Energy Efficient Chessboard Clustering and Routing in Heterogeneous Sensor Networks

Xiaojiang Du*
Department of Computer Science, North Dakota State University
Fargo, ND 58105, USA
Email: Xiaojiang.Du@ndsu.edu , Tel: 1-701-231-5914, Fax: 1-701-231-8255
*Corresponding author

Yang Xiao
Department of Computer Science, The University of Memphis
Memphis, TN 38152, USA
Email: yangxiao@ieee.org , Tel: 1-901-678-2487, Fax: 1-901-678-2480

Abstract: Sensor nodes are powered by battery and have severe energy constrains. The typical many-to-one traffic pattern causes uneven energy consumption among sensor nodes, i.e., sensor nodes near sink or cluster head have much heavier traffic burden and run out of power much faster than other nodes. The uneven node energy dissipation dramatically reduces sensor network lifetime. In this paper, we propose a novel chessboard clustering scheme to maximize network lifetime by balancing node energy consumption. To achieve good scalability and performance, we propose to form a heterogeneous sensor network by deploying a small number of powerful high-end sensors in addition to a large number of low-end sensors. We also design an efficient routing protocol based on the chessboard clustering scheme. Extensive simulation experiments show that our scheme balances node energy consumption very well and significantly increases network lifetime, and it performs much better than two other clustering schemes – LEACH and LRS.

Keywords: Energy Efficiency, Clustering, Routing, Heterogeneous Sensor Networks.

Biographical notes: Xiaojiang (James) Du is an assistant professor in Dept. of Computer Science, North Dakota State University. He received his MS and Ph.D. in EE from University of Maryland, College Park in 2002 and 2003, respectively, and BS in EE from Tsing-hua University, Beijing, China in 1996. His research interests are wireless sensor networks, mobile ad hoc networks, computer networks, network security and network management. He is a TPC member for several international conferences (including IEEE ICC 2006, Globecom 2005, BroadNets 2005, WirelessCom 2005, IPCCC 2005, BroadWise 2004, etc.). He is a member of IEEE.

Yang Xiao is an assistant professor of Department of Computer Science at The University of Memphis. He is an IEEE Senior member. He was a voting member of IEEE 802.11 Working Group from 2001 to 2004. He currently serves as an associate editor or on editorial boards for six refereed journals including (Wiley) International Journal of Communication Systems, (Wiley) Wireless Communications and Mobile Computing, EURASIP Journal on Wireless Communications and Networking, International Journal of Wireless and Mobile Computing, etc. He serves as a (lead/sole) guest editor for four journal special issues, an editor for four books, and a NSF panelist. His research areas include wireless networks and mobile computing. (E-mail: yangxiao@ieee.org)

1. INTRODUCTION

Recent advances in microprocessor, VLSI, and wireless communication technologies have enabled the deployment of large-scale sensor networks where many low-power, low-cost small sensors are distributed over a vast field to obtain fine-grained, high-precision sensing data. These sensor nodes are typically powered by batteries and communicate through wireless channels, and are usually scattered densely and statically.

Sensor nodes usually operate on a non-replaceable battery. A large proportion of a node's energy resource is consumed in forwarding data [12]. A major design challenge in sensor networks is to increase the operational lifetime of the network as much as possible by employing energy efficient routing. Many routing protocols have been proposed for sensor networks, such as Directed Diffusion [1], TTDD [2], and so on. However, most of the routing protocols did not consider the Uneven Energy Consumption (UEC) problem in sensor networks. In typical sensor networks, the many-to-one traffic pattern is dominant, i.e., a large number of sensor nodes send data to the sink. Thus, sensor nodes near the sink have much heavier traffic burden and run out of power much faster than other nodes. The short lifetime of these critical nodes dramatically reduces sensor network lifetime.

In addition, most existing routing protocols consider homogeneous sensor networks, i.e., all sensor nodes have identical capabilities in terms of communication, computation, sensing, and reliability, etc. However, a homogeneous ad hoc network suffers from poor scalability. Recent research has demonstrated its performance bottleneck both theoretically (Gupta and Kumar [4] showed that the per node throughput in a homogeneous ad hoc network is $\Theta(\frac{1}{\sqrt{n}})$, where n is the number of nodes), and through simulation experiments and testbed measurement [3].

Recently deployed sensor network systems are increasingly following heterogeneous designs, incorporating a mixture of sensors with widely varying capabilities [7]. For example, in a smart home environment, sensors may be powered by AA batteries, AAA batteries or even button batteries. Some recent work starts considering heterogeneous sensor networks. In [6], Mhatre et al. studied the optimum node density and node energies to guarantee a lifetime in heterogeneous sensor networks. In [8], Du presented an energy efficient differentiated coverage algorithm (which can provide different coverage degrees for different areas in a sensor network) for heterogeneous sensor networks. Duarte-Melo and Liu analyzed energy consumption and lifetime of heterogeneous sensor networks in [16].

Clustering-base schemes are promising techniques for sensor networks because of their good scalability and performance. Several clustering-based routing protocols have been proposed for sensor networks, like LEACH [11], TTDD [2], and LRS [14]. LEACH and LRS include redundancy in the system by periodically selecting a cluster-head from the sensors in the network. However, these schemes suffer from overhead of frequent re-clustering. In addition, they did not solve the UEC problem near the sink.

In this paper, we adopt a heterogeneous sensor network model to overcome the performance bottleneck and poor scalability of the homogeneous network model. We address the UEC problem by proposing novel Chessboard Clustering (CC) scheme for the heterogeneous network model. A small number of powerful High-end sensors (H-sensors) are deployed in the field together with a large number of Low-end sensors (L-sensors) to form a heterogeneous sensor network.

