An Application-Specific Protocol Architecture for Wireless Microsensor Networks

Wendi B. Heinzelman, *Member, IEEE*, Anantha P. Chandrakasan, *Senior Member, IEEE*, and Hari Balakrishnan, *Member, IEEE*

***Abstract*—Networking together hundreds or thousands of cheap microsensor nodes allows users to accurately monitor a remote environment by intelligently combining the data from the individual nodes. These networks require robust wireless communication protocols that are energy efficient and provide low latency. In this paper, we develop and analyze low-energy adaptive clustering hierarchy (LEACH), a protocol architecture for microsensor networks that combines the ideas of energy-efficient cluster-based routing and media access together with application-specific data aggregation to achieve good performance in terms of system lifetime, latency, and application-perceived quality. LEACH includes a new, distributed cluster formation technique that enables self-organization of large numbers of nodes, algorithms for adapting clusters and rotating cluster head positions to evenly distribute the energy load among all the nodes, and techniques to enable distributed signal processing to save communication resources. Our results show that LEACH can improve system lifetime by an order of magnitude compared with general-purpose multihop approaches.**

***Index Terms*—Data aggregation, protocol architecture, wireless microsensor networks.**

I. Introduction

ADVANCES iN sensor technology, low-power electronics, and low-power radio frequency (RF) design have enabled the development of small, relatively inexpensive and low-power sensors, called *microsensors*, that can be connected via a wireless network. These wireless microsensor networks represent a new paradigm for extracting data from the environment and enable the reliable monitoring of a variety of environments for applications that include surveillance, machine failure diagnosis, and chemical/biological detection. An important challenge in the design of these networks is that two key resources—communication bandwidth and energy—are significantly more limited than in a tethered network environment. These constraints require innovative design techniques to use the available bandwidth and energy efficiently.

Manuscript received January 9, 2001; revised July 1, 2001 and August 24, 2001; accepted August 24, 2001. The editor coordinating the review of this paper and approving it for publication is M. Zorzi. The work of W. B. Heinzelman was supported by a Kodak Fellowship. This work was supported in part by the Defense Advanced Research Project Agency (DARPA) Power Aware Computing/Communication Program and the U. S. Air Force Research Laboratory, Air Force Materiel Command, under Agreement F30602-00-2-0551.

W. B. Heinzelman was with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA. She is now with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627-0126 USA (e-mail: wheinzel@ece.rochester.edu).

A. P. Chandrakasan and H. Balakrishnan are with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: anantha@mtl.mit.edu; hari@lcs.mit.edu).

Digital Object Identifier 10.1109/TWC.2002.804190

In order to design good protocols for wireless microsensor networks, it is important to understand the parameters that are relevant to the sensor applications. While there are many ways in which the properties of a sensor network protocol can be evaluated, we use the following metrics.

A. Ease of Deployment

Sensor networks may contain hundreds or thousands of nodes, and they may need to be deployed in remote or dangerous environments, allowing users to extract information in ways that would not have been possible otherwise. This requires that nodes be able to communicate with each other even in the absence of an established network infrastructure and predefined node locations.

B. System Lifetime

These networks should function for as long as possible. It may be inconvenient or impossible to recharge node batteries. Therefore, all aspects of the node, from the hardware to the protocols, must be designed to be extremely energy efficient.

C. Latency

Data from sensor networks are typically time sensitive, so it is important to receive the data in a timely manner.

D. Quality

The notion of "quality" in a microsensor network is very different than in traditional wireless data networks. For sensor networks, the end user does not require all the data in the network because 1) the data from neighboring nodes are highly correlated, making the data redundant and 2) the end user cares about a higher-level description of events occurring in the environment being monitored. The quality of the network is, therefore, based on the quality of the aggregate data set, so protocols should be designed to optimize for the unique, application- specific quality of a sensor network.

This paper builds on the work described in [11] by giving a detailed description and analysis of low-energy adaptive clustering hierarchy (LEACH), an application-specific protocol architecture for wireless microsensor networks. LEACH employs the following techniques to achieve the design goals stated: 1) randomized, adaptive, self-configuring cluster formation; 2) localized control for data transfers; 3) low-energy media access control (MAC); and 4) application-specific data processing, such as data aggregation or compression. Simulation results show that LEACH is able to achieve the desired properties of sensor networks.

1536-1276/02$17.00 © 2002 IEEE

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.

II. Background

Since both device and battery technology have only recently matured to the point where microsensor nodes are feasible, this is a fairly new field of study. Researchers have begun discussing not only the uses and challenges facing sensor networks [2], [7], [20], but have also been developing preliminary ideas as to how these networks should function [4], [5], [13] as well as the appropriate low-energy architecture for the sensor nodes themselves [6], [21].

There have been some application-specific protocols developed for microsensor networks. Clare *et al.* developed a time-divison multiple-access (TDMA) MAC protocol for low-energy operation [5]. Using a TDMA approach saves energy by allowing the nodes to remain in the sleep state, with radios powered-down, for a long time. Intanagonwiwat *et al.* developed directed diffusion, a protocol that employs a data-driven model to achieve low-energy routing [13].

Recently, there has been much work on "power-aware" routing protocols for wireless networks [19], [25]. In these protocols, optimal routes are chosen based on the energy at each node along the route. Routes that are longer, but which use nodes with more energy than the nodes along the shorter routes, are favored, helping avoid "hot spots" in the network. In LEACH, we use randomized rotation of the cluster head positions to achieve the same goal.

One method of choosing routes is to use "minimum transmission energy" (MTE) routing [8], [24], where intermediate nodes are chosen such that the sum of squared distances (and, hence, the total transmit energy $E_{TX}(d)$, assuming a d^2 power loss) is minimized. Thus, for three nodes A, B, and C, node A would transmit to node C through node B if and only if

$$E_{TX}(d = d_{AB}) + E_{TX}(d = d_{BC}) < E_{TX}(d = d_{AC}) \quad (1)$$

or $d_{AB}^2 + d_{BC}^2 < d_{AC}^2$. This approach ignores the energy dissipated in the radio to send and receive the data and, therefore, may not actually produce the lowest energy routes.

Another method of wireless communication is to use *clustering*. In this case, nodes send their data to a central *cluster head* that forwards the data to get it closer to the desired recipient. Clustering enables bandwidth reuse and can, thus, increase system capacity. Using clustering enables better resource allocation and helps improve power control [14].

While conventional cluster-based networks rely on a fixed infrastructure, new research is focusing on ways to deploy clustering architectures in an ad-hoc fashion [3], [15], [23]. Early work by Baker *et al.* developed a linked cluster architecture, where nodes are assigned to be either ordinary nodes, cluster head nodes, or gateways between different clusters [3]. The cluster heads act as local control centers, whereas the gateways act as the backbone network, transporting data between clusters. This enables robust networking with point-to-point connectivity. Another ad-hoc clustering protocol, the near term digital radio (NTDR), uses a clustering approach with a two-tier hierarchical routing algorithm [23]. Nodes form local clusters, and intra-cluster data are sent directly from one node to the next, whereas inter-cluster data are routed through the cluster head nodes. This protocol enables point-to-point connectivity and does not use low-energy routing or MAC; therefore, it is not suited for microsensor networks. LEACH builds on this work by creating a new ad-hoc cluster formation algorithm that better suits microsensor network applications.

