An Efficient Clustering–based Heuristic for Data Gathering and Aggregation in Sensor Networks[1,2]

Koustuv Dasgupta, Konstantinos Kalpakis, Parag Namjoshi

***Abstract*— The rapid advances in processor, memory, and radio technology have enabled the development of distributed networks of small, inexpensive nodes that are capable of sensing, computation, and wireless communication. Sensor networks of the future are envisioned to revolutionize the paradigm of collecting and processing information in diverse environments. However, the severe energy constraints and limited computing resources of the sensors, present major challenges for such a vision to become a reality.**

We consider a network of energy–constrained sensors that are deployed over a region. Each sensor periodically produces information as it monitors its vicinity. The basic operation in such a network is the systematic gathering and transmission of sensed data to a base station for further processing. During data gathering, sensors have the ability to perform in–network aggregation (fusion) of data packets enroute to the base station. The lifetime of such a sensor system is the time during which we can gather information from all the sensors to the base station. A key challenge in data gathering is to maximize the system lifetime, given the energy constraints.

Given the location of sensors and the base station and the available energy at each sensor, we are interested in finding an efficient manner in which the data should be collected from all the sensors and transmitted to the base station, such that the system lifetime is maximized. This is the maximum lifetime data gathering problem. We present an efficient clustering–based heuristic to solve the data gathering problem. Our experimental results demonstrate that the proposed algorithms significantly outperform previous methods, in terms of system lifetime.

I. INTRODUCTION

The recent advances in micro–sensor technology and low–power analog/digital electronics, have led to the development of distributed, wireless networks of sensor devices ([9], [17], [18]). Sensor networks of the future are envisioned to consist of hundreds of inexpensive nodes, that can be readily deployed in physical environments to collect useful information (e.g. seismic, acoustic, medical and surveillance data) in a robust and autonomous manner. However, there are several obstacles that need to be overcome before this vision becomes a reality [7]. Such obstacles arise from the limited energy, computing capabilities and communication resources available to the sensors.

We consider a system of sensor nodes that are homogeneous and highly energy–constrained. Further, replenishing energy via replacing batteries on hundreds of nodes (in possibly harsh terrains) is infeasible. The basic operation in such a system is the systematic gathering of sensed data to be eventually transmitted to a base station for processing. The key challenge in such data gathering is conserving the sensor energies, so as to maximize their lifetime. To this end, there are several power–aware routing protocols for wireless ad hoc networks discussed in the literature ([12], [19]). In the context of sensor networks, LEACH [6] proposes a clustering–based protocol for transmitting data to the base station. The main features include local co–ordination for cluster formation among sensors, randomized rotation of cluster heads for improved energy utilization, and local data compression to reduce global communication. Chang and Tassiulas ([2], [3]) describe data routing algorithms that maximize the time until the energies of the sensors drain out. In related work, Bhardwaj et al [1] derive upper bounds on the lifetime of a sensor network that collects data from a specified region using some energy–constrained nodes.

Data fusion or aggregation has emerged as a basic tenet in sensor networks. The key idea is to combine data from different sensors to eliminate redundant transmissions, and provide a rich, multi–dimensional view of the environment being monitored. Krishnamachari et al [11] argue that this paradigm shifts the focus from address–centric approaches (finding routes between pairs of end nodes) to a more data–centric approach (finding routes from multiple sources to a destination that allows in–network consolidation of data). Madden et al [16] describe the TinyOS operating system that can be used by an ad-hoc network of sensors to locate each other and route data. The authors discuss the implementation of five basic database aggregates, i.e. COUNT, MIN, MAX, SUM, and AVERAGE, based on the TinyOS platform and demonstrate that such a generic approach for aggregation leads to significant power (energy) savings. The focus of the work in [16] is on a class of aggregation predicates that is particularly well suited to the in-network regime. Such aggregates can be expressed as an aggregate function f over the sets a and b, such that $f(a \cup b) = g(f(a), f(b))$. Other previous works ([8], [7], [13], [14]) in the related area aim at reducing the energy expended by the sensors during the process of data gathering. Directed diffusion [8] is based on a network of nodes that can co–ordinate to perform distributed sensing of an environmental phenomenon. Such an approach achieves significant energy savings when intermediate nodes aggregate responses to queries. The SPIN protocol [7] uses meta–data negotiations between sensors to eliminate redundant data transmissions through the network. In PEGASIS [13], sensors form chains so that each node transmits and receives from a nearby neighbor. Gathered data moves from node to

[0]Supported in part by NASA under Cooperative Agreement NCC5–315 and Contract NAS5-32337, and by NSF under grant IRI–9729495.

[1]Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250. E-mail: {kalpakis,dasgupta,nam1}@csee.umbc.edu. Phone: 410-455-3143. Fax: 410-455-3969.

