have limited energy storage and the efficient use of this energy will be vital in determining the range of suitable applications for

these networks. The limited energy in sensor nodes must be considered as proper clustering can reduce the overall energy usage in a network.
- *Network Lifetime:* The energy limitation on nodes results in a limited network lifetime for nodes in a network. Proper clustering should attempt to reduce the energy usage, and hereby increase network lifetime.
- *Limited Abilities:* The small physical size and small amount of stored energy in a sensor node limits many of the abilities of nodes in terms of processing and communication abilities. A good clustering algorithm should make use of shared resources within an organizational structure, while taking into account the limitation on individual node abilities [8].
- *Application Dependency:* Often a given application will heavily rely on cluster organization. When designing a clustering algorithm, application robustness must be considered as a good clustering algorithm should be able to adapt to a variety of application requirements.

The rest of this paper is organized in the following manner: Section II will introduce the main design goals of clustering. Section III will provide an overview of proposed algorithms while Section IV will draw comparisons on these schemes in terms of power and quality. We will conclude this paper with Section V, in which we will examine some future research problems and draw conclusions on the current state of sensor network clustering.

II. Design philosophy

Wireless Sensor Networks present a vast challenges in terms of implementation. Design goals targeted in traditional networking provide little more than a basis for the design in wireless sensor networks [9], [10], [11]. Clustering algorithms play a vital role in achieving the targeted design goals for a given implementation. There are several key attributes that designers must carefully consider, which are of particular importance in wireless sensor networks.

- *Cost of Clustering:* Although clustering plays a vital role in organizing sensor network topology, there are often many resources such as communication and processing tasks needed in the creation and maintenance of the clustering topology. Such costs as the required resources are not being used for data transmission or sensing tasks.
- *Selection of Clusterheads and Clusters:* The clustering concept offers tremendous benefits for wireless sensor networks. However when designing for a particular application, designers must carefully examine the formation of clusters in the network. Depending on the application, certain requirements for the number of nodes in a cluster or its physical size may play an important role in its operation. This prerequisite may have an impact on how clusterheads are selected in this application.
- *Real-Time Operation:* Useful lifetime of data is also a fundamental criterion in designing Wireless Sensor Networks. In applications such as habitat monitoring [12], [13], simply receiving data is sufficient for analysis, meaning delay is not an important issue. When we look at a military tracking [14], the issue of real-time data acquisition becomes much more vital. When looking at clustering algorithms, important attention must be paid to the delay created by the clustering scheme itself. In addition, the time required for cluster recovery mechanisms must also be taken into account.
- *Synchronization:* One of the primary limitations in Wireless Sensor Networks is the limited energy capacity of nodes. Slotted transmission schemes (such as TDMA), allow nodes to regularly schedule sleep intervals to minimize energy used. Such schemes require synchronization mechanisms to setup and maintain the transmission schedule. When considering a clustering scheme, synchronization and scheduling will have a considerable effect on network lifetime and the overall network performance.
- *Data Aggregation:* One major advantage of wireless sensor networks is the ability for data aggregation to occur in the network. In a densely populated network there are often multiple nodes sensing similar information. Data aggregation allows the differentiation between *sensed data* and *useful data*. In-network processing makes this process possible and now it is fundamental in many sensor network schemes [15], as the power required for processing tasks is substantially less than communication tasks. As such, the amount of data transferred in-network should be minimized. Many clustering schemes provide data aggregation capabilities [15], and as such, the requirement for data aggregation should be carefully considered when selecting a clustering approach.
- *Repair Mechanisms:* Due to the nature of Wireless Sensor Networks, they are often prone to node mobility, node death and interference. All of these situations can result in link failure. When looking at clustering schemes, it is important to look at the mechanisms in place for link recovery and reliable data communication.
- *Quality of Service (QoS):* From an overall network standpoint, we can look at QoS requirements in Wireless Sensor Networks. Many of these requirements are application dependant (such as acceptable delay and packet loss tolerance), and as such, it is important to look at these metrics when choosing a clustering scheme. Implementations can vary widely in terms of these metrics, and as a result, the design process should consider these aspects.

III. Overview of Proposed Algorithms

A. Heuristic Algorithms

An heuristic algorithm is an algorithm that usually has one or both of the following goals in solving a problem:

- Finding an algorithm with reasonable run-time (time needed to set up clusters is affordable); and/or
- With finding the optimal solution

This means that an heuristic algorithm leads to reasonable performance and is not based on particular metrics.