Various energy saving protocols have been proposed for sensor networks. Ye et al. proposed PEAS [11] to let redundant sensors go to sleep and save energy. Tian et al. proposed node-sleeping scheme based on "sponsored area" in [9]. Our CC scheme can be used with these protocols together to save sensor energy. The major differences between our CC scheme and other clustering-based routing protocols like LEACH, LRS, and TTDD, are: 1) In CC, physically more powerful H-sensors are the cluster heads, while other protocols need an algorithm to elect cluster heads. 2) In CC, Two different sets of clusters are formed at different time to balance node energy consumption.

2. THE UNEVEN ENERGY CONSUMPTION PROBLEM

In LEACH and LRS, periodically a cluster head is elected from the sensors to solve the uneven energy consumption in cluster heads. However, these schemes suffer from overhead of frequent re-clustering. Further more, rotating cluster-head among sensors does not solve the uneven energy consumption caused by the many-to-one traffic pattern, where the sensors near the sink have much heavier communication burden than other sensors.

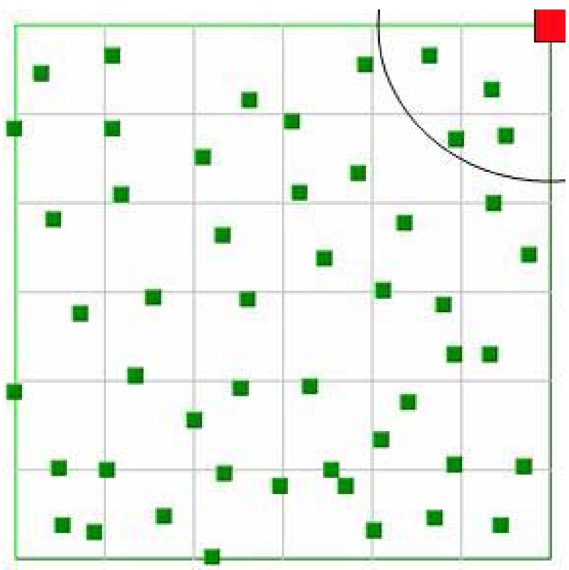

Figure 1 (a): UEC near the Sink

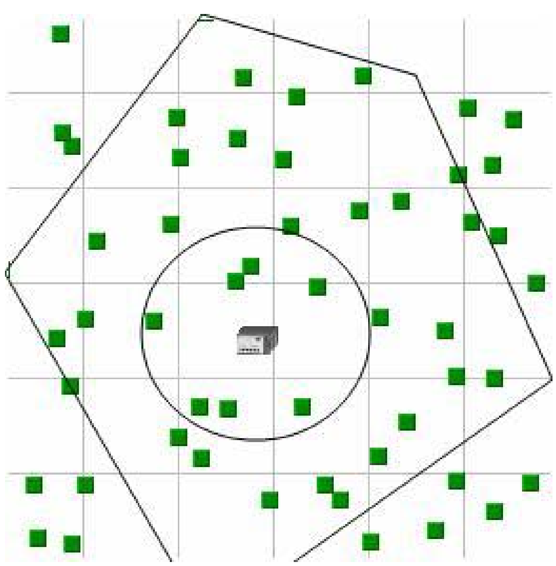

Figure 1 (b): The Critical Nodes in A Cluster

For example, in Figure 1(a), the sink is located in the top-right corner of the field, and all sensors send data packets to the sink via multi-hops. The sensors within one hop to the sink are the critical nodes, and need to relay packets from all other nodes. When all the critical nodes fail, other sensor nodes will be disconnected from the sink, and the sensor network becomes unavailable. The UEC problem exists no matter where the sink is located.

For a heterogeneous sensor network, it is natural to let the more powerful H-sensors become cluster heads. Each L-sensor sends data to its cluster head, and cluster heads forward data to the sink. If H-sensors have sufficient energy supply, the heterogeneous architecture solves the UEC near the sink. Unfortunately, there is another UEC problem in schemes with fixed cluster heads. Consider a typical cluster in Figure 1 (b), where a node has a transmission range of r. The nodes that are within a distance r from the cluster head are referred to as critical nodes. Every transmission of a node in the cluster to the cluster head has to go through one of these critical nodes. This is because the critical nodes are the last hop nodes for all the paths. Hence among all the nodes in a cluster, the critical nodes have the highest burden of relaying data. Since the critical nodes have much heavier traffic load than other nodes in a cluster, they will run out of their power much faster than other nodes. When the critical nodes drain out their energy and become unavailable, other nodes will not be able to send packets to the cluster head, and the entire cluster becomes unavailable even though the remaining energy in many sensor nodes are still high. The remaining energy in the peripheral nodes is wasted.

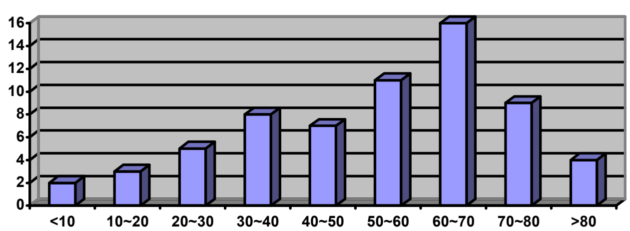

Figure 2: Remaining Energy in Sensor Nodes

We use simulation to demonstrate the uneven energy drain among sensor nodes in a cluster. The results are shown in Figure 2. In the simulation, there are totally 73 sensors in the cluster. The number of sensors that are 1-hop, 2-hop, 3-hop, and 4-hop away from the cluster head is 8, 21, 38 and 6, respectively. Each sensor sends to the cluster head one packet per second. Each node has a fixed amount of energy, and it dies when the energy is run out. The routing protocol used is a greedy geographic routing algorithm [10]. Figure 2 shows the remaining energy in the sensors when all the critical nodes run out of energy, where the *x*-axis is the remaining energy percentage. As we can see from Figure 2, more than half nodes have higher than 50% energy left, and this energy will be wasted in a real sensor network. In fact, the sum of all the remaining energy is equivalent to 38 sensors with full energy. The simulation results show that a significant amount of energy is wasted because the unbalanced energy consumption in sensor networks with fixed cluster heads.