Please send all correspondence to Prof. Kalpakis.

node, gets aggregated and is eventually transmitted to the base station. Nodes take turns to transmit so that the average energy spent by each node gets reduced. Lindsey et al [14] describe a hierarchical scheme based on PEGASIS that reduces the average energy consumed and delay incurred in gathering the sensed data.

In a recent paper [10], we proposed a polynomial–time near–optimal *Maximum Lifetime Data Aggregation* (MLDA) algorithm for data gathering in sensor networks. The proposed method, while performing significantly better than existing protocols in terms of system lifetime, is computationally expensive for large sensor networks. In this paper, we present a simple and efficient clustering–based heuristic for maximum lifetime data aggregation (CMLDA) in large–scale sensor networks. Further, we provide experimental results to show that (i) for smaller sensor networks the CMLDA heuristic achieves system lifetimes that are within 10% of the optimal and 1.6 to 4.5 times better when compared to an existing data gathering protocol, (ii) for larger networks, the CMLDA heuristic can achieve as much as a factor of 5.8 increase in system lifetime when compared to the same protocol.

The rest of the paper is organized as follows. We formulate the data gathering problem in section II and briefly describe the MLDA algorithm to solve the maximum lifetime data aggregation problem in section III. Next, in section IV, we propose a clustering–based heuristic to efficiently for large sensor networks. In section V, we present experimental results based on our algorithms. Finally, in section VI we conclude the paper.

II. The Data Gathering Problem

A. System Model

Consider a network of n sensor nodes $1, 2, \ldots, n$ and a base station node t labeled $n+1$ distributed over a region. The locations of the sensors and the base station are fixed and known apriori. Each sensor produces some information as it monitors its vicinity. We assume that each sensor generates one data packet per time unit to be transmitted to the base station. For simplicity, we refer to each time unit as a *round*. We assume that all data packets have size k bits. The information from all the sensors needs to be gathered at each round and sent to the base station for processing. We assume that each sensor has the ability to transmit its packet to any other sensor in the network or directly to the base station. Further, each sensor i has a battery with finite, non-replenishable energy $\mathcal{E}_i$. Whenever a sensor transmits or receives a data packet, it consumes some energy from its battery. The base station has an unlimited amount of energy available to it.

Our energy model for the sensors is based on the first order radio model described in [6]. A sensor consumes $\epsilon_{elec} = 50nJ/bit$ to run the transmitter or receiver circuitry and $\epsilon_{amp} = 100pJ/bit/m^2$ for the transmitter amplifier. Thus, the energy consumed by a sensor i in receiving a k-bit data packet is given by,

$$R\text{x}_i = \epsilon_{elec} \cdot k \tag{1}$$

while the energy consumed in transmitting a data packet to sensor j is given by,

$$T\text{x}_{i,j} = \epsilon_{elec} \cdot k + \epsilon_{amp} \cdot d_{i,j}^2 \cdot k \tag{2}$$

where $d_{i,j}$ is the distance between nodes i and j.

B. Problem Statement

We define the *lifetime* T of the system to be the number of rounds until the first sensor is drained of its energy. A *data gathering schedule* specifies, for each round, how the data packets from all the sensors are collected and transmitted to the base station. For brevity, we refer to a data gathering schedule simply as a schedule. Observe that a schedule can be thought of as a collection of T directed trees, each rooted at the base station and spanning all the sensors i.e. a schedule has one tree for each round. The lifetime of a schedule equals the lifetime of the system under that schedule. Clearly, the system lifetime is intrinsically connected to the data gathering schedule. Our objective is to find a schedule that maximizes the system lifetime T.

III. MLDA : Maximum Lifetime Data Gathering with Aggregation

Data aggregation performs in-network fusion of data packets, coming from different sensors enroute to the base station, in an attempt to minimize the number and size of data transmissions and thus save sensor energies [8], [6], [11], [13]. Such aggregation can be performed when the data from different sensors are highly correlated. As in previous work [8], [6], [11], [13], we make the simplistic assumption that an intermediate sensor can aggregate multiple incoming packets into a single outgoing packet.

Maximum Lifetime Data Aggregation (MLDA) problem: Given a collection of sensors and a base station, together with their locations and the energy of each sensor, find a data gathering schedule with maximum lifetime, where sensors are permitted to aggregate incoming data packets.

We proposed a polynomial–time algorithm to solve the MLDA problem in [10]. For the sake of completeness, we next provide a brief description of our algorithm.