III. LEACH Protocol Architecture

To meet the unique requirements of wireless microsensor networks, we developed LEACH, an application-specific protocol architecture [10], [11]. The application that typical microsensor networks support is the monitoring of a remote environment. Since individual nodes' data are often correlated in a microsensor network, the end user does not require all the (redundant) data; rather, the end user needs a high-level function of the data that describes the events occurring in the environment. Because the correlation is strongest between data signals from nodes located close to each other, we chose to use a clustering infrastructure as the basis for LEACH. This allows all data from nodes within the cluster to be processed locally, reducing the data set that needs to be transmitted to the end user. In particular, data aggregation techniques can be used to combine several correlated data signals into a smaller set of information that maintains the *effective* data (i.e., the information content) of the original signals [9]. Therefore, much less actual data needs to be transmitted from the cluster to the base station (BS).

For the development of LEACH, we made some assumptions about the sensor nodes and the underlying network model. For the sensor nodes, we assume that all nodes can transmit with enough power to reach the BS if needed, that the nodes can use power control to vary the amount of transmit power, and that each node has the computational power to support different MAC protocols and perform signal processing functions. These assumptions are reasonable due to technological advances in radio hardware and low-power computing. For the network, we use a model where nodes always have data to send to the end user and nodes located close to each other have correlated data. Although LEACH is optimized for this situation, it will continue to work if it were not true. In Section V, we discuss ways in which LEACH may be improved when these assumptions do not hold.

In LEACH, the nodes organize themselves into local clusters, with one node acting as the cluster head. All non-cluster head nodes transmit their data to the cluster head, while the cluster head node receives data from all the cluster members, performs signal processing functions on the data (e.g., data aggregation), and transmits data to the remote BS. Therefore, being a cluster head node is much more energy intensive than being a non-cluster head node. If the cluster heads were chosen *a priori* and fixed throughout the system lifetime, these nodes would quickly use up their limited energy. Once the cluster head runs out of energy, it is no longer operational, and all the nodes that belong to the cluster lose communication ability. Thus, LEACH incorporates randomized rotation of the high-energy cluster head position among the sensors to avoid draining the battery of any one sensor in the network. In this way, the energy load of being a cluster head is evenly distributed among the nodes.

The operation of LEACH is divided into *rounds*. Each round begins with a set-up phase when the clusters are organized, followed by a steady-state phase when data are transfered from the nodes to the cluster head and on to the BS, as shown in

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.

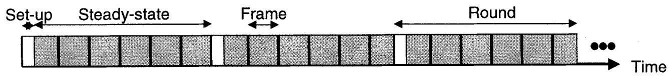


Fig. 1. Time line showing LEACH operation. Adaptive clusters are formed during the set-up phase and data transfers occur during the steady-state phase.

Fig. 1. The following sections describe the cluster head selection and distributed cluster formation algorithms and the steady-state operation of LEACH.

A. Cluster Head Selection Algorithms

LEACH forms clusters by using a distributed algorithm, where nodes make autonomous decisions without any centralized control. Our goal is to design a cluster formation algorithm such that there are a certain number of clusters, k, during each round. In addition, if nodes begin with equal energy, our goal is to try to evenly distribute the energy load among all the nodes in the network so that there are no overly-utilized nodes that will run out of energy before the others. As being a cluster head node is much more energy intensive than being a non-cluster head node, this requires that each node take its turn as cluster head.

Each sensor i elects itself to be a cluster head at the beginning of round $r+1$ (which starts at time t) with probability $P_i(t)$. $P_i(t)$ is chosen such that the expected number of cluster head nodes for this round is k. Thus, if there are N nodes in the network

$$E[\#CH] = \sum_{i=1}^{N} P_i(t) * 1 = k. \tag{2}$$

Ensuring that all nodes are cluster heads the same number of times requires each node to be a cluster head once in N/k rounds on average. If $C_i(t)$ is the indicator function determining whether or not node i has been a cluster head in the most recent ($r \bmod (N/k)$) rounds (i.e., $C_i(t) = 0$ if node i has been a cluster head and one otherwise), then each node should choose to become a cluster head at round r with probability

$$P_i(t) = \begin{cases} \frac{k}{N-k*(r \bmod \frac{N}{k})} & : \quad C_i(t) = 1 \\ 0 & : \quad C_i(t) = 0 \end{cases}. \tag{3}$$

Therefore, only nodes that have not already been cluster heads recently, and which presumably have more energy available than nodes that have recently performed this energy-intensive function, may become cluster heads at round $r+1$.

The expected number of nodes that have not been cluster heads in the first r rounds is $N - k * r$. After N/k rounds, all nodes are expected to have been cluster head once, following which they are all eligible to perform this task in the next sequence of rounds. Since $C_i(t)$ is one if node i is eligible to be a cluster head at time t and zero otherwise, the term $\sum_{i=1}^{N} C_i(t)$ represents the total number of nodes that are eligible to be a cluster head at time t and

$$E\left[\sum_{i=1}^{N} C_i(t)\right] = N - k * \left(r \bmod \frac{N}{k}\right). \tag{4}$$

This ensures that the energy at all nodes are approximately equal to each other after every N/k rounds. Using (3) and (4), the expected number of cluster heads per round is

$$\begin{aligned} E[\#CH] &= \sum_{i=1}^{N} P_i(t) * 1 \\ &= \left(N - k * \left(r \bmod \frac{N}{k}\right)\right) * \frac{k}{N - k * \left(r \bmod \frac{N}{k}\right)} \\ &= k. \end{aligned} \tag{5}$$

In Section IV-B, we analytically determine the optimal k based on our energy dissipation models for computation and communication.

This choice of probability for becoming a cluster head is based on the assumption that all nodes start with an equal amount of energy, and that all nodes have data to send during each frame. If nodes have different amounts of energy (or an event-driven model is used, whereby nodes only send data when some event occurs in the environment), the nodes with more energy should be cluster heads more often than the nodes with less energy, to ensure that all nodes die at approximately the same time. This can be achieved by setting the probability of becoming a cluster head as a function of a node's energy level relative to the aggregate energy remaining in the network, rather than purely as a function of the number of times the node has been cluster head, Thus

$$P_i(t) = \min\left\{\frac{E_i(t)}{E_{\text{total}}(t)} k, 1\right\} \tag{6}$$

where $E_i(t)$ is the current energy of node i and

$$E_{\text{total}}(t) = \sum_{i=1}^{N} E_i(t). \tag{7}$$

Using these probabilities, the nodes with higher energy are more likely to become cluster heads than nodes with less energy. The expected number of cluster head nodes is[1]

$$E[\#CH] = \sum_{i=1}^{N} P_i(t) * 1 = \left(\frac{E_1(t)}{E_{\text{total}}} + \cdots + \frac{E_N(t)}{E_{\text{total}}}\right) k = k. \tag{8}$$

Equation (6) can be approximated by (3) when the nodes begin with equal energy [10].

To use the probabilities in (6), each node must have an estimate of the total energy of all nodes in the network. This requires a routing protocol that allows each node to determine the total energy, whereas the probabilities in (3) enable each node to make completely autonomous decisions. One approach to avoid this might be to approximate the aggregate node energy by multiplying the average energy of the nodes in each cluster by N.

Note that to compute the probabilities in (3) and (6) requires that each node knows the parameters k and N. In this paper, we assume these parameters are programmed into the nodes *a priori*. However, this approach does not work well in dynamic networks. As we show in Section IV-B, the optimal number of clusters k is a function of the number of nodes N distributed throughout an $M \times M$ region of space. Therefore, the nodes

[1]Note that if any node i has $E_i > (E_{\text{total}}/k)$, which occurs with a small but nonzero probability, the expected number of cluster heads will be less than k.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.

