2003

Self-Configuring Clusters, Data Aggregation and Authentication in Microsensor Networks

Maleq Khan

Bharat Bhargava
Purdue University, bb@cs.purdue.edu

Leszek Lilien

Report Number:
03-005

SELF-CONFIGURING CLUSTERS, DATA AGGREGATION AND AUTHENTICATION IN MICROSENSOR NETWORKS

Maleq Khan
Bharat Bhargava
Leszek Lilien

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD TR #03-005
March 2003
(Revised August 2003)

Self-configuring Clusters, Data Aggregation, and Authentication in Microsensor Networks*

Maleq Khan, Bharat Bhargava, and Leszek Lilien

Department of Computer Sciences and
Center for Education and Research in Information Assurance and Security (CERIAS)
Purdue University, West Lafayette, IN 47907
{mmkhan, bb, llilien}@cs.purdue.edu

Abstract

Microsensors operate under severe energy constraints and should be deployed in large numbers without any pre-configuration. The main contribution of this paper is a generalized self-clustering protocol, called Low-energy Localized Clustering (LLC). It incorporates the best features of two other recently proposed self-configuring protocols for sensor networks: the Localized protocol and the Low Energy Adaptive Clustering Hierarchy (LEACH) protocol. LLC covers a range of behaviors from the better-clustering performance of the Localized method to the more energy-efficient operation of the LEACH method. As experimental results show, the main advantage of LLC is that it can be energy-efficient while maintaining a high cluster quality. We outline data aggregation approaches such as summarization, finding representative data items, and pattern matching. Data aggregation is a necessity in microsensor networks to avoid transmitting huge volumes of raw data, which is energy-intensive. Finally, an energy-efficient Randomized Data Authentication protocol is designed specifically for microsensor applications.

Keywords Sensor Networks, Microsensors, Self-configuring Clusters, Data Aggregation, Security in Microsensor Networks.

* This research was supported by CERIAS, and NSF Grants CCR-0001788 and EIA-0103676.

1 Introduction

Advances in integrated circuit technology have enabled mass production of tiny, cost-effective, and energy-efficient wireless sensor devices with on-board processing capabilities. The emergence of mobile and pervasive computing has created new applications for them. Sensor-based applications span a wide range of areas, including remote monitoring of seismic activities and environmental factors (e.g., air, water, soil, wind, chemicals), condition-based maintenance, smart spaces, military surveillance, precision agriculture, transportation, factory instrumentation, and inventory tracking [2, 9].

A *microsensor* is a device which is equipped with a sensor module (e.g., an acoustic, a seismic, or an image sensor) capable of sensing some entity in the environment, a digital unit for processing the signals from the sensors and performing network protocol functions, a radio module for communication, and a battery to provide energy for its operation [9]. Microsensors typically have low processing power and slow communication ability. For example, Berkeley mote [1] has a 8-bit Atmel AT90LS8535 microcontroller running at 4 MHz. A low-power radio transceiver MICA2, designed for sensor networks, operates at 916 MHz and provides a data transmission rate of 19.2 Kbps [4]. The size of a MICA2 MPR400CB is 2.25″×1.25″×0.25″. These parameters ensure limited weight, size, and cost. We use the term *sensor* to refer to a microsensor.

When deployed in large numbers and embedded deeply within large-scale physical systems, sensors are able to measure aspects of

the physical environment in unprecedented detail [2]. Networking these sensors with the ability to coordinate amongst themselves in a large sensing task revolutionizes information gathering and processing. Large scale, dynamically-changing, and robust sensor colonies can be deployed in inhospitable physical environments such as remote geographic regions or toxic urban locations. They will also enable low-maintenance sensing in more benign but less accessible environments such as large industrial plants, enemy terrain, aircraft interiors, etc. [6].

In this paper, we use a cluster-based hierarchical architecture for sensor networks to achieve and support scalability. The architecture with description of its components and their functionalities is given in Section 2.

A large number of sensor nodes deployed for an application precludes manual configuration, and the environmental dynamics preclude design-time pre-configuration [3]. Nodes will have to self-configure to establish a topology that enables communication and sensing coverage under stringent energy constraints. Two existing self-configuring clustering protocols: the Localized protocol [6] and the Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol [8] are analyzed in Section 3. While the Localized protocol forms good quality clusters, LEACH focuses on using lower energy consumption in forming clusters. Incorporating the best features of these two protocols, we built a generalized scheme called Low-energy Localized Clustering (LLC).

Data aggregation is a good paradigm for wireless routing in sensor networks [7, 10]. The idea is to combine the data coming from different sources and routes. This eliminates redundancy, minimizes the number of transmissions and thus saves energy [12]. Beamforming [14, 18] and functional decomposition [9] are two ways of aggregating sensor data. Their limitations are identified and a few other data aggregation methods are outlined in Section 4.