In order to utilize the good scalability and performance in clustering-based schemes, while at the same solve the UEC problem and prolong network lifetime, we propose a novel chessboard clustering scheme for sensor networks. The details are presented in next section.

3. THE CHESSBOARD CLUSTERING ROUTING PROTOCOL

In this section, we present the Chessboard Clustering (CC) routing protocol for heterogeneous sensor networks. We consider a heterogeneous sensor network consisting of two types of nodes: a small number of powerful High-end sensors (H-sensors) and a large number of Low-end sensors (L-sensors). Each sensor node is aware of its own location. Sensor nodes can use location services such as [5, 15] to estimate their locations, and no GPS receiver is required at each node.

First, we briefly present the idea of our chessboard clustering scheme in the following. The sensor network is installed with a chessboard. The sensor network is divided into several small equal-sized cells, and adjacent cells are colored with different colors – white or green, as illustrated in Figure 3. The H-sensors and the L-sensors are assumed to be uniformly and randomly distributed in the field. Given location information, a H-sensor can determine if it is in a white cell or a green cell. During the initialization phase, only the H-sensors in white cells are active, and the H-sensors in green cells turn themselves off. All the L-sensors are active. Clusters are formed around the H-sensors in white cells, and these H-sensors become cluster heads. Later when the H-sensors in white cells run out of energy, the H-sensors in green cells wake up and form a different set of clusters in the network. Because of the formation of a different set of clusters, previous critical sensors become non-critical sensors, and previous non-critical sensors become critical sensors. Since critical sensors consume much more energy than other sensors, this shift balances the energy consumption among sensors, and dramatically prolongs the network lifetime. The detail of the chessboard clustering scheme is presented below.

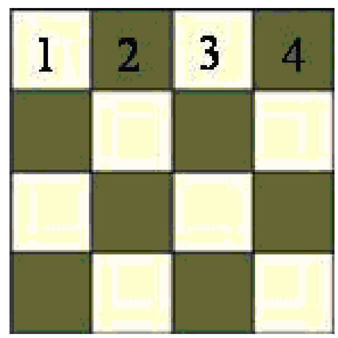

Figure 3: Chessboard Clustering Scheme

3.1 Initial Cluster Formation

During the initialization phase, all H-sensors in white cells broadcast Hello messages to nearby L-sensors with a random delay. The random delay is to avoid the collision of Hello messages from two neighbor H-sensors. The Hello message includes the ID of the H-sensor and its location. The transmission range of the broadcast is large enough so that most L-sensors can receive Hello messages from several H-sensors. Then each L-sensor chooses the H-sensor whose Hello message has the best signal noise ratio (SNR) as the cluster head. Each L-sensor also records other H-sensors from which it receives the Hello messages, and these H-sensors are listed as backup cluster heads in case the primary cluster head fails.

If a L-sensor does not hear any Hello message during the initialization phase (e.g., T seconds after deployment), the node will broadcast an

Explore message. When the neighbor L-sensors receive the Explore message, they will response an Ack message with a random delay. The Ack message includes the location and ID of the sender's cluster head. A L-sensor will not send Ack message again if it hear an Ack response from another neighbor. This mechanism reduces the number of response messages and the consumed energy. Then the L-sensor can select a cluster head based on the Ack message. This ensures all L-sensors have a cluster head. The sensor network is divided into multiple clusters, where each H-sensor in white cells serves as the cluster head. If the network is a two-dimension plane, each L-sensor will select the closest H-sensor as the cluster head (except when there is an obstacle in between), and this leads to the formation of Voronoi cells wherein the cluster heads are the nuclei of the cells. An example of the initial cluster formation is shown in Figure 4(a). The large rectangle nodes in Figure 4(a) are H-sensors, and the small square nodes are L-sensors. The L-sensors within circles are the critical nodes. The H-sensors with a cross are the H-sensors located in green cells, and they are not active until the second half period of the sensor network operation.