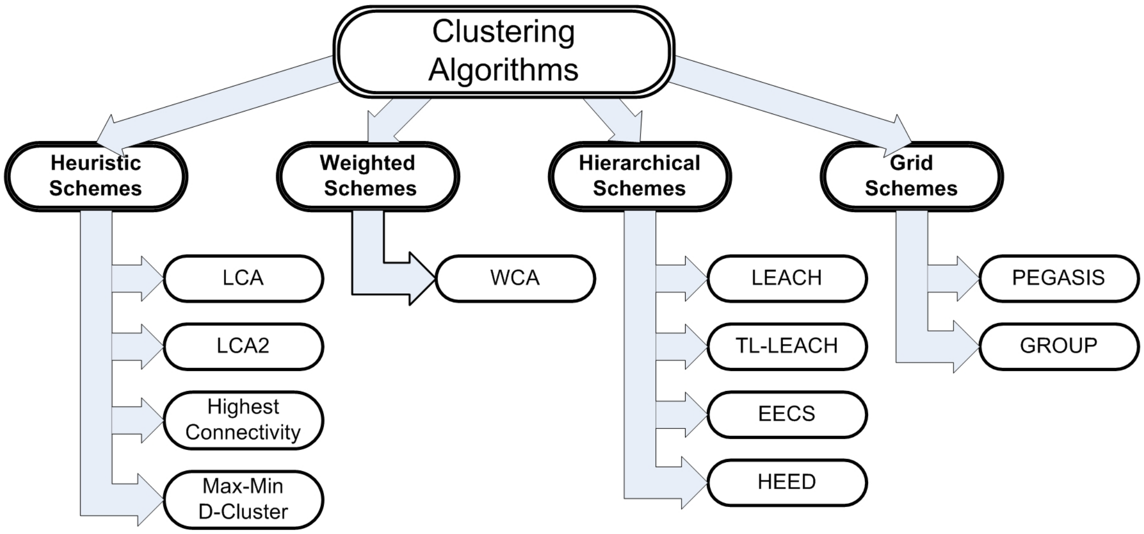

Fig. 2. Classification of Proposed Clustering Schemes

There are many types of heuristic algorithms that exist in choosing clusterheads. We each will see that these algorithms deal only with a subset of parameters which impose constraints on the system. From this point of view, each one of these algorithms are only suitable for a specific application, rather than any arbitrary wireless mobile network.

1) Linked Cluster Algorithm (LCA) [9], [10], [11]: LCA, was one of the very first clustering algorithms developed. It was initially developed for wired sensors, but later implemented in wireless sensor networks.

In LCA, each node is assigned a unique ID number and has two ways of becoming a clusterhead. The first way is if the node has the highest ID number in the set including all neighbor nodes and the node itself. The second way, assuming none of its neighbors are clusterheads, then it becomes a clusterhead.

2) Linked Cluster Algorithm 2 (LCA2) [10], [11]: LCA2 was proposed to eliminate the election of an unnecessary number of clusterheads, as in LCA. In LCA2, they introduce the concept of a node being covered and non-covered. A node is considered covered if one of its neighbors is a clusterhead. Clusterheads are elected starting with the node having the lowest ID among non-covered neighbors.

3) Highest-Connectivity Cluster Algorithm [10]: This algorithm is similar to LCA. In this scheme the number of node neighbors is broadcast to the surrounding nodes. The result is that instead of looking at the ID number, the connectivity of a node is considered. The node with the highest connectivity (connected to the most number of nodes) is elected clusterhead, but in the case of a tie, the node with the lowest ID prevails.

4) Max-Min D-Cluster Algorithm [11]: With Max-Min D-cluster, the authors[11] propose a new distributed clusterhead election procedure, where no node is more than d (d is a value selected for the heuristic) hops away from the clusterhead. This algorithm provides load balancing among clusterheads.

The clusterhead selection criteria is developed by having each node initiate $2d$ rounds of flooding, from which the results are logged. Then each node follows a simple set of rules to determine their respective clusterhead. The 1st d rounds are called *floodmax*, used to propagate the largest node ids. After this is complete, the 2nd d rounds of flooding occur. This round is called *floodmin*, used to allow the smaller node ids to reclaim some of their territory. Then each node evaluates the logged entries following the rules listed below[11]:

- Rule 1: Each node checks to see if it has received its own id in the 2nd d rounds of flooding. If it has, then it can declare itself the clusterhead and skip the other rules. Otherwise it proceeds to Rule 2.
- Rule 2: Each node looks for node pairs. Once this is complete, it selects the minimum node pair to be the clusterhead. If a node pair does not exist, they proceed to Rule 3.
- Rule 3: Elects the maximum node id in the 1st d rounds of flooding as the clusterhead for this node.

After the node has completed following the rules, it needs to determine if it is a gateway node or not [11]. This is done by sending a broadcast out to its neighbors listing its elected clusterhead. After hearing back from all neighbors, a node is able to determine whether or not it is a gateway node. A gateway node is a node where some, or most of the neighboring nodes have a different clusterhead than the node in question. Once the gateway node has been found, each node communicates with the clusterhead. This is done by sending a message inward from the fringes of the cluster (the gateway node). This message contains its node id, all neighboring gateway nodes, and their associated clusterheads.

Information is added by each node, as the message propagates towards all clusterheads.

This algorithm is valid only if the following two assumption are made:

- Assumption 1: During the flooding, no node id will propagate further than d-hops from the originating node.
- Assumption 2: All nodes that survive the *floodmax* elect themselves clusterheads.