Data authentication mechanism in sensor networks is presented in Section 5. We propose an energy-efficient Randomized Data Authentication protocol.

Finally the experimental results of the LLC clustering protocols are presented in section 6.

2 Cluster-based Architecture for a Sensor Network

Sensor networks are large-scale data-intensive systems that manage parallel and real-time communications in dynamic environments. To support scalability we use a cluster-based hierarchical structure (Fig. 1). As the number of sensors is increased, more clusters can be formed without increasing the processing or communication loads on individual cluster heads.

The three levels in the hierarchical design of this architecture consist of a base station (a data sink) at the top level, cluster heads at the middle level, and the other sensors at the leaf level.

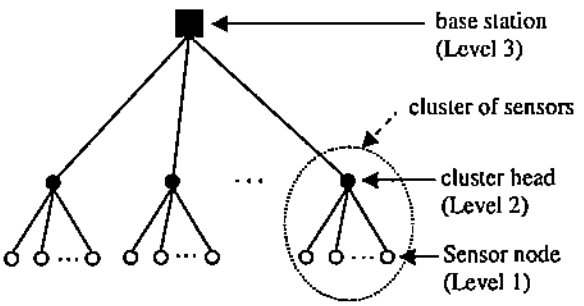

Fig. 1 A cluster-based hierarchical architecture for sensor networks.

The *base station* is a machine capable of analyzing data collected from the cluster heads and displaying a global view of events being monitored. It is responsible for initiating and managing the network and is ultimately the gateway of the sensor network to the Internet or another network.

Sensors are deployed in large numbers across an area of observation. Their primary function is to collect data from their surroundings. A direct communication among the level-1 sensors occurs only at the time of cluster formation or cluster reconfiguration. Otherwise, the main stream of communication consists of conveying data results to the corresponding cluster head.

Before deployment, each sensor is given an *id* that uniquely identifies it. Similarly a *security code*, which could be implemented as a hardware-embedded signature on the microsensor chip, is assigned to each sensor. It is used to authenticate data sent by the node.

Cluster heads are selected from among the deployed sensors by a self-configuring mechanism. Sensors in a particular cluster register themselves with their respective cluster head. The cluster heads become immediate points of contact for their sensors for communication and reporting purposes. The heads collect data from the sensors, aggregate them, and send the results to the base station.

3 Self–configuring Sensor Clusters

3.1 The Need for Clustering

As explained in the introduction, sensors must be able to self-configure. Clustering allows sensors to efficiently coordinate their local interactions in order to achieve global goals. Localized clustering can contribute to a more scalable behavior. As the number of nodes increases, it leads to improved robustness and more efficient resource utilization for many distributed sensor coordination tasks [6].

Localization saves transmission energy since it allows for communicating with a closer local coordinator instead of a more distant base station [8]. To transmit a signal over a distance d, the required radiation energy E is proportional to d^m where m is 2 in the free space and ranges up to 4 in environments with multiple-path interferences or local noise [5].

Another advantage of using clusters is data aggregation at cluster heads, in which data collected from sensors is aggregated before forwarding to the base station, and thus the amount of energy required to transmit huge volumes of data is reduced.

3.2 Analysis of Two Protocols for Sensor Clustering

We introduce a self-configuring protocol, which is a generalization of the Localized and LEACH protocols. Before presenting the new protocol, some details of these two need to be explained.

Localized Protocol The *Localized protocol* for forming self-configuring clusters of sensor nodes is presented in [6]. All sensors start by sending advertisements to sensors within a pre-specified radius. Sensors wait after setting their *wait timer* to values proportional to their advertising radius. This allows advertisements from various sensors to reach each other.

At the end of the wait period, sensors start a *promotion timer* which is set to be inversely proportional to the sensor's remaining energy and the number of other sensors from whom the advertisement were received. That is, the sensors in the dense regions and with higher energy have smaller timeout values.

When a sensor's promotion timer expires, it promotes itself to Level 2 (a cluster head), and advertises itself as a cluster head by broadcasting the list of its potential child sensors. The list consists of the sensors whose advertisements it previously received. If a sensor appears in the lists of potential children of several cluster heads, it chooses the closest one as its cluster head. Now it cancels its own promotion timer (if it is still running), and thus drops out of the election process.

LEACH Protocol The *Low Energy Adaptive Clustering Hierarchy (LEACH) protocol* for self configuring clusters is proposed in [8]. Periodically, every sensor elects itself a cluster head with a certain probability. This probability for node n in round r is defined as:

$$T(n)=\begin{cases}\dfrac{P}{1-P\left(r \bmod \dfrac{1}{P}\right)} & \text{if } n \in G,\\ 0 & \text{otherwise}\end{cases}$$

where P is the predefined percentage of sensors that should become cluster heads, and G is the set of nodes that have not been cluster heads in the last $\frac{1}{P}$ rounds. This guarantees that in every $\frac{1}{P}$ rounds each node is elected a cluster head once. Thus, the energy-intensive tasks of cluster heads are evenly distributed among the sensor nodes. The elected cluster heads broadcast an advertisement message to the rest of the nodes. A sensor selects its cluster head based on the strength of the received advertisement signal.