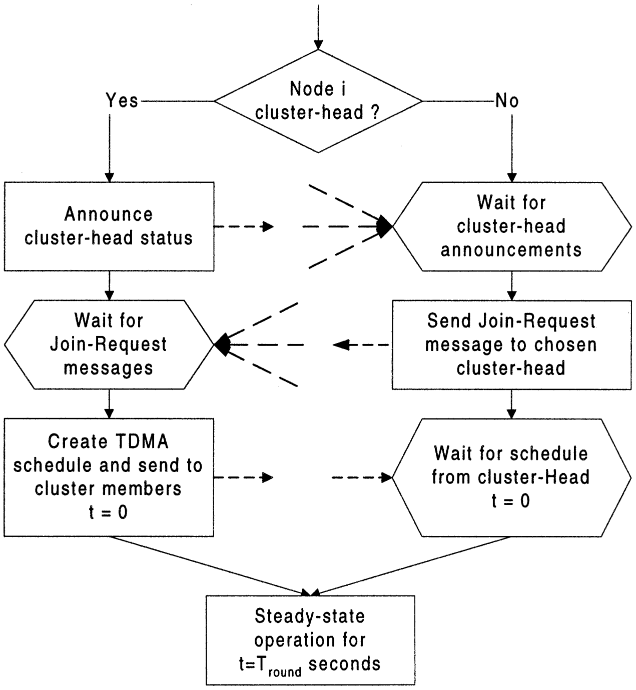


Fig. 2. Flowchart of the distributed cluster formation algorithm for LEACH.

only need to determine N assuming there is a predefined specification for M. To do this, nodes can send "hello" messages to all neighbors within a predetermined number of hops (to approximate M). Each node can count the number of "hello" messages it receives—this is that node's estimate for N. The desired number of clusters k can then be determined based on these parameters. This approach allows LEACH to adapt to changing networks at the cost of increased overhead.

B. Cluster Formation Algorithm

Once the nodes have elected themselves to be cluster heads using the probabilities in (3) or (6), the cluster head nodes must let all the other nodes in the network know that they have chosen this role for the current round. To do this, each cluster head node broadcasts an advertisement message (ADV) using a nonpersistent carrier-sense multiple access (CSMA) MAC protocol [18]. This message is a small message containing the node's ID and a header that distinguishes this message as an announcement message. Each non-cluster head node determines its cluster for this round by choosing the cluster head that requires the minimum communication energy, based on the *received signal strength* of the advertisement from each cluster head. Assuming symmetric propagation channels for pure signal strength, the cluster head advertisement heard with the largest signal strength is the cluster head that requires the minimum amount of transmit energy to communicate with. Note that typically this will be the cluster head closest to the sensor, unless there is an obstacle impeding communication. In the case of ties, a random cluster head is chosen.

After each node has decided to which cluster it belongs, it must inform the cluster head node that it will be a member of the cluster. Each node transmits a join-request message (Join-REQ) back to the chosen cluster head using a nonpersistent CSMA MAC protocol. This message is again a short message, consisting of the node's ID and the cluster head's ID.

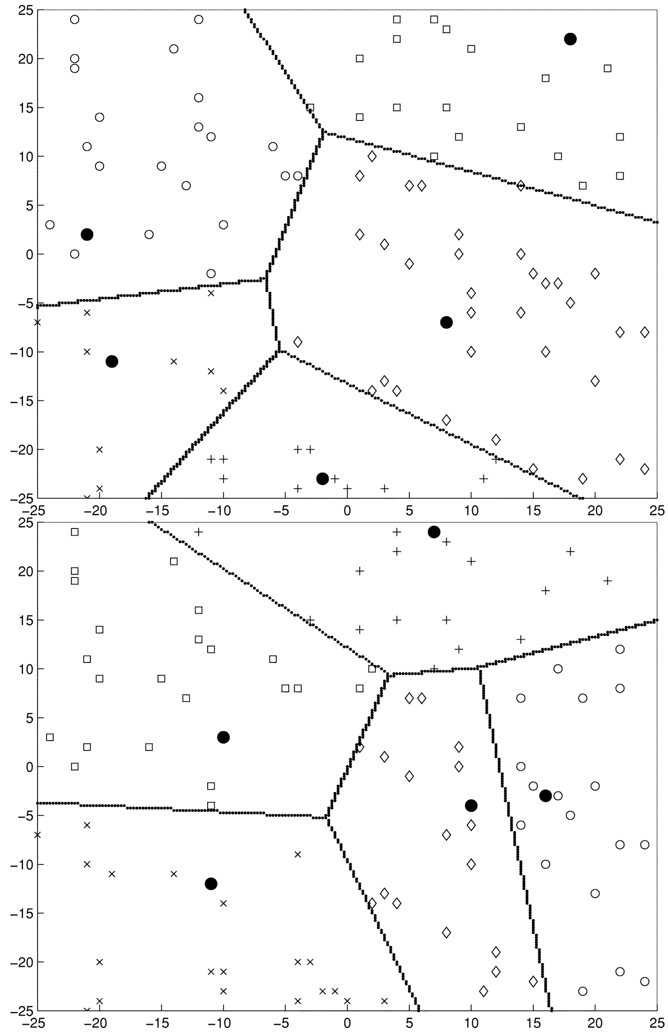


Fig. 3. Dynamic cluster formation during two different rounds of LEACH. All nodes marked with a given symbol belong to the same cluster, and the cluster head nodes are marked with •.

The cluster heads in LEACH act as local control centers to coordinate the data transmissions in their cluster. The cluster head node sets up a TDMA schedule and transmits this schedule to the nodes in the cluster. This ensures that there are no collisions among data messages and also allows the radio components of each non-cluster head node to be turned off at all times except during their transmit time, thus reducing the energy consumed by the individual sensors. After the TDMA schedule is known by all nodes in the cluster, the set-up phase is complete and the steady-state operation (data transmission) can begin. A flowchart of this distributed cluster formation algorithm is shown in Fig. 2. Fig. 3 shows an example of the clusters formed during two different rounds of LEACH.

C. Steady-State Phase

The steady-state operation is broken into frames, where nodes send their data to the cluster head at most once per frame during their allocated transmission slot. The duration of each slot in which a node transmits data is constant, so the time to send a

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.

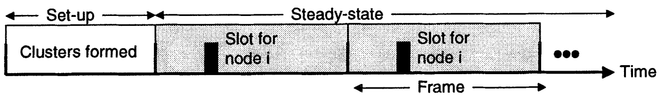


Fig. 4. Time line showing LEACH operation. Data transmissions are explicitly scheduled to avoid collisions and increase the amount of time each non-cluster head node can remain in the sleep state.

frame of data depends on the number of nodes in the cluster. Fig. 4 shows the time line for one round of LEACH. We assume that the nodes are all time synchronized and start the set-up phase at the same time. This could be achieved, for example, by having the BS send out synchronization pulses to the nodes.

To reduce energy dissipation, each non-cluster head node uses power control to set the amount of transmit power based on the received strength of the cluster head advertisement.[2] Furthermore, the radio of each non-cluster head node is turned off until its allocated transmission time. Since we optimize our design for the situation when all the nodes have data to send to the cluster head, using a TDMA schedule is an efficient use of bandwidth and represents a low-latency and energy-efficient approach.