These assumptions are proved in detail by the authors, in [11].

B. Weighted Schemes

1) Weighted Clustering Algorithm (WCA) [16]: The algorithm explained in this section is a non-periodic procedure to the clusterhead election, invoked on demand every time a reconfiguration of the networks topology is unavoidable.[16] This clustering algorithm tries to find a long-lasting architecture during the first clusterhead election. When a sensor loses the connection with any clusterhead, the election procedure is invoked to find a new clustering topology. This is an important feature in power saving, as the re-election procedure, which consumes energy, occurs less frequently. This algorithm is based on a combination of metrics that takes into account several system parameters such as: the ideal node degree; transmission power; mobility; and the remaining energy of the nodes. Depending on the specific application, any or all of these parameters can be used as a metric to elect clusterheads. Another important aspect of the algorithm is that it is fully distributed; meaning that all the nodes in the mobile network share the same responsibility acting as clusterheads.

2) Clusterhead election procedure: The election procedure is based upon a global parameter, that is called *combined weight*, which is described by[16]:

$$W_v = w_1 \Delta_v + w_2 D_v + w_3 M_v + w_4 P_v \qquad (1)$$

where w_1, w_2, w_3, w_4 are the *weighing factors* for the corresponding system parameters. The weighting factors can be chosen based upon the specific application. The *combined weight* is calculated by each node and broadcast across the network. The node with smallest W_v is chosen as the clusterhead. The first component, $w_1 \Delta_v$, helps in efficient MAC functionality, as it is always important to have a bound on the maximum number of nodes in a cluster. The second component, D_v, is the average distance from the neighbors and is strictly related to power consumption. It is known [16] that more power is required for long range transmission. The third component is due to mobility of the nodes. It is desirable that a clusterhead moves very slow, in order to have a more stable cluster architecture. From this point of view a node that moves slowly is always a better choice to be a clusterhead [16]. The last component is directly related to the available energy in a node: if a node was already a clusterhead it may have consumed a large amount of energy and should not be considered for the next clusterhead election. The weighing factors (w_1, w_2, w_3, w_4) can be chosen according to the system needs. For example, power control is very important in CDMA networks [17], thus the weight of the corresponding parameter can be increased. The flexibility of changing the weight factors helps in the application of this algorithm for different implementations.

3) Complexity due to distributiveness: The time required for the selection of the node with minimum W_v depends on the implementation of the algorithm. As it is not possible [16] to have a centralized server in ad hoc sensor networks, the algorithm proposes a distributed solution in which all nodes broadcast their ids along with W_v values. Each node receives the broadcast from its neighbors and stores the information. The stored information is again exchanged with the immediate neighbors and the process continues until all the nodes become aware of the node with the smallest W_v. The time required will depend on the diameter of the underlying network.

C. Hierarchical Schemes

1) LEACH [18]: Low-Energy Adaptive Clustering Hierarchy (or LEACH) was one of the first major improvements on conventional clustering approaches in wireless sensor networks. Conventional approaches algorithms such as MTE (Minimum-Transmission-Energy) [19] or direct-transmission do not lead to even energy dissipation throughout a network. LEACH provides a balancing of energy usage [18] by random rotation of clusterheads. The algorithm is also organized in such a manner that data-fusion can be used to reduce the amount of data transmission.

The decision of whether a node elevates to clusterhead is made dynamically at each interval. The elevation decision is made solely by each node independent of other nodes to minimize overhead in clusterhead establishment. This decision is a function of the percentage of optimal clusterheads in a network (determined a priori on application), in combination with how often and the last time a given node has been a clusterhead in the past. The threshold function is defined as [18]:

$$T(n) = \begin{cases} \frac{P}{1-P(r mod \frac{1}{P})} & \text{if n} \in \text{G} \\ 0 & \text{otherwise} \end{cases}$$

Where n is the given node, P is the a priori probability of a node being elected as a clusterhead, r is the current round number and G is the set of nodes that have not been elected as clusterheads in the last $\frac{1}{P}$ rounds. Each node during clusterhead selection will generate a random number between 0 and 1. If the number is less than the threshold (*T(n)*), the node will become a clusterhead. Following elevation to clusterhead, the new clusterhead will broadcast its status to neighboring nodes. These nodes will then determine the optimal clusterhead (in terms of minimum energy required for transmission) and relay their desire to be in that particular cluster. The broadcast messages as well as cluster establishment messages are transmitted using CSMA (Carrier Sense Multiple Access) to minimize collisions. Following cluster establishment, clusterheads will create a transmission schedule and broadcast the schedule to all nodes in their respective cluster. The schedule consists of TDMA slots for each neighboring node. This scheduling scheme allows for energy minimization [18] as nodes can turn off their radio during all but their scheduled time-slot.