Analysis The main disadvantage of the Localized protocol is that every node needs to broadcast messages and manage wait and promotion timers in each round of a cluster head

election process, which requires a significant amount of energy.

The LEACH protocol is energy-efficient but the expected number of clusters is predefined. The optimal number of cluster heads depends on the network topology and transmission power related parameters. Its authors presented experimental results showing that the optimal number of cluster heads to minimize energy dissipation in data communications is approximately 5% for their setting. We use a similar simulation setting and observe that optimal number is 5% for our setting as well.

In a sensor network, usually, nodes are distributed randomly and thus topology of the network cannot be determined apriori. Unfortunately, when the sensors are highly dispersed, this predefined number of randomly selected cluster heads might not be sufficient to cover the whole area of sensor deployment. Since the LEACH protocol selects cluster heads randomly, in some instances all selected cluster heads could group in one end of the region. The sensors at the other end might not hear any cluster heads, and hence remain isolated from any cluster. Even if a full coverage can be accomplished, the area covered by a cluster could increase to a point where long range communications and thus higher energy are required.

In the Localized protocol, there can be no isolated groups of sensors. Every sensor has a promotion timer, which expires at some time and, if it does not hear from any other cluster head, it promotes itself to a cluster head.

The number of cluster heads is not predefined in Localized protocol. By selecting a cluster head from a dense region and keeping the cluster heads well separated, the Localized method reduces intra-cluster communications distances (for the data transmission phase) resulting in good quality clusters. Cluster quality is formally defined below in the following subsection.

The conclusion is that LEACH saves more energy in the self-configuration process while the Localized method saves more energy in the data transmission phase.

3.3 Cluster Quality

Quality of clusters refers to the compactness of the clusters. It is usually measured by the total variance, which is the sum of the squared distances of the nodes to their cluster heads.

Our purpose of forming clusters is to transmit data sensed by the nodes to the base station. Therefore, we quantify the cluster quality as the inverse of the energy consumption for transmitting 1-bit of data by all nodes to the base station via cluster heads. More precisely, it is the inverse of the normalized sum of the cubic distances (considering r^3, $m = 3$, radio path loss) between the nodes and the cluster heads, and between the cluster heads and the base station. Isolated nodes are penalized by making it to send data directly to the base station. Hence, existence of an isolated node reduces the quality of the clusters. A good quality clustering saves energy later in the data transmission phase.

3.4 The Low-energy Localized Clustering (LLC) Protocol

We propose a new protocol called Low-energy Localized Clustering (LLC) that incorporates the best features of the protocols discussed above. It reduces the required energy in an election process (to improve upon the Localized protocol), reduces the chance of having isolated sensors, keeps the number of cluster heads variable, and produces good quality clusters (to improve upon LEACH).

The protocol consists of two phases: (a) a specified percentage of the nodes, called the *candidate ratio*, are randomly selected to be candidates for being cluster heads; (b) only the selected candidates compete to become cluster heads. Details of these two phases are given below.

Candidate Selection Every node selects itself as a candidate for a cluster head with a probability proportional to its remaining energy. Thus a sensor with higher energy has a greater chance to become a candidate. If the desired candidate ratio is x, the probability that node i becomes a candidate is given by

$$p_i = \frac{e_i}{e_{ir}} nx,$$

where e_i is the initial energy of sensor node i, $e_{ir} = \sum_{i=1}^{n} e_i$ is the total remaining energy in the system, and n is the number of sensor nodes.

To avoid inter-node communications for calculating total remaining energy, it is individually estimated by each node. Let t be the estimated lifetime of the system, which is estimated before deploying the sensors, and t_p be time that passed since the deployment of the sensors. The estimated total energy remaining in the system becomes:

$$e_{tr} = \frac{ne(t - t_p)}{t},$$

where e is the initial energy of a sensor node (considered to be the same for each node). Hence,

$$p_i = e_i nx \times \frac{t}{ne(t - t_p)} = \frac{e_i xt}{e(t - t_p)}.$$

Cluster Head Election Cluster heads are elected from the pool of candidates almost following the Localized protocol. There are two exceptions. First, only the candidate sensors compete while the remaining sensors sleep, and thus conserve energy, until the election process is completed. Second, after a promotion, a node declares itself a cluster head but does not publish any potential children list. The other sensors (both former candidates and non-candidates) select their cluster heads based on the strength of the signal of these declaration messages.

Analysis LLC overcomes the shortcomings of the two analyzed protocols. Suppose that 20% of sensors are selected as candidates. Then, 80% of the sensors do not participate in the election process thus saving 80% of energy that is used up to broadcast advertisements in the election process in the Localized protocol.