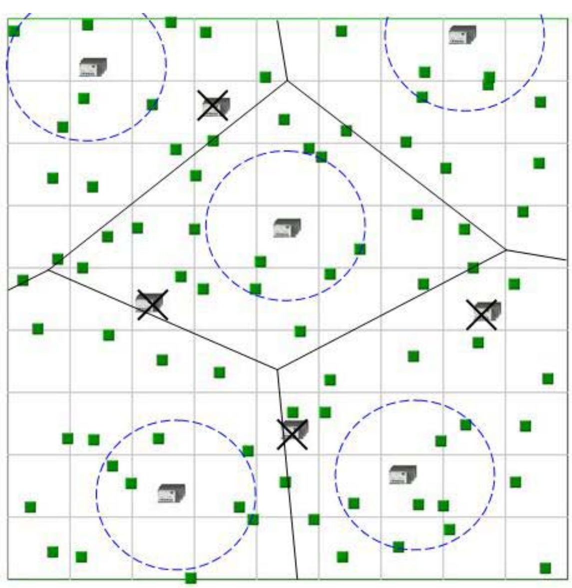

Figure 4 (a): Initial Cluster Formation

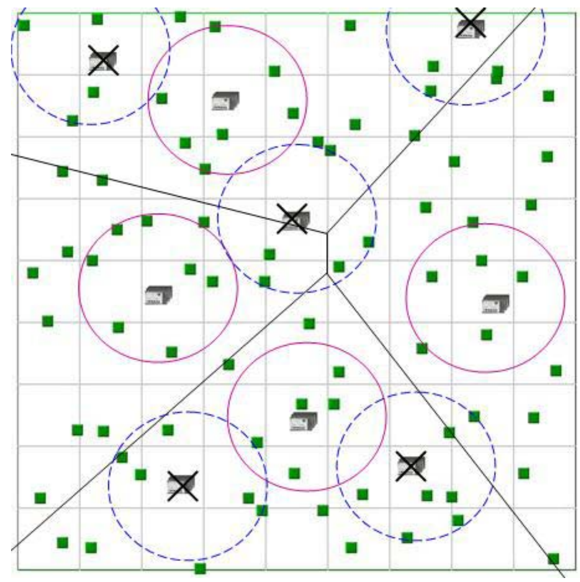

Figure 4 (b): Cluster Re-Formation

3.2. Cluster Re-Formation

For each active H-sensor in white cells, there is a Pairing H-Sensor (PHS) in the neighbor green cell. For example, in Figure 3, the H-sensor in cell 2 (or cell 4) is the PHS for H-sensor in cell 1 (or cell 3). Periodically, the PHS in green cells (say sensor H2) wakes up and sends a query message to the corresponding H-sensor (say sensor H1) in the white cell. Then H1 replies to H2 with its remaining energy included. If the remaining energy of H1 is higher than a threshold (e.g., 1%), H2 will go to sleep again. When the remaining energy of H1 is below the threshold, H2 becomes active and broadcasts Hello messages to nearby L-sensors, and form a cluster around itself. The cluster formation process is the same as the process during initialization. The initial sleeping-time of H-sensors in green cells can be computed based on a conservative estimation of the lifetime of the H-sensors in white cells. The follow-up sleeping-time can be adaptively adjusted by H-sensors in green cells based on the remaining energy of the H-sensor in the white cells. The purpose is to reduce the number of query, since query increases the overhead and consumes energy, while also ensure a PHS becomes active before the H-sensor in white cell dies.

After the sensor network running for certain time, all the H-sensors in white cells drain out of energy and become unavailable. Gradually the H-sensors in green cells become active and form clusters around them, and eventually all the H-sensors in green cells become active and a different set of clusters are formed in the network. A L-sensor can change its cluster head if later it receives a Hello message with better SNR. Figure 4(b) illustrates the cluster re-formation, where H-sensors with cross are the died H-sensors in white cells. In Figure 4(b), the solid circles show the critical nodes in the new clusters. For comparison, the critical nodes in previous clusters are also shown (in dashed circles). As we can see, because of the formation of a different set of clusters, most previous critical nodes become non-critical nodes, and vice verse. In addition, many nodes that were close to the cluster head (like 2 or 3 hops away) are now far away from the cluster head, and vice verse. Since nodes closer to cluster heads have heavier burden than nodes far away, this also causes uneven energy consumption among sensor nodes. The cluster reformation reverses the energy consumption pattern in the network, and thus balances the node energy consumption in the network. In section 3.3 and 3.4, we will present the routing protocol based on Chessboard Clustering scheme.

3.3 Intra-Cluster Routing

Here we discuss the routing scheme inside a cluster. Each L-sensor sends data packets to its cluster head. Since the location of the cluster head is known from the Hello message, a greedy geographic routing protocol can be used for intra-cluster routing. A L-sensor sends the data packet to the neighbor that has the shortest distance to the cluster head, and the next node performs the similar thing, until the data packet reaches the cluster head. Since nodes within a cluster are not far away from the cluster head, the greedy geographic routing should be able to route data packets to cluster head with high probability. The chance of having a void during greedy geographic routing (i.e. all the neighbors have longer distance to the cluster head than the node itself) is small. In case such thing happens, several recover schemes can be used to solve the problem, e.g., GPSR [10] and GOAFR [13] route a packet around the faces of a planar subgraph extracted from the original network.

3.4 Inter-Cluster Routing

Cluster heads know the location of the sink (e.g., from the sink broadcast), and communicate with the sink via multi-hop transmissions over other cluster heads. If enough number of H-sensors (cluster heads) are uniformly and randomly deployed in the network, then with high probability a cluster head can directly communicate with a neighbor cluster head. After cluster formation in the network, each cluster head sends its location information to the sink. Then the sink broadcasts the locations of all cluster heads to each cluster head. When a cluster head wants to send data packets to the sink, it draws a straight line L between itself and the sink. Line L intersects with several Voronoi cells, and these cells are denoted as $C_0, C_1, ..., C_k$, which are referred to as *Relay Cells*. The packet is forwarded from the source cluster head to the sink via the cluster heads in the *Relay Cells*. The Inter-Cluster routing scheme is presented in the following. The cluster head initiating the transmission is referred to as source node R.

1. Based on the location of source and sink, the source node determines the *Relay Cells* $C_0, C_1, ..., C_k$, starting from the cell with node R. R records the *Relay Cells* in a cell_list field, which is stored in the header of the packet. The header contains the following fields: session_id, source_id, sink_id and cell_list. session_id plus source_id uniquely determines a data transmission session.
2. First the data packet is sent from source node R to the cluster head R_1 in cell C_1. Contention-based mechanism is used in MAC layer, e.g., CSMA/CA or IEEE 802.11. A RTS (Request To Send) is broadcast to neighbor cluster heads, and there is a next_cell field in the RTS packet. The next_cell refers to the next cell to relay the data packet. For the RTS from node R, the next_cell is C_1. Based on the next_cell field, only the cluster head in cell C_1 responses to this RTS.
3. Then R sends the data packet to node R_1. After receiving the data packet, R_1 set the next_cell as C_2 according the cell_list field, and proceeds the similar way as above, and sends the

data packet to the cluster head R_2 in cell C_2. To guarantee the delivery, each relay node is responsible for confirming that its successor has successfully received the packet. This may be implemented by the transmitter monitoring the packet just sent out to next node and overhearing if that node has passed it on within a time period. Of course, if link level acknowledgement is supported by the MAC layer protocol (for instance, 802.11 has such function), the above passive acknowledgement scheme is unnecessary. The transmitted data packet has to be kept in the buffer before its receipt has been confirmed. The acknowledgement scheme reduces the impact of channel error.