2) TL-LEACH [20]: Two-Level Hierarchy LEACH (or TL-LEACH) is a proposed extension to the LEACH algorithm. It utilizes two levels of clusterheads (primary and secondary) in addition to the other simple sensing nodes. In this algorithm, the primary clusterhead in each cluster communicates with the secondaries, and the corresponding secondaries communicate with the nodes in their sub-cluster. Data-fusion can also be performed as in LEACH. In addition, communication within a cluster is still scheduled using TDMA time-slots. The organization of a round will consist of first selecting the primary and secondary clusterheads using the same mechanism as LEACH, with the a priori probability of being elevated to a primary clusterhead less than that of a secondary node. Communication of data from source node to sink is achieved in two steps [20]:

1) Secondary nodes collect data from nodes in their respective clusters. Data-fusion can be performed at this level.
2) Primary nodes collect data from their respective secondary clusters. Data-fusion can also be implemented at the primary clusterhead level.

The two-level structure of TL-LEACH reduces the amount of nodes that need to transmit to the base station, effectively reducing the total energy usage.

3) EECS [21]: An Energy Efficient Clustering Scheme (or EECS) is a clustering algorithm in which clusterhead candidates compete for the ability to elevate to clusterhead for a given round. This competition involves candidates broadcasting their residual energy to neighboring candidates. If a given node does not find a node with more residual energy, it becomes a clusterhead. Cluster formation is different than that of LEACH. LEACH forms clusters based on the minimum distance of nodes to their corresponding clusterhead [18]. EECS extends this algorithm by dynamic sizing of clusters based on cluster distance from the base station [21]. The result is an algorithm that addresses the problem that clusters at a greater range from the base station require more energy for transmission than those that are closer. Ultimately, this improves the distribution of energy throughout the network, resulting in better resource usage and extended network lifetime.

4) HEED [22]: Hybrid Energy-Efficient Distributed Clustering (or HEED) is a multi-hop clustering algorithm for wireless sensor networks, with a focus on efficient clustering by proper selection of clusterheads based on the physical distance between nodes. The main objectives of HEED are to [22]:

- Distribute energy consumption to prolong network lifetime;
- Minimize energy during the clusterhead selection phase;
- Minimize the control overhead of the network.

The most important aspect of HEED is the method of clusterhead selection. Clusterheads are determined based on two important parameters [22]:

1) The *residual energy* of each node is used to probabilistically choose the initial set of clusterheads. This parameter is commonly used in many other clustering schemes.
2) *Intra-Cluster Communication Cost* is used by nodes to determine the cluster to join. This is especially useful if a given node falls within the range of more than one clusterhead. In HEED it is important to identify what the range of a node is in terms of its power levels as a given node will have multiple discrete transmission power levels. The power level used by a node for intra-cluster announcements and during clustering is referred to as cluster power level [22]. Low cluster power levels promote an increase in spatial reuse [22] while high cluster power levels are required for inter-cluster communication as they span two or more cluster areas. Therefore, when choosing a cluster, a node will communicate with the clusterhead that yields the lowest intra-cluster communication cost. The intra-cluster communication cost is measured using the *Average Minimum Reachability Power* (AMRP) measurement [22]. The AMRP is the average of all minimum power levels required for each node within a cluster range R to communicate effectively with the clusterhead i. The AMRP of a node i then becomes a measure of the expected intra-cluster communication energy if this node is elevated to clusterhead. Utilizing AMRP as a second parameter in clusterhead selection is more efficient then a node selecting the nearest clusterhead [22].

D. Grid Schemes

1) PEGASIS [23]: Power-Efficient GAthering in Sensor Information Systems (or PEGASIS) is a data-gathering algorithm that establishes the concept that energy savings can result from nodes not directly forming clusters. The algorithm presents the idea that if nodes form a chain from source to sink, only 1 node in any given transmission time-frame will be transmitting to the base station. Data-fusion occurs at every node in the sensor network allowing for all relevant information to permeate across the network [23]. In addition, the average transmission range required by a node to relay information can be much less than in LEACH [23], resulting in an energy improvement versus the hierarchial clustering approach.

2) GROUP [24]: The Group algorithm is a grid-based clustering algorithm. In this algorithm one of the sinks (called the primary sink), dynamically, and randomly builds the cluster grid [24]. The clusterheads are arranged in a grid-like manner as in *Fig. 3*. Forwarding of data queries from the sink to source node are propagated from the Grid Seed (GS) to its clusterheads, and so on. The GS is a node within a given radius from the primary sink. In terms of clusterhead selection,