In LLC there is a slight chance of having an isolated group of sensors. If LEACH optimally selects 5% of the nodes as cluster heads, then in LLC the probability of selecting a candidate from a group isolated in LEACH increases fourfold (for the candidate ratio set to 20%). If any sensor in the group that would become isolated in LEACH is selected as a candidate, the group will have a cluster head in LLC. The cluster radius is bound as in the Localized protocol.

LLC is an adaptive generalization of the Localization and the LEACH protocols, with the candidate ratio being the control parameter. When the ratio is 100%, LLC behaves similar to the Localized method, since all sensors are competing to become cluster heads. When the ratio is very low (such as 5%, which is considered optimal for LEACH), the protocol operates nearly identically to the LEACH method with the same number of cluster heads. The reason is that in LLC with a low number of candidates almost all of them become cluster heads.

The main advantage of the LLC protocol is that not all sensor nodes need be involved in the election process to produce good quality clusters (nearly as good as produced by the Localized method). The experimental results show that the 40% candidate ratio is sufficient to produce such good quality clusters. The results are presented in details in Section 6.

4 Sensor Data Aggregation

4.1 Motivation and Related Methods

Data aggregation is a paradigm for wireless routing in sensor networks [7, 10]. The idea is to combine data coming from different sources and routes. This eliminates redundancy, minimizes the number of transmissions, and saves energy [12]. Automatic methods of combining or aggregating data into a small set of meaningful information are required [9].

Sensor data is different from data associated with traditional wireless networks since it is not data itself that is important. Instead, it is the analysis of data, which allows an end-user to determine something about the monitored environment, that is the important result derived from a sensor network [9]. For example, if sensors are monitoring temperature, the measurements from all sensors in a cluster need not be transmitted. Temperatures at different points of a certain area are highly correlated and the end users are only interested in a high-level description of the events occurring. The type of a high-level description of data or data aggregation that needs to be performed depends on the monitored events and user requirements. In this example, only the minimum, maximum, or the average of the temperatures might be needed.

One method of data aggregation, called *beamforming* [14, 18], combines signals from multiple sensors by calculating the weighted sum of the signals as follows:

$$y[n] = \sum_{i=1}^{N} \sum_{l=1}^{L} w_i[l] s_i[n-l]$$

where $s_i[n]$ is the signal from the i^{th} sensor, $w_i[n]$ is the weighting filter for the signal from the i^{th} sensor, N is the number of sensors, and L is the number of taps in the filter. Although beamforming has a good property that the weighting filters can be chosen to satisfy an optimization criteria, such as minimizing mean squared error or maximizing signal to noise ratio, the weighted sum of signals may not be useful for some applications.

A *functional decomposition* can sometimes be used to perform local data processing on a subset of data [9]. The base station receives all data X and processes it to find $f(X)$. The function f can sometimes be broken up into several smaller functions f_1, f_2, f_3, ..., f_n that operate on subsets of data $X_1, X_2, X_3, \ldots, X_n$ such that

$$f(X) \approx g(f_1(X_1), f_2(X_2), f_3(X_3), \ldots, f_n(X_n)).$$

Even though many data aggregation functions can not be decomposed in such a manner, we can find some applications where special data aggregation schemes can be applied for local data processing in order to reduce the communication.

We outline some approaches to data aggregation below.

4.2 Data Summarization

For some data types and applications, only the summarized information is needed to serve the purpose of monitoring environmental events. Different summarizations are suitable for different applications. They include averages, sums, minimums, maximums, medians, modes, standard deviations, quartiles, percentiles, and histograms. In addition, one can use the number of nodes detected to have crossed a threshold and the total number of active nodes associated with the cluster head. For some applications, instead of using a single summarized value, a combination of the above values can be employed.

4.3 Finding Representative Data Items

In this scheme of data aggregation, using *k-means clustering method* [17], we calculate a predefined number (k) of representatives of data. k-means clustering method received a considerable attention and used in many applications in different fields such as data mining, image processing, and bioinformatics.

Let n_c be the number of active sensors associated with a particular cluster head, and k be the desired number of representatives. The k-means data clustering produces a good minimization of the sum-of-the-squared-error, or the total variance, function. The method randomly selects k initial cluster centers. Next k clusters are formed by associating each data point with its closest cluster center. The centroids, or means, of these k clusters become the new cluster centers. The above procedure is repeated until there is no change in the cluster memberships.

Each cluster center is representative of data items in its cluster. The cluster heads send these k representatives to the base station. (Note that cluster centers represent data, while cluster heads represent sensors.) Each representative is accompanied by the number of data items associated with it, which is used to indicate the weight of the representative.

The k-means algorithm converges very fast when the dimension of data is small. For example, for a temperature sensor network the dimension is one, whereas for a sensor network measuring both temperature and humidity this dimension is two. Usually the dimensionality of the sensor data is small. This fact explains why the k-means is a suitable clustering method for sensor data.