The cluster head must be awake to receive all the data from the nodes in the cluster. Once the cluster head receives all the data, it performs data aggregation to enhance the common signal and reduce the uncorrelated noise among the signals. In our analysis, we assume perfect correlation such that all individual signals can be combined into a single representative signal. The resultant data are sent from the cluster head to the BS. Since the BS may be far away and the data messages are large, this is a high-energy transmission.

The preceding discussion describes communication within a cluster, where the MAC and routing protocols are designed to ensure low energy dissipation in the nodes and no collisions of data messages within a cluster. However, radio is inherently a broadcast medium. As such, transmission in one cluster will affect (and often degrade) communication in a nearby cluster. To reduce inter-cluster interference, each cluster in LEACH communicates using direct-sequence spread spectrum (DSSS). Each cluster uses a unique spreading code; all the nodes in the cluster transmit their data to the cluster head using this spreading code and the cluster head filters all received energy using this spreading code. This is known as *transmitter-based code assignment* [12], since all transmitters within the cluster use the same code. The first cluster head to advertise its position is assigned the first code on a predefined list, the second cluster head to advertise its position is assigned the second code, and so on.[3] With enough spreading, neighboring clusters' radio signals will be filtered out as noise during decorrelation and not corrupt the transmission from nodes in the cluster. To reduce the possibility of interfering with nearby clusters and reduce its own energy dissipation, each node adjusts its transmit power. Therefore, there will be few overlapping transmissions and little spreading of the data is actually needed to ensure a low probability of collision.

Data is sent from the cluster head nodes to the BS using a fixed spreading code and CSMA. When a cluster head has data to send (at the end of its frame), it must sense the channel to see if anyone else is transmitting using the BS spreading code. If so, the cluster head waits to transmit the data. Otherwise, the cluster head sends the data using the BS spreading code.

Other channelization techniques, such as having each cluster use a different frequency band (e.g., FDMA), are possible. However, since the number of clusters in LEACH is not fixed, using DSSS ensures that all nodes will receive better communication channels when there are fewer clusters. It is much harder to dynamically assign frequency bands so that all the bandwidth is utilized in a fixed channelization scheme. Of course, the drawback of using DSSS is the need for tight timing synchronization, which may necessitate extra communication between the cluster head and the non-cluster head nodes.

D. LEACH-C: BS Cluster Formation

While there are advantages to using LEACHs distributed cluster formation algorithm, this protocol offers no guarantee about the placement and/or number of cluster head nodes. Since the clusters are adaptive, obtaining a poor clustering set-up during a given round will not greatly affect overall performance. However, using a central control algorithm to form the clusters may produce better clusters by dispersing the cluster head nodes throughout the network. This is the basis for LEACH-centralized (LEACH-C), a protocol that uses a centralized clustering algorithm and the same steady-state protocol as LEACH.

During the set-up phase of LEACH-C, each node sends information about its current location (possibly determined using a GPS receiver) and energy level to the BS. In addition to determining good clusters, the BS needs to ensure that the energy load is evenly distributed among all the nodes. To do this, the BS computes the average node energy, and whichever nodes have energy below this average cannot be cluster heads for the current round. Using the remaining nodes as possible cluster heads, the BS finds clusters using the simulated annealing algorithm [16] to solve the NP-hard problem of finding k optimal clusters [1]. This algorithm attempts to minimize the amount of energy for the non-cluster head nodes to transmit their data to the cluster head, by minimizing the total sum of squared distances between all the non-cluster head nodes and the closest cluster head.[4]

Once the cluster heads and associated clusters are found, the BS broadcasts a message that contains the cluster head ID for each node. If a node's cluster head ID matches its own ID, the node is a cluster head; otherwise, the node determines its TDMA slot for data transmission and goes to sleep until it is time to transmit data. The steady-state phase of LEACH-C is identical to that of LEACH.

[2]To ensure connectivity in a dynamic environment, the node can either set its transmit power slightly greater than the minimum needed to reach the cluster head, or the cluster head can send short feedback messages to each of the nodes telling them to increase or decrease their transmitted power, as is done in cellular systems.

[3]If there are more clusters than spreading codes, some clusters will use the same code, possibly causing data collisions if the clusters are located close to each other.

[4]Communication energy often does not scale exactly with distance. However, gathering information about the communication channel between all nodes is impractical. Using distance calculated from the nodes' GPS coordinates is, therefore, an approximation to the energy that will be required for communication.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.

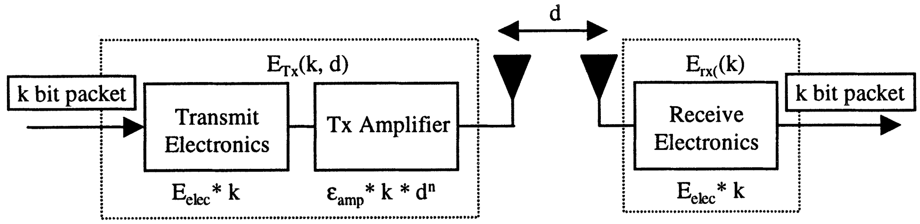


Fig. 5. Radio energy dissipation model.

IV. Analysis and Simulation of LEACH

For even moderately-sized networks with tens of nodes, it is extremely difficult to analytically model the interactions between all the nodes. Therefore, we used the network simulator ns [17] to evaluate LEACH and compare it to other protocols. We compare LEACH to LEACH-C, MTE routing, and static clustering in terms of system lifetime, energy dissipation, amount of data transfer, and latency. The code for our experiments can be found at http://www-mtl.mit.edu/research/icsystems/uamps/cadtools.

For MTE routing, each node runs a start-up routine to determine its next-hop neighbor, defined to be the closest node that is in the direction of the BS. We assume that each node knows the location of all nodes in the network, to simplify the set-up of MTE routes. In general, some sort of initialization phase would be needed where this information is disseminated through the network. Data packets are passed along via next-hop neighbors until they reach the BS. As there is no central control in MTE routing, it is difficult to set up fixed MAC protocols (e.g., TDMA), so each node uses CSMA to listen to the channel before transmitting data. If the channel is busy, the node backs off; otherwise, the node transmits its data to the next-hop node. As nodes run out of energy, the routes are recomputed to ensure connectivity with the BS. We do not account for the energy requirements or delay for such updates in our simulations. Each node transmits its own data once every t_{delay} seconds, where t_{delay} is set to minimize congestion but ensure efficient use of the channel bandwidth. If t_{delay} is too small, nodes end up sending their own data before the previous set of data was able to reach the BS. Large queues will build up, there will be many collisions of data, and nothing will be transmitted to the BS. If t_{delay} is too large, the channel is idle when it could be used for data transmission. We set t_{delay} based on N, the total number of nodes in the network, the average number of hops to get a message to the BS, and the time it takes a message to traverse a single hop.

For static clustering, nodes are organized into clusters initially by the BS using the same method as in LEACH-C to ensure that good clusters are formed. These clusters and cluster heads remain fixed throughout the lifetime of the network. As in LEACH and LEACH-C, nodes transmit their data to the cluster head node during each frame of data transfer (using TDMA and a DSSS spreading code to ensure minimal inter-cluster interference), and the cluster head aggregates the data and sends the resultant data to the BS. When the cluster head node's energy is depleted, the nodes in the cluster lose communication ability with the BS and are essentially "dead."

A. Experiment Setup

For our experiments, we used a 100-node network where nodes were randomly distributed between ($x = 0$, $y = 0$) and ($x = 100$, $y = 100$) with the BS at location ($x = 50$, $y = 175$). The bandwidth of the channel was set to 1 Mb/s, each data message was 500 bytes long, and the packet header for each type of packet was 25 bytes long.