4. If R_1 does not get any acknowledgement within a time period, R_1 will re-transmit the data packet to R_2 once. And if the retransmission fails, R_1 will find a backup path.

5. A backup path is set up as follows. The current cluster head (say R_1) draws a straight line L between itself and the sink S, and line L intersects with several cells $C_l^{'},...,C_{k-1}^{'},C_k$. If the next cell is the cell with the failed cluster head, R_1 will use a detoured path to avoid the cell. Otherwise, the sequence of new cells $C_l^{'},...,C_{k-1}^{'},C_k$ will be the new *Relay Cells*. And the data packet is forwarded to the sink via the new *Relay Cells*.

6. This process continuous until the data packet reaches the sink.

An example of inter-cluster routing is shown in Figure 5, where cluster head in cell C_0 wants to send data packets to the sink, which is the square in the top-right corner. A straight line from the source cluster head to the sink is used to determine the original *Relay Cells*: cells C_0, C_1, C_2, C_3. If the cluster head in cell C_2 is not available, the cluster head in cell C_1 will use a backup path $C_2^{'}, C_3$ (dotted arrows) to connect the sink.

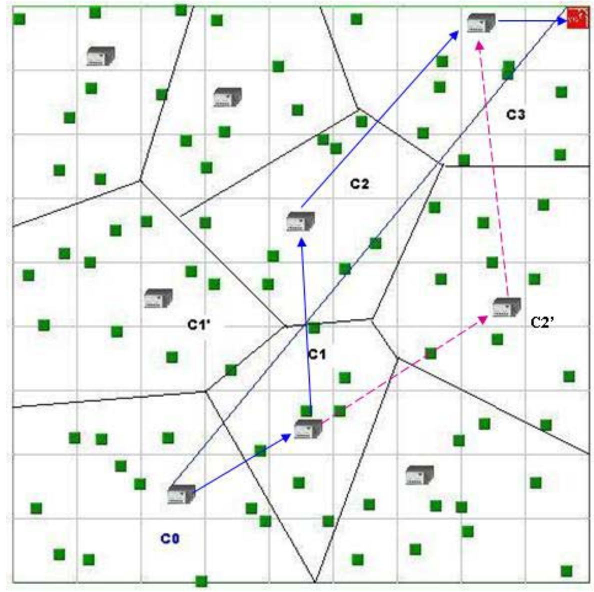

Figure 5: An Example of Inter-Cluster Routing

3.5 More Protocol Details

3.5 1 Robustness to Cluster Head Failure

A cluster head failure can be detected by the L-sensors in the cluster. If a L-sensor *l* does not overhear any transmission from the cluster head after a certain time when it sends a packet to the cluster head, node *l* will assume the cluster head has failed. Recall that each L-sensor records several H-sensors as backup cluster heads during the initialization. Then node *l* will send data packets to one of the backup cluster heads.

3.5.2 Balancing Energy Consumption in Homogeneous Sensor Networks

The unbalanced energy consumption problem also exists in homogeneous sensor networks, as we discussed in section 2 and illustrated in Figure 1 (a). To solve the problem, we propose to change the sink location periodically, among several pre-selected sites. This is possible for many sensor networks. Typically, sensor networks are expected to operate for a long time, e.g., several months or even years. Many sinks are just desktop computers or laptops, and they connect with outside via satellites or air planes. Thus the cost of moving the sink to another location every 2 or

3 months is low. Also the cost for sensors to re-discover route to the sink amortizes over the long period of time, and this cost only incurs once for each sink location.

The positions to install sink are selected such that the energy drain in sensors can be balanced. For example, in Figure 6 we can put the sink in each one of the four locations (1, 2, 3, 4) for a period of time, e.g., put the sink in location 1 for 2 months, and when the sensors near location 1 run out of most power, move the sink to location 2, …, and so on. Then the node energy consumption in the network can be balanced, and the network lifetime can be significantly increased.

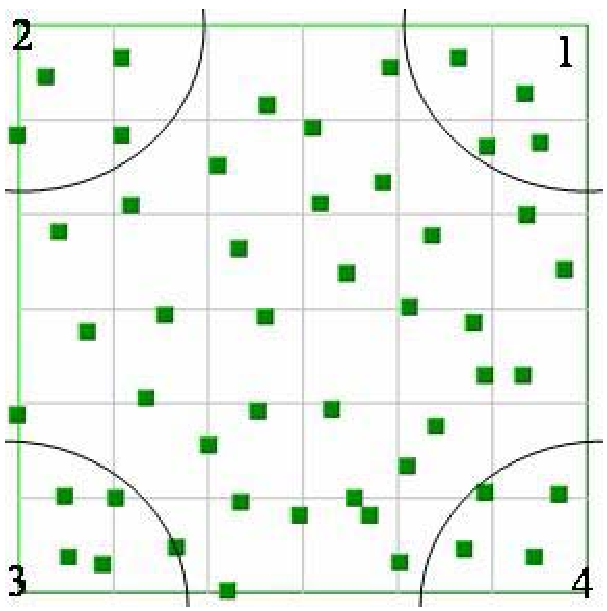

Figure 6: Multiple Sink Locations

Note that it does not solve the unbalanced energy consumption problem by installing multiple sinks in the network without changing their locations. For example, if each of the four locations in Figure 6 has a sink installed, the energy consumption is still not well balanced among nodes. Sensors at each corner become the critical nodes and run out of energy faster than nodes in the middle of the field. The nodes at top-left corner are always critical nodes for the sink in location 1, and so on. On the other hand, if we move the sink from location 1 to location 3, the sensors at (or near) top-left corner will be far away from the sink. More generally, the sensor-to-sink distance changes for many sensors, from close to far away, or vice verse, and this change balances node energy consumption.