4.4 Pattern Matching

In this scheme, sensors find patterns of data measured over a predefined time interval and send only these patterns to their cluster heads, instead of sending raw data. Thus, data aggregations are performed both at the leaf level and the cluster head level. Cluster heads collect individual patterns from their sensors and search for higher-level critical patterns that describe some critical events. Only these critical patterns are sent to the base station.

For example, consider a simple application with sensors deployed to predict thunder storms in a certain area. Each sensor collects temperature and pressure data. Periodically, a sensor finds the pattern of changing temperature and pressure that

fits best the collected data. Six example patterns for pressure changes are depicted in Fig. 2.

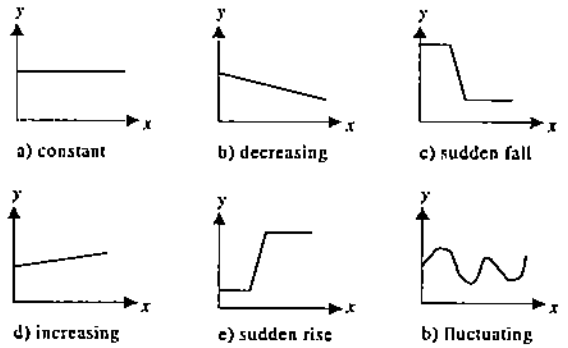

Fig. 2 Possible pressure patterns. Time is plotted on the x-axis and pressure on the y-axis.

Information sent periodically by a sensor to its cluster head is concise, which is simply a code denoting the pattern. The sudden fall, sudden rise, and fluctuation in pattern pressure are categorized as critical patterns that forecast a storm. Similar patterns are defined for temperature. Each cluster head selects those patterns and sends them to the base station. The base station analyzes the critical patterns and prepares forecasts.

4.5 Tradeoffs between Data Aggregation and Communication

The main purpose of data aggregation is to reduce required communication at various levels, and in turn to reduce the total energy consumption. Data aggregation saves energy if energy required to perform aggregation is lower than energy required to send raw data to the upper level. Different data aggregation techniques require different amounts of energy to process raw data. The choice of data aggregation method depends not only on the application requirements but also on the relative energy savings obtained by using this method.

Another tradeoff between data aggregation and communication involves time required to perform data aggregation at cluster heads or leaf-level sensors, and time required to transmit raw data. There is a delay associated with processing data at sensor nodes due to their limited processing power. Depending on an application, partial processing of data can be done at the sensor nodes to maximize data processing throughput, while still satisfying the real-time system limitations.

5 Data Authentication Protocol

The requirements for security in a sensor network, as stated in [15], include: data confidentiality, data authentication, data integrity, and data freshness. We propose a data authentication protocol, called Randomized Data Authentication (RDA) Protocol, which satisfies the energy constraints of sensor networks.

One of the main security hazards in sensor networks is the presence of foreign sensor nodes (which send false data to a cluster head or pretend to be cluster heads) in the deployment area. Authentication is the mechanism by which the receiver of a message can ascertain its origin [16].

Most of the existing authentication protocols require a trusted third party that generates secret keys for the communicating parties. Using a third party is not suitable for authentication of sensor nodes, deployed on a temporary basis and frequently reconfigured. Moreover, no IP address, required to communicate with a third party, is associated with a sensor node. Routing in sensor networks is *data-centric* [11] in contrast to the traditional IP-based end-to-end *address-centric* routing.

The RDA Protocol The proposed *Randomized Data Authentication protocol* randomly selects data items for authentication with a probability p instead of authenticating all data items.

Energy is saved by not authenticating each data item. Also the risk of compromising security is reduced, since less frequent random authentication gives attackers fewer opportunities to capture a security code. On the other hand, more intrusions may remain undetected. However, in most cases a few intrusions can be tolerated. Since data is being gathered from a large number of sensors, a relatively few malicious data items do not affect the overall results significantly. It should be noted that, in this protocol, repeated intrusions are detected with high probability. In each sequence of $\frac{1}{p}$ intrusions, one intrusion is

expected to be detected. As soon as an intrusion is detected, the existence of a foreign sensor node is confirmed and authentication probability is increased to a high level. The detail of the protocol is given below.