We assume a simple model for the radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics, as shown in Fig. 5. For the experiments described here, both the free space (d^2 power loss) and the multipath fading (d^4 power loss) channel models were used, depending on the distance between the transmitter and receiver[5] [22]. Power control can be used to invert this loss by appropriately setting the power amplifier—if the distance is less than a threshold d_o, the free space (fs) model is used; otherwise, the multipath (mp) model is used. Thus, to transmit an l-bit message a distance d, the radio expends

$$\begin{aligned} E_{Tx}(l,d) &= E_{Tx-\text{elec}}(l) + E_{Tx-amp}(l,d) \\ &= \begin{cases} lE_{\text{elec}} + l\epsilon_{\text{fs}}d^2, & d < d_o \\ lE_{\text{elec}} + l\epsilon_{\text{mp}}d^4, & d \geq d_o. \end{cases} \end{aligned} \tag{9}$$

and to receive this message, the radio expends:

$$E_{Rx}(l) = E_{Rx-\text{elec}}(l) = lE_{\text{elec}}. \tag{10}$$

The electronics energy, E_{elec}, depends on factors such as the digital coding, modulation, filtering, and spreading of the signal, whereas the amplifier energy, $\epsilon_{\text{fs}}d^2$ or $\epsilon_{\text{mp}}d^4$, depends on the distance to the receiver and the acceptable bit-error rate. For the experiments described in this paper, the communication energy parameters are set as: $E_{\text{elec}} = 50$ nJ/bit, $\epsilon_{\text{fs}} = 10$ pJ/bit/m^2, and $\epsilon_{\text{mp}} = 0.0013$ pJ/bit/m^4. Using our previous experimental results [26], the energy for data aggregation is set as $E_{DA} = 5$ nJ/bit/signal.

B. Optimum Number of Clusters

In LEACH, the cluster formation algorithm was created to ensure that the expected number of clusters per round is k, a system parameter. We can analytically determine the optimal value of k in LEACH using the computation and communication energy models. Assume that there are N nodes distributed uniformly in an $M \times M$ region. If there are k clusters, there are on average N/k nodes per cluster (one cluster head and $(N/k) - 1$ non-cluster head nodes). Each cluster head dissipates energy

[5]Note that this is a simplified model; in general, radio wave propagation is highly variable and difficult to model.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.

receiving signals from the nodes, aggregating the signals, and transmitting the aggregate signal to the BS. Since the BS is far from the nodes, presumably the energy dissipation follows the multipath model (d^4 power loss). Therefore, the energy dissipated in the cluster head node during a single frame is

$$E_{CH} = lE_{\mathrm{elec}}\left(\frac{N}{k}-1\right) + lE_{DA}\frac{N}{k} + lE_{\mathrm{elec}} + l\epsilon_{mp}d^4_{\mathrm{toBS}} \tag{11}$$

where l is the number of bits in each data message, d_{toBS} is the distance from the cluster head node to the BS, and we have assumed perfect data aggregation.

Each non-cluster head node only needs to transmit its data to the cluster head once during a frame. Presumably the distance to the cluster head is small, so the energy dissipation follows the Friss free-space model (d^2 power loss). Thus, the energy used in each non-cluster head node is

$$E_{\mathrm{non-CH}} = lE_{\mathrm{elec}} + l\epsilon_{\mathrm{fs}}d^2_{\mathrm{toCH}} \tag{12}$$

where d_{toCH} is the distance from the node to the cluster head. The area occupied by each cluster is approximately M^2/k. In general, this is an arbitrary-shaped region with a node distribution $\rho(x,y)$. The expected squared distance from the nodes to the cluster head (assumed to be at the center of mass of the cluster) is given by

$$\begin{aligned} E[d^2_{\mathrm{toCH}}] &= \int\int (x^2+y^2)\rho(x,y)dxdy \\ &= \int\int r^2\rho(r,\theta)rdrd\theta. \end{aligned} \tag{13}$$

If we assume this area is a circle with radius $R = (M/\sqrt{\pi k})$ and $\rho(r,\theta)$ is constant for r and θ, (13) simplifies to

$$E[d^2_{\mathrm{toCH}}] = \rho\int_{\theta=0}^{2\pi}\int_{r=0}^{M/\sqrt{\pi k}} r^3 drd\theta = \frac{\rho}{2\pi}\frac{M^4}{k^2}. \tag{14}$$

If the density of nodes is uniform throughout the cluster area, then $\rho = (1/(M^2/k))$ and

$$E[d^2_{\mathrm{toCH}}] = \frac{1}{2\pi}\frac{M^2}{k}. \tag{15}$$

Therefore, in this case

$$E_{\mathrm{non-CH}} = lE_{\mathrm{elec}} + l\epsilon_{fs}\frac{1}{2\pi}\frac{M^2}{k}. \tag{16}$$

The energy dissipated in a cluster during the frame is

$$E_{\mathrm{cluster}} = E_{\mathrm{CH}} + \left(\frac{N}{k}-1\right)E_{\mathrm{non-CH}} \approx E_{CH} + \frac{N}{k}E_{\mathrm{non-CH}} \tag{17}$$

and the total energy for the frame is

$$\begin{aligned} E_{\mathrm{total}} =& kE_{\mathrm{cluster}} \\ =& l\left(E_{\mathrm{elec}}N + E_{DA}N + k\epsilon_{\mathrm{mp}}d^4_{\mathrm{toBS}} + E_{elec}N \right. \\ & \left. + \epsilon_{fs}\frac{1}{2\pi}\frac{M^2}{k}N\right). \end{aligned} \tag{18}$$

We can find the optimum number of clusters by setting the derivative of E_{total} with respect to k to zero

$$k_{opt} = \frac{\sqrt{N}}{\sqrt{2\pi}}\sqrt{\frac{\epsilon_{\mathrm{fs}}}{\epsilon_{\mathrm{mp}}}}\frac{M}{d^2_{\mathrm{toBS}}}. \tag{19}$$

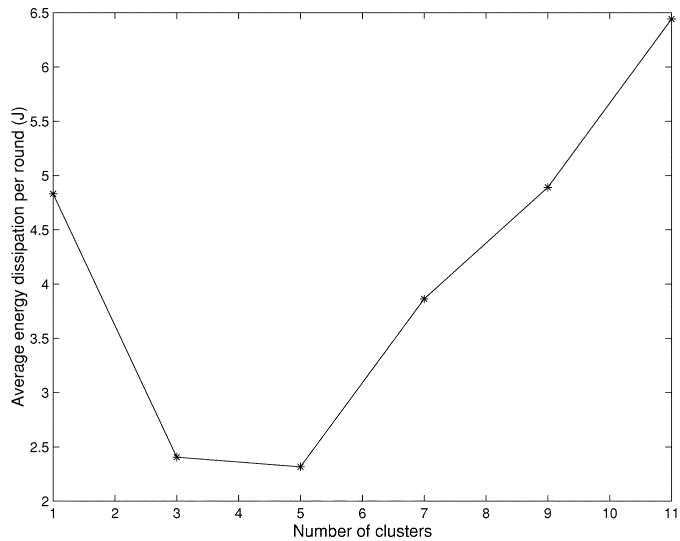


Fig. 6. Average energy dissipated per round in LEACH as the number of clusters is varied between 1 and 11. This graph shows that LEACH is most energy efficient when there are between 3 and 5 clusters in the 100-node network, as predicted by the analysis.