4. PERFORMANCE EVALUATION

We evaluate the performance of the Chessboard Clustering (CC) routing protocol through experiments, and compare CC with two other clustering-based routing schemes – LEACH and LRS. LRS is a chain-based 3–level hierarchical protocol proposed by Lindsey, Raghavendra and Sivalingam [14]. In this protocol, sensor nodes are initially grouped into clusters based on their distances from the sink. A chain is formed among the sensor nodes in a cluster at the lowest level of the hierarchy. Gathered data, moves from node to node, gets aggregated, and reaches a designated leader in the chain i.e. the cluster head. At the next level of the hierarchy, the leaders from the previous level are clustered into one or more chains, and the data is collected and aggregated in each chain in a similar manner.

We implemented the Chessboard Clustering routing protocol in QualNet. For comparison, LEACH and LRS were also implemented in QualNet, and the underlying MAC is IEEE 802.11 DCF (RTS in CC has the next_cell field). For CC, the default simulation testbed has 1 sink, 1000 L-sensors and 40 H-sensors randomly, uniformly distributed in a 300*m* x 300*m* area. Since LEACH and LRS are designed for homogeneous sensor networks, for fair comparison, 1500 L-sensors are distributed in the 300*m* x 300*m* area. Here we consider the higher cost of H-sensors compared to L-sensors, and 500 additional L-sensors are used in LEACH and LRS to make the investment similar. Of course the actual costs of H-sensor and L-sensor depend on the type of sensor, manufacture, etc, and this issue is out of the scope of this paper. Here we just want to illustrate the idea.

Each simulation runs for 2000 seconds, and each result is averaged over ten random network topologies. A L-sensor in CC generates 3 data packets per second, and a L-sensor in LEACH (and LRS) generates 2 data packets per second. Thus, the total volume of data generated in CC is the same as in LEACH and LRS. All the data packets have the same length - 32 bytes. The transmission range of a H-sensor and a L-sensor is 80*m* and 20*m* respectively. Both H-sensors and

L-sensors have a fixed amount of energy supply – 10J and 2J respectively. Many high-end sensors have sufficient energy supply, e.g., some sensors have a solar cell to recharge the battery as needed.

Our energy model for the L-sensors is based on the first order radio model described in [11]. A sensor consumes $\epsilon_{elec} = 50$ *nJ/bit* to run the transmitter or receiver circuitry and $\epsilon_{amp} = 100$ $pJ/bit/m^2$ for the transmitter amplifier. Thus, the energy consumed by a L-sensor in receiving a k-bit data packet is given by, $Rx = \epsilon_{elec} \cdot k$, while the energy consumed by sensor *i* in transmitting a data packet to sensor *j* is given by, $Tx = \epsilon_{elec} \cdot k + \epsilon_{amp} \cdot d_{i,j}^2 \cdot k$, where $d_{i,j}$ is the distance between nodes *i* and *j*. The energy parameters for H-sensors are $\epsilon_{elec} = 100$ *nJ/bit* and $\epsilon_{amp} = 200$ $pJ/bit/m^2$.

4.1 Network Lifetime

First we compare the network lifetime for different sensor node density. The network lifetime here is defined as the time that no sensor can send packet to the sink. For the fixed 300*m* x 300*m* routing area, the number of L-sensors in CC varies from 500 to 2000 with an increment of 500, while the number of H-sensors remains 40 for all cases. The numbers of L-sensors in LEACH and LRS are always 1.5 times the number of L-sensors in CC, i.e. varies from 750 to 3000 with an increment of 750. The network lifetimes under the three routing protocols are plotted in Figure 7, where the *x*-axis represents the number of L-sensors in CC.

As we can see, the network lifetimes under all the routing protocols increase as sensor density increases. With higher node density, more sensors are available to forward packet to the sink, and hence the network lifetime increases. Figure 7 also shows that CC has much longer lifetime than both LRS and LEACH. In LRS and LEACH, L-sensors serve as cluster heads in turn to balance node energy consumption and to ensure the availability of cluster heads. However, since L-sensor has limited energy supply, the cluster heads need to re-elected periodically. Even if each L-sensor only serves as cluster head once, there will be 2000 elections in a 2000-node network. Each cluster head election introduces large overhead in the network and drains lots of energy from nearby sensor nodes. Large number of cluster head elections cause sensor nodes to die out quickly. Thus, the network lifetimes in LRS and LEACH are much shorter than CC. In CC, the chessboard clustering scheme balances the energy consumption among different L-sensors very well, avoid causing some nodes being out of energy too soon. In addition, only the more powerful H-sensors serve as cluster heads in CC, and there is only one election for each H-node, which means there are only 40 elections in total. Thus, in CC the overhead from cluster head election is very small. Because of the above two reasons, CC prolongs network lifetime.

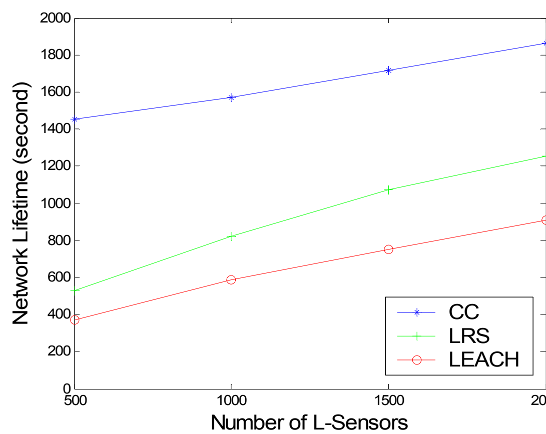

Figure 7: Delivery Ratio Vs Node Density

4.2 Total Energy Consumption

H-sensors have more initial energy than L-sensors, also H-sensors consume more energy than L-sensors for transmitting or receiving one bit data. To fully understand the energy consumption in CC, we measure the total energy consumption in the network, including energy spent by both H-sensors and L-sensors. In the experiments, there are 1000 L-sensors in CC, and 1500 L-sensors in LRS and LEACH. All the measures are taken before 500 second simulation time, during which the network is connected for all the three routing protocols. The results are shown in Figure 8.