1) Every sensor node is given a unique *id* and a *security code* (set up before deployment). All of the id–security code pairs are stored at the base station. During data transmission from the sensors (or the cluster heads), the cluster heads (or base station, respectively) randomly verify the sender of the data item as follows:
   a. When a cluster head (or the base station) receives a data item from a sensor (or cluster head), it generates a random number x, where $0 \le x \le 1$. If $x \le p$, the cluster head (or the base station) requests the sensor to send its security code. Initially p is low (such as 0.1).
   b. The sensor (or cluster head) sends its encrypted security code using a key to the cluster-head (or the base station, respectively).
   c. After receiving the security code, the cluster head sends the id–security code pair to the base station for verification. Since the network is reconfigured often and role of a cluster head rotates among the nodes, building the id-Security code database at the cluster heads is costly.
2) When an intrusion indicating the presence of a foreign sensor is detected by a cluster head, it sends an alarm signal to the base station.
3) The base station sets the probability p to 1 and notifies the change to all cluster heads. This high probability remains active until the intruder is identified.
4) In addition, the more powerful base station periodically analyzes and checks consistencies of data. Since sensor nodes measure the environmental phenomena, sensed data is spatially continuous, that is the value sensed by the neighboring nodes should not differ drastically. Any significant difference indicates either a disaster in the area or an attack by an intruder. In such a case, the base station sets the probability p to 1 and notifies the cluster heads.

6 Simulation Results

The goals of the experiments are to evaluate the energy consumption in cluster formation phase and to assess cluster quality in LLC, Localized, and LEACH protocols.

6.1 Experimental Setup

200 sensor nodes are randomly (uniform distribution) distributed in a square area 200m×200m and the base station (data-sink) is considered to be at 200m from the center of the sensor field. For the sake of fairness, every measured parameter is computed by averaging 50 different random distributions of the nodes. Considering that the self-configuration process needs to be repeated over the lifespan of a sensor network and the nodes already utilized a portion of their energy, the current energy of each node is randomly selected from the range of 30 to 50 Joules.

The power and transmission related specifications are collected from [8, 13, 19]. They are consistent with specifications for the motes developed at the University of California, Berkeley [1]. We compute the radio path loss with an empirical r^3 model. The energy consumption to transmit k bits to distance r is given by $k(E_{tx} + E_{rad}r^3)$, where E_{tx} is the energy consumed by the radio electronics to transmit and receive one bit of data, and E_{rad} is the radio path loss per bit per cubic meter. The simulation parameters are given in Table 1.

Table 1 Simulation Parameters for Power Usage and Transmission

Parameters	Value
Digital Electronics	11 mW
Radio Receiver Electronics	13.5 mW
Radio Idle Listening	13.5 mW
Radio Trans. Electronics	24.8 mW
Radio Sleep Mode	15 μW
Radio Path Loss Rate	200 pJ/bit/m^3
Transmission Rate	20 Kbps

For the purpose of cluster formation, small size messages, which contain node id and some control bits, are used. Assuming a 16-bit id, we use 64-bit messages.

6.2 Results

We assume that every node broadcasts messages up to a specified distance r, called *broadcast radius*. That is, only those nodes that are within the radius r can hear the message. Two nodes are said to be *neighbors* if they can hear each other, i.e. if the distance between them is no larger than r.

Legend Explanation LLC 40 denotes the performance of the LLC protocol with candidate ratio of 40% and so on. LEACH 5 labels performance of the LEACH protocol with the number of cluster heads equal to 5% of the nodes and so on.

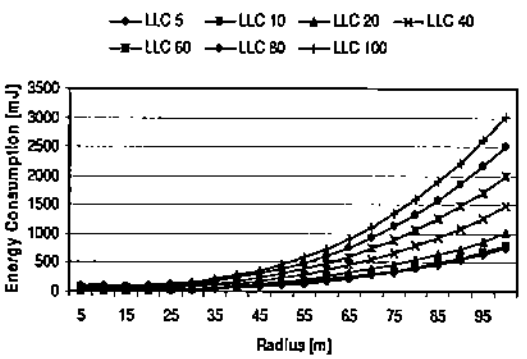

Fig. 3 Energy consumption in cluster formation for the LLC protocol.

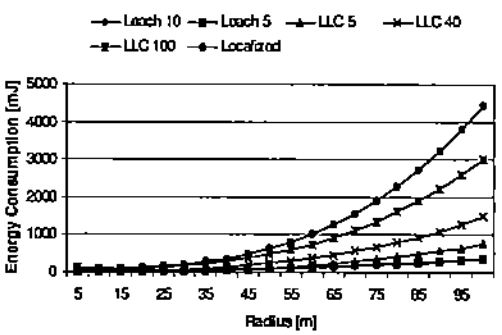

Fig. 4 Comparison of energy consumption in cluster formation for the LLC, Localized, and LEACH protocols.

Energy Consumptions Fig. 3 shows energy consumption in the cluster configuration phase by the LLC protocol for various candidate ratios with a varying broadcast radius. A larger candidate ratio results in a higher energy consumption. Energy consumption in LLC is compared with Localized and LEACH protocol in Fig. 4. LEACH consumes the least amount of energy. Localized protocol consumes the largest amount of energy, which is three times larger than that in LEACH. Energy requirements for LLC are in between those for Localized and LEACH protocols. For all of the protocols, energy consumption increases faster than linearly with the broadcast radius.