For our experiments, $N = 100$ nodes, $M = 100$ m, $\epsilon_{fs} =$ 10 pJ, $\epsilon_{mp} = 0.0013$ pJ, and 75 m $< d_{\mathrm{toBS}} <$ 185 m, so we expect the optimum number of clusters to be $1 < k_{\mathrm{opt}} < 6$.

These analytical results were verified using simulations on a 100-node network where we varied the number of clusters between 1 and 11 and ran LEACH for 1000 simulated seconds. Note that for these simulations, the nodes were placed randomly throughout the 100 m $\times$ 100 m area and we made no restrictions on the distance between the nodes and their cluster heads (e.g., d_{toCH}) or between the nodes and the BS (e.g., d_{toBS}). Even though we made these assumptions for the analysis, Fig. 6, which shows the average energy dissipated per round as a function of the number of clusters, shows that the simulation agrees well with the analysis. This graph shows that the optimum number of clusters is around 3–5 for the 100-node network. When there is only one cluster, the non-cluster head nodes often have to transmit data very far to reach the cluster head node, draining their energy, and when there are more than five clusters, there is not as much local data aggregation being performed. For the rest of the experiments, we set k to five.

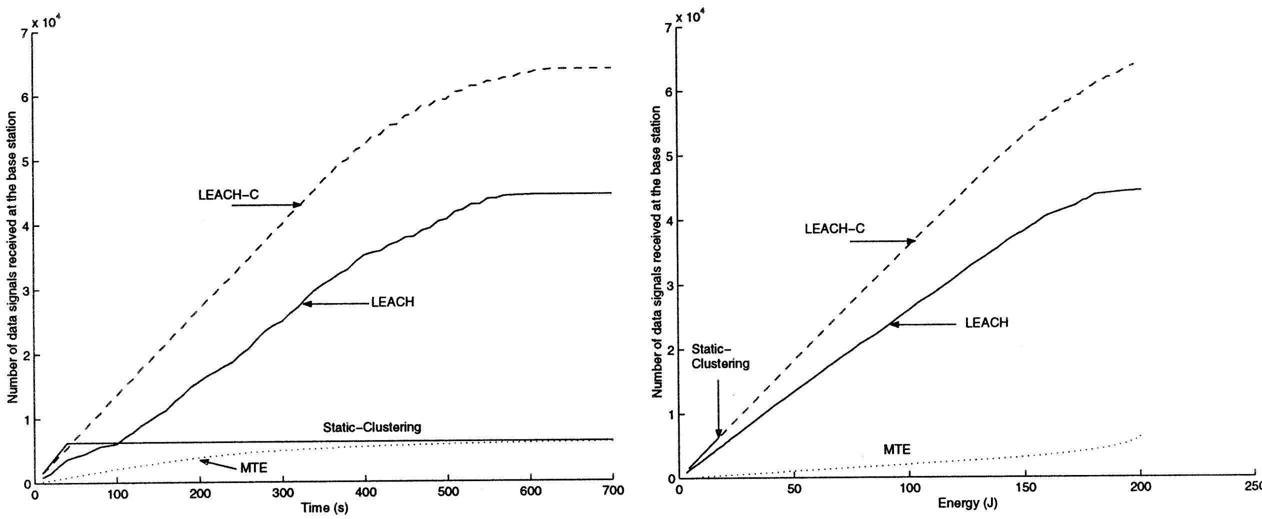
C. Energy Gains

In these experiments, each node begins with only 2 J of energy and an unlimited amount of data to send to the BS. Each node uses the probabilities in (3) to determine its cluster head status at the beginning of each round, and each round lasts for 20 s.[6] We tracked the rate at which the data packets are transfered to the BS and the amount of energy required to get the data to the BS. When the nodes use up their limited energy during the course of the simulation, they can no longer transmit or receive data.

For these simulations, energy is consumed whenever a node transmits or receives data or performs data aggregation. Using spread-spectrum increases the number of bits transmitted,

[6]The time for a round was chosen so that on average each node has enough energy to act as cluster head once and non-cluster head several times throughout the simulation lifetime [10].

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 7. Data for the limited energy simulations, where each node begins with 2 J of energy. (a) Total amount of data received at the BS over time. (b) Total amount of data received at the BS per given amount of energy. These graphs show that LEACH distributes an order of magnitude more data per unit energy than MTE routing, LEACH-C delivers 40% more data per unit energy than LEACH, and static clustering does not perform well when the nodes have limited energy.

thereby increasing the amount of energy dissipated in the electronics of the radio. We do not assume any static energy dissipation nor do we assume energy is consumed during carrier-sense operations; hence, the results here do not account for the potential energy benefits of using TDMA in LEACH compared with CSMA in MTE.

Although quality is an application-specific and data-dependent quantity, one application-independent method of determining quality is to measure the amount of data (number of actual data signals or number of data signals represented by an aggregate signal) received at the BS. The more data the BS receives, the more accurate its view of the remote environment will be. If all the nodes within a cluster are sensing the same event, the actual and effective data will contain the same information, and there is no loss in quality by sending effective or aggregate data. If, on the other hand, the nodes are seeing different events, the cluster head will pick out the strongest event (strongest signal within the signals of the cluster members) and send that as the data from the cluster. In this case, there will be a loss in quality by aggregating signals into a single representative signal. As with radio wave propagation, it is difficult to quantify signal propagation as it depends on factors such as the nature of the signal, the path between the source and the sensor, and the sensitivity of the sensors. If the distance between nodes within a cluster is small compared with the distance from which events can be sensed, or if the distance between events occurring in the environment is large, there is a high probability that the nodes will be sensing the same event. For our experiments, we assume that all nodes in a cluster sense the same events.

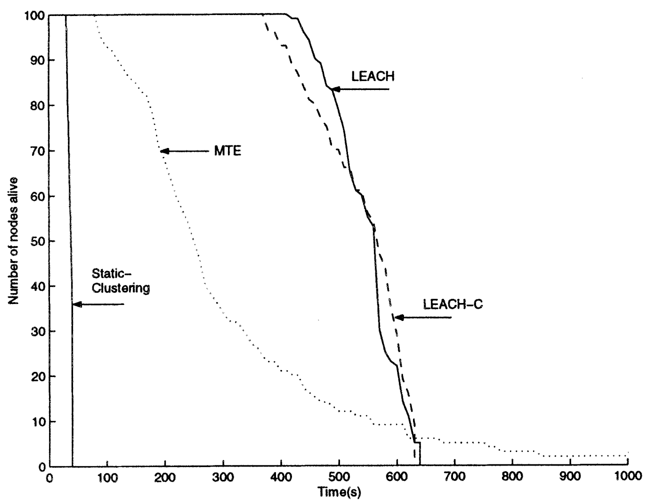
Fig. 7 shows the total number of data signals (actual for MTE and effective for LEACH, LEACH-C, and static clustering) received at the BS over time and the total data received at the BS for a given amount of energy. Fig. 7(a) shows that LEACH sends much more data to the BS in the simulation time than MTE routing. The reason MTE requires so much time to send data from the nodes to the BS is that each message traverses several hops. In the other protocols, each message is transmitted over a single hop, to the cluster head, where data aggregation occurs. The aggregate signals are sent to the BS, greatly reducing the amount of data transmitted. Fig. 7(b) shows the total data received at the BS for a given amount of energy. This graph shows that LEACH and LEACH-C deliver the most data per unit energy, achieving both energy and latency efficiency. A routing protocol such as MTE does not enable local computation to reduce the amount of data that needs to be transmitted to the BS.