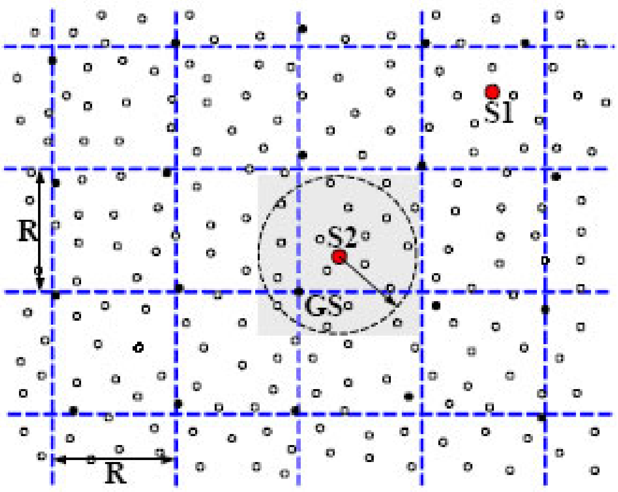

Fig. 3. GROUP Example Cluster Grid [24]

on a given round the primary sink selects a GS based on residual energy. Once the GS has been selected, the GS selects clusterheads along the corners of the grid at a range *R*. Each new clusterhead will then select more clusterheads along the grid until all clusterheads have been selected. These selections are based on the residual energy of nodes near the corners of the grid. Data transmission in GROUP is dependant on the type of data being collected. In the case of a location unaware data query (data that is not dependant on the location of the sensing node), the query is passed from the central most sink in the network to its nearest clusterhead. That clusterhead will then broadcast the message to neighboring clusterheads. If the data is location aware, then the requests are sent down the chain of clusterheads towards the specified region using unicast packets. For both data queries, data is transmitted upstream through the chain of clusterheads established during cluster formation. Energy conservation is achieved due to the lower transmission distance for upstream data. In LEACH, a clusterhead must transmit data to the base station directly [18], while in GROUP, the data is transmitted across short ranges through the upstream path [24].

IV. Performance of Proposed Algorithms

When analyzing the performance of the proposed clustering algorithms, there are two major areas that will be examined.

- *Power, Energy and Network Lifetime:* The power utilized in a sensor network is consumed as sensors are performing sensing, processing and communication tasks. Due to the limited energy nature of the sensor nodes, network lifetime is dependant on the efficient use of this energy. The primary comparison measurement when looking at the efficiency of a given algorithm is the network lifetime.
- *Quality and Reliability of the Links:* When comparing clustering algorithms, the quality of the links is an important comparison. Each clustering scheme proposes various recovery mechanisms. The performance of these recovery mechanisms has dramatic impact on the overall performance of the scheme. Also the reliability of data transport from source to sink is a critical feature in implementation.

These are key elements for clustering in WSNs, and they will be looked at further in the following sections.

With regards to heuristic algorithms, explicit comparisons cannot be made in terms of power and quality, as by definition these algorithms are not based on quantitative parameters. We can however draw some conclusions on the global qualities of the algorithms.

In LCA and LCA2 [9], [10], nodes communicate using TDMA frames, where each frame has slots for each node in the network to communicate. This means that LCA is only realistic for small networks (less than 100 nodes) [11]. The result is that larger LCA networks impose greater communication delays.

In Highest-Connectivity [10], a high turnover of clusterheads occurs when the network topology changes. Network topology information must be stored in each node. When clusterhead change occurs, that information must be transferred from old to new clusterheads. This makes clusterhead rotation undesirable due to the high overhead associated with this task.

The authors of [11], showed that the d-hop Max-Min Algorithm produces: a smaller number of clusterheads, much larger clusters, and longer clusterhead duration compared to LCA. They also showed that while the Highest-Connectivity algorithm has a slightly larger cluster size than d-hop Max-Min, it suffers from additional overhead associated with more frequent topology changes. For dense networks, a node in d-hop Max-Min exists as clusterhead approximately 100% longer than LCA2 and this duration continues to increase proportional to the network density [11]. These features show that d-hop Max-Min provides the best overall clusterhead election characteristics compared to all other heuristic algorithms.

A. Power, Energy and Network Lifetime

1) WCA: It evaluates a weight for each node and the clusterheads are chosen among the best suitable nodes in terms of node degree, distance from neighbors, mobility and energy available [16]. In terms of energy consumption, the algorithm tries to achieve the most stable cluster architecture, meaning after the first iteration the algorithm is executed only when there is a demand. This reduces system updates and hence computation and communication costs.The algorithm performs better than existing heuristic schemes [25], [26], [27] in terms of reaffiliation in time, as shown in *Fig.4*: Another important feature of this scheme is that the clusterheads are chosen among the nodes that have enough energy available. This leads to a fair clusterhead distribution amongst nodes; avoiding the problem of power drainage for nodes that serve as clusterheads for long periods of time.

2) LEACH: It provides the following key areas of energy savings:

- No overhead is wasted making the decision of which node becomes clusterhead as each node decides independent of

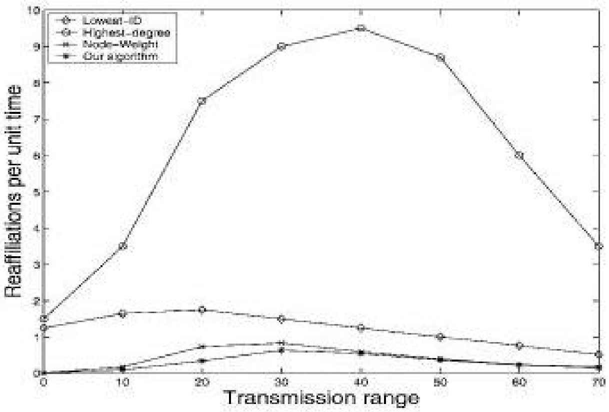

Fig. 4. Comparison of Reaffiliations for Heuristic Algorithms [16]

other nodes [18].