As we can see, the total energy consumption in LRS and LEACH are close to each other, and they are much larger than the total energy consumption in CC. In LRS and LEACH, the large number of L-sensors communicate with each other and cause interferences and consume lots of energy, also the frequent re-clustering consumes significant amount of energy. So LRS and LEACH consume more total energy than CC.

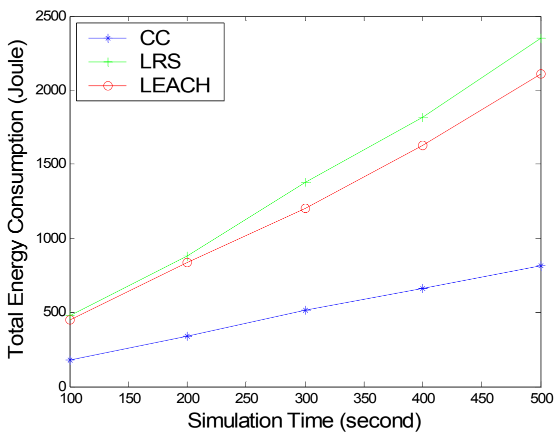

Figure 8: Total Energy Consumption

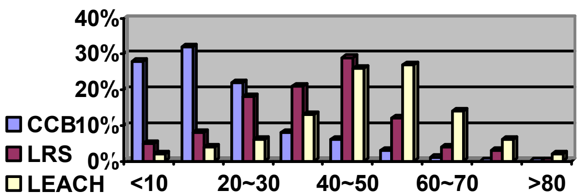

Figure 9: The Distribution of Remaining Node Energy

4.3 Remaining Node Energy

Figure 9 reports the distribution of the remaining node energy when the sensor network became unavailable. The *x*-axis is the remaining energy in terms of the percentage of initial L-sensor energy. We can see that most nodes in CC have remaining energy below 20%, while in LRS most nodes have remaining energy between 20% and 50%, and in LEACH most nodes have 30% to 70% energy left. Figure 9 shows that CC balances the energy consumption among nodes better than both LRS and LEACH, and LRS performs better than LEACH. In typical sensor networks, sensors send packets to sink via multi-hop communications. The failure of any node in the path will cause the route unavailable. If the node energy drain is not balanced well, then some nodes will die too soon and may cause the network disconnected and become unavailable. Besides minimizing the total energy consumption in the network, balancing node energy consumption is also very important for maximizing sensor network lifetime.

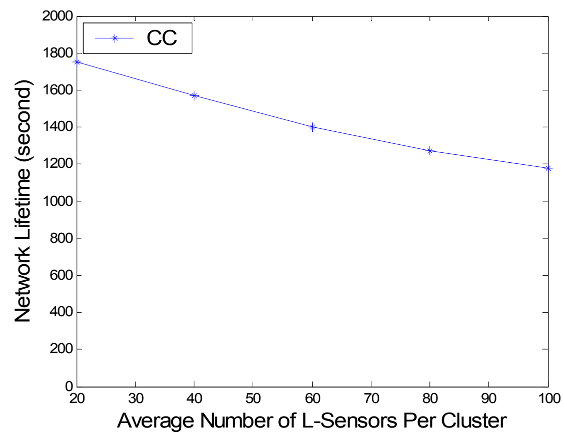

Figure 10: Network Lifetime Vs Cluster Size

4.4 Network Lifetime for Different Cluster Size

We study the effect of cluster size on the performance of CC routing protocol. In the experiments, the number of L-sensors is always 1000, distributed in the 300*m* x 300*m* area. The cluster size is changed by varying the number of H-sensors from 20 to 100, with an increase of 20. Recall that for most of the time, only half of the H-sensors are active in the network (except during the transition of the cluster re-formation). So the size of the cluster varies from average 100 nodes per cluster to average 20 nodes per cluster. The network lifetimes for different cluster sizes are reported in Figure 10. As we can see from Figure 10, the network lifetime decreases as the cluster size becomes larger. Since the total number of L-sensors and the network size do not change, the sensor density does not change. Thus, the number of critical nodes in each cluster is about the same even the cluster size changes. For a larger cluster with more nodes, the relaying burden on each critical node increases, and this causes the critical nodes run out of energy sooner. This is why the network lifetime is shorter for larger clusters. The result suggests that more H-sensors should be deployed in the network to form smaller clusters, and hence increases the network lifetime, given the cost of H-sensors is under budget.

4.5 Increasing Lifetime of Homogeneous Sensor Networks

As we discussed in subsection 3.5, the unbalanced energy consumption problem also exists in homogeneous sensor networks. Sensors near the sink have much heavier traffic burden than sensors far away from the sink. One way to balance the uneven energy consumption is to change the sink location periodically. We study the effect of changing sink location on network lifetime. Three experiments are run where the sink can be moved among 1, 2 and 4 locations of the field (as in Figure 6). The sink locates at one position for 500 seconds, and then moves to another location if multiple locations are used. Since this approach is designed for homogeneous sensor networks, we only run simulations for LEACH and LRS. The results are presented in Figure 11, where 1, 2 and 4 are the number of sink locations, and the y-axis is the network lifetime in seconds. Figure 11 shows that the network lifetime is substantially increased by moving the sink in multiple selected locations, for both LEACH and LRS. The reason is already explained in subsection 3.5.