Fig. 7 shows that LEACH is not as efficient as LEACH-C (LEACH-C delivers about 40% more data per unit energy than LEACH). This is because the BS has global knowledge of the location and energy of all the nodes in the network, so it can produce better clusters that require less energy for data transmission. In addition, the BS formation algorithm ensures that there are $k = 5$ clusters during each round of operation. As there are only 100 nodes in the simulation, even though the expected number of clusters per round is $k = 5$ in LEACH, each round does not always have five clusters.

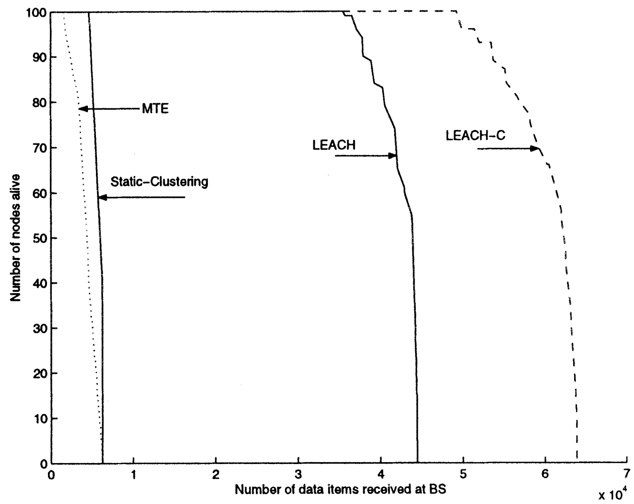
Fig. 8(a) shows the total number of nodes that remain alive over the simulation time. While nodes remain alive for a long time in MTE, this is because a much smaller amount of data has been transmitted to the BS. If we plot the total number of nodes that remain alive per amount of data received at the BS [Fig. 8(b)], we see that nodes in LEACH can deliver ten times more effective data than MTE for the same number of node deaths. There are two reasons that MTE requires more energy to send data to the BS (hence, causing more node deaths for the same amount of data delivery): collisions and lack of data aggregation. Because MTE does not have any centralized control over when nodes transmit and receive packets, collisions increase the amount of energy required to send each successful message. Furthermore, each message in MTE must traverse approximately $0.6\sqrt{N} = 6$ hops to get to the BS,[7] whereas

[7]The analysis for finding the average number of hops is similar to the analysis for finding $E[d^2_{\mathrm{toCH}}]$ in (13).

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.



(a)



(b)

Fig. 8. Data for the limited energy simulations, where each node begins with 2 J of energy. (a) Number of nodes alive over time. (b) Number of nodes alive per amount of data sent to the BS. LEACH can deliver ten times the amount of effective data to the BS as MTE routing for the same number of node deaths. The benefit of rotating cluster heads in LEACH is clearly seen by comparing the number of nodes alive in LEACH and static clustering.

each message in LEACH need only traverse one hop due to data aggregation at the cluster head. Of course, this assumes perfect aggregation—the advantages of using MTE become greater when this assumption is relaxed.

Fig. 8 shows why static clustering performs poorly (as seen in the results from Fig. 7)—the cluster head nodes die quickly, ending the lifetime of all nodes belonging to those clusters. Therefore, rotating the cluster head position enables LEACH to achieve a longer lifetime than static clustering.

V. Discussion

While LEACH appears to be a promising protocol, there are some areas for improvement to make the protocol more widely applicable. In the current implementation of LEACH, we assume sensors always transmit data to the cluster head during their allocated TDMA slot (or, for MTE routing, each $t_{\rm delay}$ seconds). To save energy, nodes may only need to transmit data after they detect some interesting event. In this case, we may need to rethink the intra-cluster communication scheme to make sure that we efficiently utilize bandwidth when not all nodes communicate to the cluster head all the time.

Another assumption we have made is that all nodes are within communication range of each other and the BS. This assumption limits the scalability of the protocol but can be relaxed by using collision-avoidance techniques during the set-up phase to reduce collisions in ADV and Join-REQ messages and using a hierarchical or multihop routing approach to get data from the cluster head nodes to the BS. The cluster heads could form a multihop backbone whereby data are transmitted among cluster heads until they reach the BS. Alternatively, LEACH can evolve into a hierarchical protocol by forming "super clusters" out of the cluster head nodes and having a "super-cluster head" that processes the data from all the cluster head nodes in the super cluster. These changes will make LEACH suitable for a wider range of wireless microsensor networks.

As our results have clearly shown the advantage of rotating the cluster head position among all the nodes, it would be interesting to compare LEACH to a fixed clustering protocol that assigns the cluster head role to another node in the cluster when the current cluster head node dies. Adapting the clusters depending on which nodes are cluster heads for a particular round (as in LEACH) is advantageous because it ensures that nodes communicate with the cluster head node that requires the lowest amount of transmit power. In addition to reducing energy dissipation, this ensures minimum inter-cluster interference. If, on the other hand, the clusters were fixed and only the cluster head nodes were rotated, a node may have to use a large amount of power to communicate with its cluster head when there is another cluster's cluster head close by. Therefore, using fixed clusters and rotating cluster head nodes within the cluster may require more transmit power from the nodes, increasing non-cluster head node energy dissipation and increasing inter-cluster interference. However, the advantage of fixed clusters is that once the clusters are formed, there is no set-up overhead at the beginning of each round. Depending on the cost of forming adaptive clusters, an approach where the clusters are formed once and fixed and the cluster head position rotates among the nodes in the cluster may be more energy efficient than LEACH.

Finally, we showed that using data aggregation reduces energy dissipation and latency in data transfer compared with an approach like MTE that cannot take advantage of local data correlation. However, if there is no correlation among the data (and, hence, the cluster head cannot compress the data from the cluster members), a multihop approach like MTE will outperform LEACH. In Section III-A we discussed an approach to determine the number of nodes N in an approximately $M \times M$ region of space around each node. We can set M appropriately so that there is a high probability that all the sensors within the $M \times M$ area have correlated data. Using the determined value for N, each node can find the approximate optimal k value (19) and compute the appropriate probability that it should become a cluster head during the next round. Using this approach we can assure that, with high probability, the clusters have correlated data and the protocol can scale to a large number of nodes and a large network area, and can handle dynamic nodes.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.

VI. Conclusion

When designing protocol architectures for wireless microsensor networks, it is important to consider the function of the application, the need for ease of deployment, and the severe energy constraints of the nodes. These features led us to design LEACH, a protocol architecture where computation is performed locally to reduce the amount of transmitted data, network configuration and operation is done using local control, and media access control (MAC) and routing protocols enable low-energy networking. Results from our experiments show that LEACH provides the high performance needed under the tight constraints of the wireless channel.

Acknowledgment

The authors would like to thank the reviewers for their detailed comments on earlier versions of this paper.