- CDMA allows clusters to operate independently, as each cluster is assigned a different code.
- Each node calculates the minimum transmission energy to communicate with its clusterhead and only transmits with that power level.

LEACH provides the following improvements over conventional networks [18]:

- LEACH reduces transmission energy by a factor of 8 versus MTE and direct-transmission.
- The first death occurs in LEACH 8 times later than that of MTE, direct-transmission and static clustering. In addition the final death of a node occurs more than 3 times later than that of the other listed protocols.

3) TL-LEACH: It improves upon LEACH by utilizing a two-level hierarchy [20]. The energy improvements are achieved from smaller transmission distance for the majority of nodes [20]. This network configuration requires that merely a few nodes transmit large distances.

Simulations have shown that the addition of the two-level hierarchial algorithm TL-LEACH results in an improvement of network lifetime by approximately 30% versus its basis algorithm LEACH [20].

4) PEGASIS: The minimization of energy in this algorithm is achieved from four areas [23]:

- During a given round, only 1 node in the network is transmitting data to the base station. Since the transmission range to the base station is large, this can result in an improvement with regards to energy savings.
- Since each node communicates with its nearest neighbor, the energy utilized by each node is also minimized.
- Each node performs data-fusion, effectively distributing the energy required for this task across the network.
- The overhead associated with dynamic cluster formation during each round is eliminated.

Simulations in C have shown that PEGASIS can result in a 100% to 300% improvement over LEACH for a variety of different network sizes and configurations [23].

5) EECS: Minimization of energy consumption in EECS is accomplished in a similar manner to that of LEACH [21], however the algorithm attempts to improve on LEACH. This is accomplished by creating dynamic cluster sizes which are a function of the distance from the base station to the cluster. This addresses the larger transmission power requirements for nodes at a greater range from the base station.

The energy utilization rate η is used as a comparison measurement for energy in the simulations of this algorithm [21]. It is the ratio of the total energy consumed in the network at the time the first node dies, to the total initial energy. This measurement is related to the efficient spread of energy in the network. η in EECS was found to be approximately 93% while LEACH had only η of 53% [21].

The EECS protocol has shown a 35% improvement in network lifetime versus the original LEACH in a simulation environment [21].

6) GROUP: Energy conservation is achieved by the lower transmission distance for upstream data. In LEACH, a clusterhead must transmit data to the base station directly, while in GROUP, data is transmitted short ranges along the upstream path.

Simulations show that energy consumption is related to the number of nodes in the network. With 75 nodes, the energy consumption is comparable to that of LEACH, however for a larger number of nodes, GROUP can offer a reduction in the maximum energy consumption per node by a factor of 4 [24]. It can also be seen that the energy distribution for a larger number of nodes is more consistent with GROUP than it is with LEACH *Fig. 5*.

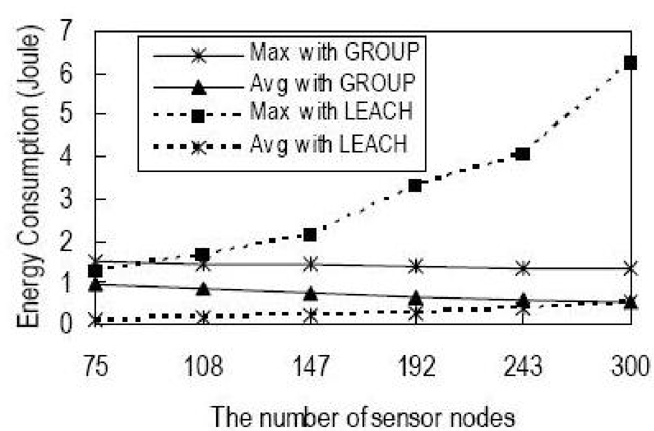

Fig. 5. Energy consumption for GROUP vs. LEACH [24]

Energy distribution is closely tied to network lifetime. Since the death of the first node will happen substantially later than that of LEACH [21] (this can be determined from the difference between maximum and average energy consumed,

TABLE I

NETWORK LIFETIME COMPARED TO STANDARD LEACH

Algorithm	Lifetime Characteristics
LEACH[18]	First node death occurs 8 time later than conventional methods, increasing network lifetime
TL-LEACH[20]	Results in 30% improvement in network lifetime versus standard LEACH
EECS[21]	35% improvement in network lifetime versus standard LEACH
PEGASIS[23]	Simulations have shown that PEGASIS can result in a 100% to 300% increase in network lifetime versus standard LEACH
GROUP[24]	First node death occurs substantially later than LEACH, see Fig.5

shown in *Fig. 5*), one can assume that there will be increased network lifetime, although no such measurements had been examined at the time the protocol was proposed.