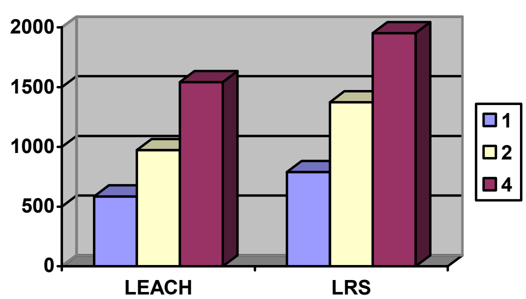

Figure 11: Network Lifetime for Multiple Sink Locations

4.6 Packet Delivery Ratio

To evaluate the effectiveness of CC routing protocol, we measure the packet delivery ratio of CC under different node density, and compare the performance with LEACH and LRS. The packet delivery ratio is defined as the ratio between the number of packets generated in the network and the number of packets received by the sink. In the simulation, the node density change is the same as in subsection 4.1. The number of L-sensor nodes in CC varies from 500 to 2000 with an increment of 500. The number of H-sensor node remains 40 for all cases. The number of L-sensors in LEACH and LRS varies from 750 to 3000 with an increment of 750. The packet delivery ratio is computed with the data from 0 ~ 300 seconds, during which the sensor network is always connected for all routing schemes. The results are plotted in Figure 12, where the x-axis is the number of L-sensors in CC.

Figure 12 shows that CC always achieves very high packet delivery ratio, close to 1 in all tested node density. In CC, L-Sensors send packets to the cluster heads – H-sensors, and H-sensors forward packets to the sink. H-sensors have higher data rate and better processing capability than L-sensors, thus there is few congestion happens when using CC. In LEACH and LRS, cluster head elections occur frequently and cause large overhead. Some sensors die out too early and cause packet lost. When there are more sensors in the network, the effect of certain node failures on packet delivery ratio is reduced. This is the reason that the packet delivery ratio in LEACH and LRS increases as node density increases. In LEACH and LRS, a large number of sensors need to communicate with the sink and this may cause interference and congestion in the network, and thus causes more packets lost. This is another reason that LEACH and LRS have lower packet delivery ratio than CC.

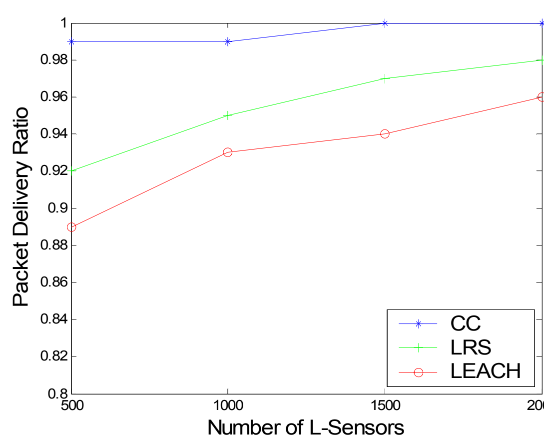

Figure 12: Packet Delivery Ratio Vs Node Density

5. CONCLUSIONS

In this paper, we have presented an energy-efficient Chessboard Clustering (CC) routing protocol for sensor networks, which maximizes network lifetime by balancing node energy consumption. To overcome the performance bottleneck and poor scalability in homogeneous sensor networks, a small number of powerful H-sensors are deployed with L-sensors to form a heterogeneous sensor network. The H-sensors are divided into two groups based on their locations. The two group H-sensors form two different set of clusters during different period of time, and causes the distance between a L-sensor and the cluster head changed. The node energy consumption is well balanced by the chessboard clustering scheme. The CC routing protocol includes intra-cluster and inter-cluster routing schemes. Our simulation experiments show that CC balances node energy consumption and substantially increases network lifetime, and it performs much better than two other clustering-based schemes – LEACH and LRS.

REFERENCES

[1] C. Intanagonwiwat, R. Govindan and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks”, Proc. of *ACM MOBICOM’00*.

[2] F. Ye, H. Luo, J. Cheng, S. Lu and L. Zhang, “A Two tier Data Dissemination Model for Large-scale Wireless Sensor Networks”, Proc. of *ACM MOBICOM’02*.

[3] K. Xu, X. Hong, M. Gerla, “An Ad Hoc Network with Mobile Backbones,” Proc. of *IEEE ICC 2002*.

[4] P. Gupta and P. R. Kumar, “The Capacity of Wireless Networks,” *IEEE Transactions on Information Theory*, vol. IT-46, no. 2, pp. 388-404, Mar. 2000.

[5] L. Doherty, L. El Ghaoui, and K. S. J. Pister, “Convex position estimation in wireless sensor networks,” Proc. of *IEEE Infocom 2001*.

[6] V. Mhatre, et al., “A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint,” accepted for publication in *IEEE Transaction on Mobile Computing,* January 2004.

[7] Lewis Girod, et al., “A System for Simulation, Emulation, and Deployment of Heterogenous Sensor Networks,” Proc. of *ACM SenSys* 2004.

[8] X. Du, “Maintaining Differentiated Coverage in Heterogeneous Sensor Networks”, accepted for publication in *EURASIP Journal on Wireless Communications and Networking,* April 2005.

[9] D. Tian and N.D. Georganas, “A Coverage-preserved Node Scheduling scheme for Large Wireless Sensor Networks,” Proc. of *WSNA'02*.

[10] B. Karp and H. T. Kung, “Gpsr: greedy perimeter stateless routing for wireless networks,” Proc. of *ACM MOBICOM’00*, pages 243-254, 2000.

[11] W. Heinzelman, et al., “Energy-Efficient Communication Protocols for Wireless Microsensor Networks,” Proc. of *Hawaiians Int'l Conf. on Systems Science,* Jan. 2000.

[12] J. Hill, et al., “System Architecture Directions for Network Sensors,” Proc. *of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 93–104, Nov. 2000.

[13] F. Kuhn, R. Wattenhofer, and A. Zollinger, “Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing,” Proc. of *the 4th ACM MobiHoc*, 2003.

[14] S. Lindsey, C. S. Raghavendra, and K. Sivalingam, “Data Gathering in Sensor Networks using the Energy Delay Metric,” Proc. of *the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, 2001.

[15] A. Savvides, C. Han, and M. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” Proc. of *ACM MOBICOM’01*, pg 166–179.

[16] E. Duarte-Melo and M. Liu, “Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks,” Proc. of *IEEE Globecom* 2002.