References

[1] P. Agarwal and C. Procopiuc, "Exact and approximation algorithms for clustering," in *Proc. 9th Annu. ACM-SIAM Symp. Discrete Algorithms*, Baltimore, MD, Jan. 1999, pp. 658–667.
[2] J. Agre and L. Clare, "An integrated architecture for cooperative sensing networks," *IEEE Computer*, vol. 33, pp. 106–108, May 2000.
[3] D. Baker, A. Ephremides, and J. Flynn, "The design and simulation of a mobile radio network with distributed control," *IEEE J. Select. Areas Commun.*, vol. SAC–2, pp. 226–237, Jan. 1984.
[4] A. Chandrakasan, R. Amirtharajah, S.-H. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang, "Design considerations for distributed microsensor systems," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, San Diego, CA, May 1999, pp. 279–286.
[5] L. Clare, G. Pottie, and J. Agre, "Self-organizing distributed sensor networks," in *Proc. SPIE Conf. Unattended Ground Sensor Technologies and Applications*, vol. 3713, Orlando, FL, Apr. 1999, pp. 229–237.
[6] M. Dong, K. Yung, and W. Kaiser, "Low power signal processing architectures for network microsensors," in *Proc. Int. Symp. Low Power Electronics and Design*, Monterey, CA, Aug. 1997, pp. 173–177.
[7] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. 5th Annual ACM Int. Confe. Mobile Computing Networking (MobiCom)*, Seattle, WA, Aug. 1999, pp. 263–270.
[8] M. Ettus, "System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks," in *Proc. Radio and Wireless Conf. (RAWCON)*, Colorado Springs, CO, Aug. 1998, pp. 55–58.
[9] D. Hall, *Mathematical Techniques in Multisensor Data Fusion*. Boston, MA: Artech House, 1992.
[10] W. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. dissertstion, Mass. Inst. Technol., Cambridge, 2000.
[11] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient routing protocols for wireless microsensor networks," in *Proc. 33rd Hawaii Int. Conf. System Sciences (HICSS)*, Maui, HI, Jan. 2000.
[12] L. Hu, "Distributed code assignments for CDMA packet radio networks," *IEEE/ACM Trans. Networking*, vol. 1, pp. 668–677, Dec. 1993.
[13] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. Fourth Annu. ACM Int. Conf. Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000, pp. 56–67.
[14] T. Kwon and M. Gerla, "Clutering with power control," in *Proc. MILCOM*, vol. 2, Atlantic City, NJ, Nov. 1999.
[15] C. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1265–1275, Sept. 1997.
[16] T. Murata and H. Ishibuchi, "Performance evaluation of genetic algorithms for flowshop scheduling problems," *Proc. 1st IEEE Conf. Evolutionary Computation*, vol. 2, pp. 812–817, June 1994.
[17] UCB/LBNL/VINT Network Simulator – ns (2000). [Online]. Available: http://www.isi.edu/vint/ nsnam/
[18] K. Pahlavan and A. Levesque, *Wireless Information Networks*. New York: Wiley, 1995.
[19] S. Park and M. Srivastava, "Power aware routing in sensor networks using dynamic source routing," *ACM MONET Special Issue on Energy Conserving Protocols in Wireless Networks*, 1999.
[20] G. Pottie, "Wireless sensor networks," in *Proc. Information Theory Workshop*, San Diego, CA, June 1998, pp. 139–140.
[21] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, pp. 51–58, May 2000.
[22] T. Rappaport, *Wireless Communications: Principles & Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[23] R. Ruppe, S. Griswald, P. Walsh, and R. Martin, "Near term digital radio (NTDR) system," in *Proc. MILCOM*, vol. 3, Monterey, CA, Nov. 1997, pp. 1282–1287.
[24] T. Shepard, "A channel access scheme for large dense packet radio networks," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1996, pp. 219–230.
[25] S. Singh, M. Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. 4th Annual ACM/IEEE Int. Conf. Mobile Computing Networking (MobiCom)*, Oct. 1998.
[26] A. Wang, W. Heinzelman, and A. Chandrakasan, "Energy-scalable protocols for battery-operated microsensor networks," *Proc. 1999 IEEE Workshop Signal Processing Systems (SiPS '99)*, pp. 483–492, Oct. 1999.

Wendi B. Heinzelman (S'95–M'01) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1997 and 2000, respectively.

Since January 2001, she has been an Assistant Professor in the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY. Her current research interests lie in the area of low-power ad-hoc wireless protocol architectures, scalable sensor networks, and multimedia communication.

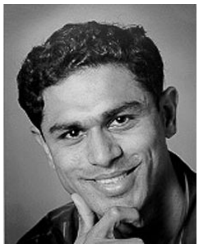
Dr. Heinzelman is an Elected Member of the Design and Implementation of Signal Processing Systems (DISPS) Technical Committee of the IEEE Signal Processing Society and a member of Sigma Xi and the ACM.

Anantha P. Chandrakasan (S'87–M'95–SM'01) received the B.S, M.S. and the Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994, respectively.

Since September 1994, he has been with the Massachusetts Institute of Technology (MIT), Cambridge, and is currently an Associate Professor of electrical engineering and computer science. He held the Analog Devices Career Development Chair from 1994 to 1997. He is a co-author of *Low Power Digital CMOS Design* (Norwell, MA: Kluwer) and a co-editor of *Low Power CMOS Design* and *Design of High-Performance Microprocessor Circuits* (Piscataway, NJ: IEEE Press). His research interests include the ultra low power implementation of custom and programmable digital signal processors, distributed wireless sensors, multimedia devices, emerging technologies, and CAD tools for VLSI.

Dr. Chandrakasan received the NSF Career Development Award in 1995, the IBM Faculty Development Award in 1995, and the National Semiconductor Faculty Development Award in 1996 and 1997, respectively. He has received several best paper awards, including the 1993 IEEE Communications Society's Best Tutorial Paper Award, the IEEE Electron Devices Society's 1997 Paul Rappaport Award for the Best Paper in an EDS publication during 1997, and the 1999 Design Automation Conference Design Contest Award. He has served on the technical program committee of various conferences, including ISSCC, VLSI Circuits Symposium, DAC, and ISLPED. He has served as a Technical Program Co-Chair for the 1997 International Symposium on Low-Power Electronics and Design (ISLPED), VLSI Design '98, and the 1998 IEEE Workshop on Signal Processing Systems, and as a General Co-Chair of the 1998 ISLPED. He was an Associate Editor for the IEEE JOURNAL ON SOLID-STATE CIRCUITS from 1998 to 2001. He served as an Elected Member of the Design and Implementation of Signal Processing Systems (DISPS) Technical Committee of the Signal Processing Society. He was the Signal Processing Subcommittee Chair for ISSCC 1999 through 2001 and the Program Vice-Chair for ISSCC 2002. He is the Technical Program Chair for ISSCC 2003.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.



Hari Balakrishnan (S'95–M'98) received the B.Tech. degree from the Indian Institute of Technology, Madras, in 1993 and the Ph.D. degree in computer science from the University of California, Berkeley, in 1998.

He is an Assistant Professor in the Department of Electrical Engineering and Computer Science and a Member of the Laboratory for Computer Science (LCS) at Massachusetts Institute of Technology (MIT), Cambridge. He leads the Networks and Mobile Systems group at LCS, exploring research issues in wireless networking, mobile systems, network protocols and architecture, and pervasive computing.

Dr. Balakrishnan is the recipient of a National Science Foundation CAREER Award (2000), an IBM University Partnership Prize (1999-2001), the ACM Doctoral Dissertation Award for his work on reliable data transport over wireless networks (1998), and Best Paper Awards at the ACM MOBICOM (1995 and 2000), IEEE HotOS (2001), and USENIX (1995) technical conferences. He holds the KDD Career Development Chair in Communications Technology at MIT, and was awarded the MIT School of Engineering Ruth and Joel Spira Award for Distinguished Teaching in 2001.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on January 21, 2009 at 09:33 from IEEE Xplore. Restrictions apply.