7) HEED: In this algorithm, network life time is prolonged through:

- Reducing the number of nodes that compete for channel access;
- Clusterhead updates, regarding cluster topology; and
- Routing through an overlay among clusterheads, which has a small network diameter.

Comparing HEED to a generic weight-based clustering (GC) protocol such as WCA [22]:

- When using a GC algorithm, the number of iterations grows quickly as the cluster radius increases, so each node has more neighbors. Implying a node has to wait longer for higher weighted nodes to decide which cluster to join. Therefore, we have more energy consumption [22]. Clustering in GC takes 85 iterations for a cluster radius of 400. Whereas, HEED takes only 6 iterations for all cluster ranges [22]. This means less energy consumption.
- In GC, it is guaranteed that the node with the highest residual energy will be the clusterhead, whereas in HEED, clusterheads are chosen based on their residual energy and their intra-cluster communication cost.

It is easiest to compare HEED to generalized LEACH. Generalized LEACH is a protocol with additional features added to LEACH (as described above), they are [22]:

1) The routing protocol is assumed to propagate the node residual energy throughout the network; and
2) A node selects a clusterhead in its range proximity, which is not assumed to span the entire network area.

The first feature will allow for better selection of clusterheads than original LEACH and it also prolongs the network lifetime [22]. The second feature allows for multi-hop networks.

HEED improves network lifetime over generalized LEACH, because generalized LEACH randomly selects clusterheads, thus resulting in a faster death of some nodes [22]. HEED avoids this by well distributing clusterheads across the network.

Energy consumed in clustering is measured as a fraction of the total dissipated energy in the network. HEED uses less energy in clustering than generalized LEACH because it does not propagate residual energy information [22].

When studying the effect of the distance between the sink and the clusters with respect to the network lifetime, we find that HEED prolongs network lifetime compared to generalized LEACH [22].

B. Quality and Reliability of Links

1) WCA: In terms of quality and reliability, the WCA algorithm has the flexibility to be adapted to many applications, assigning different weights to the parameters of the *combined weight* [16]. For example, according to the specific application, the effect of the node degree, distances, mobility and energy could be weighted differently. This flexibility allows the algorithm to be scalable to different applications. Load balancing amongst clusterheads is guaranteed by the constraint on the maximum number of nodes in a cluster. From this point of view, this algorithm achieves a better load balancing than the other heuristic schemes proposed in the literature [25], [26], [27].

2) LEACH & TL-LEACH: When examining the reliability of both the LEACH and TL-LEACH protocols, we can observe the several key features that have been built into the protocol to improve the reliability of transmission [18], [20]:

- The CSMA mechanism is used to avoid collisions.
- CDMA is utilized between clusters to eliminate the interference from neighboring clusters.
- Periodic rotation of clusterheads extend the network lifetime, guaranteeing full connectivity in the network for longer periods than conventional algorithms.

The TL-LEACH extension of a two-level hierarchy offers no direct reliability improvements over standard LEACH.

3) PEGASIS: It offers promising improvements with relation to network lifetime, however reliability may not be as promising. In PEGASIS, each node communicates with its nearest neighbor. This implementation may be more susceptible to failure due to gaps in the network.

4) EECS: It extends on the capability of LEACH by utilizing dynamic cluster sizing. In terms of recovery mechanisms, EECS offers similar reliability as that of LEACH. However, since EECS offers improved energy utilization throughout the network [21], full connectivity can be achieved for a longer duration. This results in reliable sensing capabilities at the range extremes of a network for a longer period of time.

5) GROUP: It utilizes its own recovery mechanisms in the case of a failed node. When a node fails in its attempt to communicate with its clusterhead it will send a broadcast message to search and establish a new clusterhead. The neighbors in the failed cluster will then elect a new clusterhead in a similar fashion to that of grid construction [24]. When a node is newly elevated to clusterhead, it will determine its corresponding up and downstream clusterheads via a broadcast recovery packet [24]. This mechanism ensures that data will be transmitted successfully from source to sink.

6) HEED: This algorithm produces balanced clusters compared to GC, where it has a higher percentage of non-single node clusters than GC [22]. HEED also reduces the likelihood that clusterheads are neighbors within the cluster range [22]. This is because HEED uses intra-cluster communication cost in selecting its clusterheads. Therefore the node distribution does not impact the quality of communication.

V. Conclusion and Future Work

In this paper we have examined the current state of proposed clustering protocols, specifically with respect to their power and reliability requirements. In wireless sensor networks, the energy limitations of nodes play a crucial role in designing any protocol for implementation [1]. In addition, Quality of Service metrics such as delay, data loss tolerance, and network lifetime expose reliability issues when designing recovery mechanisms for clustering schemes. These important characteristics are often opposed, as one often has a negative impact on the other.