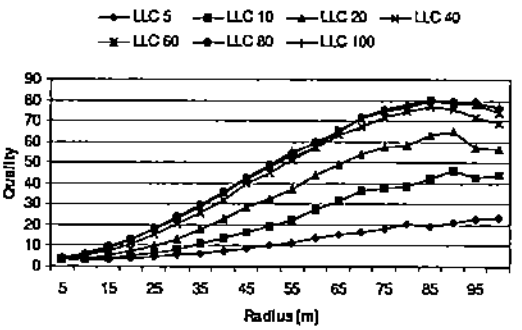

Fig. 5 Cluster quality of the LLC protocol.

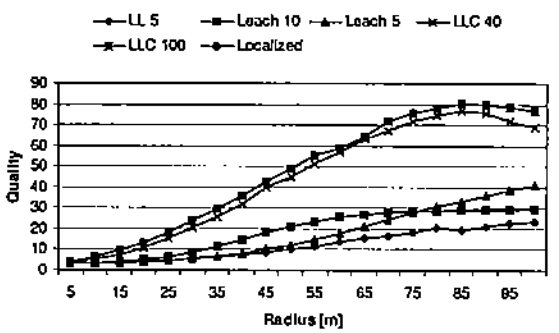

Fig. 6 Comparison of cluster quality of the LLC, LEACH, and Localized protocols.

Cluster Quality For LLC cluster quality increases with the candidate ratio (Fig. 5). As expected, the Localized protocol produces best quality cluster, and LEACH produces worst quality clusters (Fig. 6).

We observe that a lower value of candidate ratio keeps LLC closer to LEACH and a higher candidate ratio keeps LLC closer to Localized protocol. Note that LLC with 100% candidate ratio merges with Localized protocol. An interesting observation is that LLC with the 40% candidate ratio produces almost as good cluster quality as Localized. This phenomenon is described in more detail later in this section.

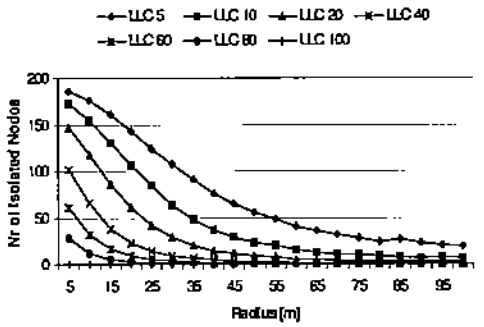

Fig. 7 The number of isolated nodes in the LLC protocol with various candidate ratios

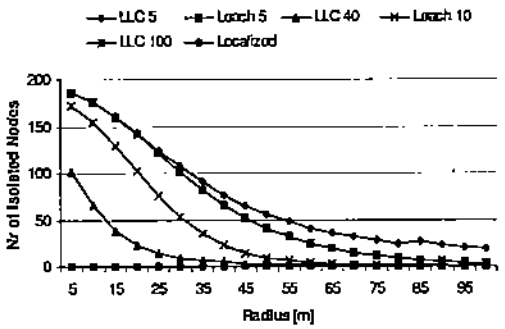

Fig. 8 Comparison of the number of isolated nodes in the LLC, LEACH, and Localized protocols.

Isolated Nodes Fig. 7 and 8 show the number of isolated nodes (which can not hear any cluster head) for LLC, Localized and LEACH protocols. LLC with the 100% candidate ratio and Localized protocol do not have any isolated nodes. Number of isolated nodes decrease when broadcast radius and candidate ratio increases.

Optimal Broadcast Radius The broadcast radius r plays an important role in cluster quality. From the experimental result we see that a smaller r results in a lower energy consumption (Fig. 3 and 4) but in more isolated nodes (Fig. 7 and 8) and worse cluster quality (Fig. 5 and 6). When r is small, there are many small-size clusters, that is many cluster heads need to perform long distance communications to the base station. This requires more energy and thus diminishes the cluster quality.

On the other hand, a larger r results in more energy consumption. Further, when r is very large, there are a few very large clusters with large intra-cluster communication distances, which again reduces cluster quality. Therefore, there is an optimal broadcast radius. Fig. 5 and 6 show that in our simulation setting, the best cluster quality is obtained when r is about 80 m.

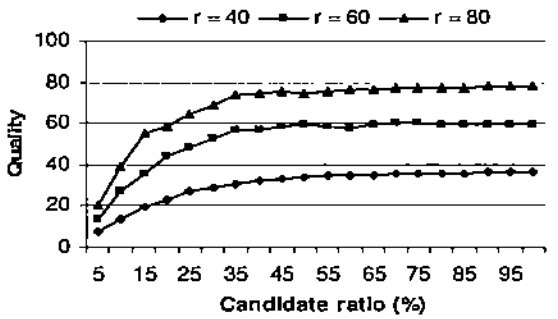

Fig. 9 Cluster quality in LLC with a varying candidate ratio (broadcast radius r = 40, 60, 80 m).