Protocols presented in this paper offer a promising improvement over conventional clustering; however there is still much work to be done. Many energy improvements thus far have focused with minimization of energy associated in the clusterhead selection process [18], [20] or with generating a desirable distribution of clusterheads [24], [22]. Optimal clustering in terms of energy efficiency should eliminate all overhead associated not only with the clusterhead selection process, but also with node association to their respective clusterheads. Sensor network reliability is currently addressed in various algorithms by utilizing re-clustering that occurs at various time intervals; however the result is often energy inefficient and limits the time available within a network for data transmission and sensing tasks. Further improvements in reliability should examine possible modifications to the re-clustering mechanisms following the initial clusterhead selection. These modifications should be able to adapt the network clusters to maintain network connectivity while reducing the wasteful resources associated with periodic re-clustering. In addition, other mechanisms such as the ability of nodes to maintain membership in auxiliary clusters can reinforce the current state of sensor network reliability.

VI. Acknowledgements

The authors would like to acknowledge Dr. P. Kumarawadu for his guidance and valuable suggestions in the area of Wireless Sensor Networks.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Netowrks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug 2002.

[2] S. Meyer and A. Rakotonirainy, "A Survey of Research on Context-Aware Homes," *Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing, Adelaide Australia*, 2003.

[3] B. Warneke, M. Last, B. Liebowitz, Kristofer, and S. Pister, "Smart Dust: Communicating with a Cubic-Millimeter Computer," *Computer Magazine*, vol. 34, no. 1, pp. 44–51, Jan 2001.

[4] J. M. Kahn, R. H. Katz, and K. Pister, "Next Century Challenges Mobile Networking for Smart Dust," *5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Aug 1999.

[5] V. Hsu, M. Kahn, and K. S. J. Pister, "Wireless Communication for Smart Dust," *Electronic Research Laboratory Technical Memorandum*, Feb 1998.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Micro Sensor Networks," *Proceedings of IEEE HICSS*, Jan 2000.

[7] C. F. Chiasserini, I. Chlamtac, P. Monti, and A. Nucci, "Energy Efficient Design of Wireless Ad Hoc Networks," *Proceedings of European Wireless*, Feb 2002.

[8] S. Bandyopadhyay and E. J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *IEEE INFOCOM*, April 2003.

[9] D. J. Baker and A. Epheremides, "The Architectural Organization of a Moblie Radio Network via a Distributed Algorithm," *IEEE Transactions on Communications*, vol. Com-29, no. 11, November 1981.

[10] P. Tsigas, "Project on Moblie Ad Hoc Networking and Clustering for the Course EDA390 Computer Communcation and Distributed Systems," Manual for University Course.

[11] A. Amis, R. Prakash, T. Vuong, and D. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," *IEEE INFOCOM*, March 2000.

[12] C.E.Nishimura and D.M.Conlon, "IUSS dual use: Monitoring of whales and earthquakes using SOSUS," *Mar. Technol. Soc. J.*, vol. 27, no. 4, 1994.

[13] A. Mainwaring *et al.*, "Wireless Sensor Networks for Habitat Monitoring," *Proceedings of the 1st ACM International Workshop on WSN*, 2002.

[14] C.Y.Chong, S.Mori, and K.C.Chang, "Distributed multitarget multisensor tracking," *in Multitarget Multisensor Tracking:Advanced Applications*, 1990.

[15] C. Intanagonwiwat *et al.*, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transaction on Networking*, vol. 11, no. 1, Feb. 2003.

[16] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Clustering Computing*, vol. 5, pp. 193–204, 2002.

[17] W. C. Y. Lee, "Overview of Cellular CDMA," *IEEE Trans. on Vehicular Technology*, vol. 40, no. 2, pp. 291–302, May 1991.

[18] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Sensor Networks," *Proceedings of the 33th Hawaii International Conference on System Sciences*, 2000.

[19] T. Meng and R. Volkan, "Distributed Network Protocols for Wireless Communication," *Proc. IEEE ISCAS*, May 1998.

[20] V. Loscri, G. Morabito, and S. Marano, *A Two-Level Hierarchy for Low-Energy Adaptive Clustering Hierarchy*, DEIS Department, University of Calabria.

[21] M. Ye, C. Li, G. Chen, and J. Wu, *EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks*, National Laboratory of Novel Softaware Technology,Nanjing University,China.

[22] O. Younis and S. Fahmy, "HEED: A Hybrid Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, Oct-Dec 2004.

[23] S. Lindsey and C. S. Raghavendra, *PEGASIS: Power-Efficient Gathering in Sensor Information Networks*, Computer Systems Research Department, the Aerospace Corporation.

[24] L. Yu, N. Wang, W. Zhang, and C. Zheng, *GROUP: A Grid-Clustering Routing Protocol for Wireless Sensor Networks*, East China Normal University.

[25] A. K. Parekh, "Selecting Routers in Ad Hoc Wireless Networks," *Proceedings of the SBT/IEEE International Symposium*, August 1994.

[26] D. J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Transactions on Communications*, 1981.

[27] S. Basagni, "Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks," *Proceedings of Vehicular Technology Conference, VTC*, vol. 2, 1999-Fall.