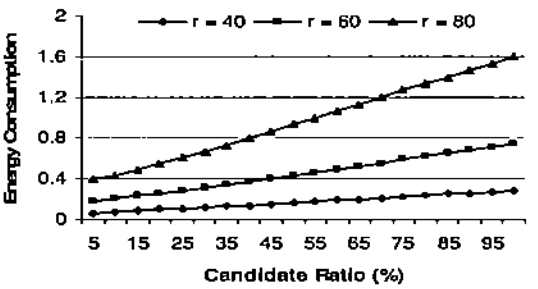

Fig. 10 Energy consumptions in LLC with a varying candidate ratio (broadcast radius r = 40, 60, 80 m).

Best Candidate Ratio In Fig. 6, we observe that LLC with the 40% candidate ratio produces almost the same high cluster quality as both Localized and LLC with the 100% candidate ratio. This means that LLC with the approximately 40% candidate ratio is as good as the Localized protocol (thus saving energy later in the data

transmission phase) while using about three times less energy in the cluster formation phase (figure 4). This phenomenon can be realized more clearly with Fig. 9. The cluster quality saturates at candidate ratio 40-45%, while energy consumption keeps increasing linearly till candidate ratio is 100% (Fig 10). That is, the best candidate ratio is 40-45%, which saves energy both in the cluster formation phase and data transmission phase.

Another factor, frequency of cluster reconfiguration, needs to be considered in selecting the best candidate ratio. If the network is reconfigured very frequently, the LLC protocol with smaller candidate ratio (such as 5%) is a better choice to reduce energy consumption in configuration phase. On the other hand, if the network needs to be reconfigured seldom, the LLC protocol with larger candidate ratio such as 45% (increasing above 45% does not increase cluster quality) is the best choice to reduce energy consumption in data transmission phase. To adapt with a dynamic environment, candidate ratio can be dynamically adjusted to minimize the total energy consumption.

7 Conclusions

The proposed self-configuring protocol, called Low-energy Localized Clustering (LLC), generalizes of the Localized and the LEACH protocols. The ratio of candidates for cluster heads is the parameter used to control the behavior of LLC. The main advantage of LLC is that it can be energy-efficient in cluster configuration phase while maintaining cluster quality, thus saving energy in data transmission phase as well.

A number of data aggregation schemes, such as finding representative data items, and pattern matching have been proposed to provide an efficient way of processing data in a sensor environment.

An energy-efficient Randomized Data Authentication protocol has been developed.

References

[1] Mote Documentation and Development Information, http://www.cs.berkeley.edu/~awoo/smartdust. UC Berkeley, CA.

[2] N. Bulusu, D. Estrin, L. Girod and J. Heidemann, "Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems," in *Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA 2001)*, Ambleside, United Kingdom, July 2001.

[3] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," in *Proceedings of the Twenty First International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, June 2002.

[4] MICA2 Wireless Measurement Sheet, http://www.xbow.com/, Crossbow Technology Inc., San Jose, CA.

[5] K. A. Delin and S. P. Jackson, "Sensor Web for In Situ Exploration of Gaseous Biosignatures," in *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2000.

[6] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in sensor Networks," in *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCom)*, Seattle, WA, August 1999.

[7] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, D. Ganesan, "Building Efficient Wireless Sensor Networks with Low-Level Naming," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, October 2001.

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrisnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," in *Proceedings of the 33rd International Conference on System Sciences (HICSS)*, January 2000.

[9] W. Heinzelman, "Application-Specific Protocol Architectures for Wireless Networks," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, June 2000.

[10] C. Itanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of the Sixth Annual*

International Conference on Mobile Computing and Networks (MobiCom), 2000.

[11] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling Data-Centric Routing in Wireless Sensor Netowrks," In *Proceedings of IEEE INFOCOM*, NY, June 2002.

[12] B. Krishanamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," in *Proceedings of the International Workshop on Distributed Event Based Systems (DEBS)*, Vienna, Austria, July 2002.

[13] R. Min and A. Chandrakasan, "Energy-Efficient Communication for Ad-Hoc Wireless Sensor Networks", *35th Asilomar Conference on Signals, Systems, and Computers*, vol. 1, November 2001, pp. 139-143.

[14] A. Oppenheim, *Applications of Digital Signal Processing*, Prentice-Hall, Inc., 1978.

[15] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networks (MobiCom)*, pp.189-199, 2001.

[16] B. Schneier, *Applied Cryptography*, John Wiley and Sons, 1995.

[17] S. Z. Selim and M. A. Ismail, "*k*-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of the Local Optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*," 6(1), pp. 81-87, 1994.

[18] K. Yao, R. Hudson, C. Reed, D. Chen, and F. Lorenzelli, "Blind Beamforming on a Randomly Distributed Sensors Array System," in *Proceedings of the 1998 IEEE Workshop on Signal Processing Systems (SiPS '98)*, October 1998.

[19] W. Ye, John Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June, 2002.