Consider a schedule $\mathcal{S}$ with lifetime T rounds. Let $f_{i,j}$ be the total number of packets that node i (a sensor) transmits to node j (a sensor or base station) in S. Since any valid schedule must respect the energy constraints at each sensor, it follows that for each sensor $i = 1, 2, \ldots, n$,

$$\sum_{j=1}^{n+1} f_{i,j} \cdot T\text{x}_{i,j} + \sum_{j=1}^{n} f_{j,i} \cdot R\text{x}_i \leq \mathcal{E}_i. \tag{3}$$

Recall that each sensor, for each one of the T rounds, generates one data packet that needs to be collected, possibly aggregated, and eventually transmitted to the base station.

The schedule $\mathcal{S}$ induces a flow network $G = (V, E)$. The flow network G is a directed graph having as nodes all the sensors and the base station, and having edges (i, j) with capacity $f_{i,j}$ whenever $f_{i,j} > 0$.

Objective :
maximize T (4)

Constraints :
$\sum_{j=1}^{n+1} f_{i,j} \cdot T\mathrm{x}_{i,j} + \sum_{j=1}^{n} f_{j,i} \cdot R\mathrm{x}_i \leq \mathcal{E}_i$.(5)
$\sum_{j=1}^{n} \pi_{j,i}^{(k)} = \sum_{j=1}^{n+1} \pi_{i,j}^{(k)}, \quad \forall i = 1, 2, \ldots, n$ and $i \neq k$(6)
T + $\sum_{j=1}^{n} \pi_{j,k}^{(k)} = \sum_{j=1}^{n+1} \pi_{k,j}^{(k)}$(7)
$0 \leq \pi_{i,j}^{(k)} \leq f_{i,j}, \quad \forall i = 1, 2, \ldots, n$ and $\forall j = 1, 2, \ldots, n+1$(8)
$\sum_{i=1}^{n} \pi_{i,n+1}^{(k)} = T$(9)
where $k = 1, 2, \ldots, n$ and all variables T, $f_{i,j}$, and $\pi_{i,j}^{(k)}$ are required to be non-negative integers.

TABLE I

INTEGER PROGRAM FOR FINDING AN OPTIMAL ADMISSIBLE FLOW NETWORK FOR THE MLDA PROBLEM.

Theorem 1: Let $\mathcal{S}$ be a schedule with lifetime T, and let G be the flow network induced by $\mathcal{S}$. Then, for each sensor s, the maximum flow from s to the base station t in G is $\geq T$.

Proof: Each data packet transmitted from a sensor must reach the base station. Observe that, the packets from s could possibly be aggregated with one or more packets from other sensors in the network. Intuitively, we need to guarantee that each of the T values from s *influences* the final value(s) received at the base station. In terms of network flows, this implies that sensor s must have a maximum $s-t$ flow of size $\geq T$ to the base station in the flow network G. ■

Thus, a necessary condition for a schedule to have lifetime T is that each node in the induced flow network can push flow T to the base station t. Stated otherwise, each sensor s must have a minimum $s-t$ cut of capacity (size) $\geq T$ to the base station [4]. Next, we consider the problem of finding a flow network G with maximum T, that allows each sensor to push flow T to the base station, while respecting the energy constraints in (5) at all the sensors. We call such a flow network G an *admissible* flow network with lifetime T. An admissible flow network with maximum lifetime is called an *optimal admissible* flow network. Clearly, what needs to be found are the capacities of the edges in G.

A. Finding a near–optimal admissible flow network

An optimal admissible flow network can be found using an integer program with linear constraints. The integer program, in addition to the variables for the lifetime T and the edge capacities $f_{i,j}$, uses the following variables: for each sensor $k = 1, 2, \ldots, n$, let $\pi_{i,j}^{(k)}$ be a flow variable indicating the flow that k sends to the base station t over the edge (i, j).

The integer program computes the maximum system lifetime T subject to the energy constraint (5) and the additional linear constraints (6)–(9) for each sensor, as shown in Table I. For each sensor $k = 1, 2, \ldots, n$, constraints (6) and (7) enforce the flow conservation principle at the sensor; constraint (9) ensures that T flow from sensor k reaches the base station; and constraint (8) ensures that the capacity constraints on the edges of the flow network are respected. Moreover, constraint (5) is used to guarantee that the edge capacities of the flow network respect the sensor's available energy. Finally, for the integer program, all variables are required to take non–negative integer values. The linear relaxation of the above integer program, i.e. when all the variables are allowed to take fractional values, can be computed in polynomial–time. Then, we can obtain a very good approximation for the optimal admissible flow network by first fixing the edge capacities to the floor of their values obtained from the linear relaxation so that the energy constrains are all satisfied; and then solving the linear program (4) subject to constraints (6)–(9) without requiring anymore that the flows are integers (since a solution with integer flows can always be found). [2]

B. Constructing a schedule from an admissible flow network

Next, we discuss how to get a schedule from an admissible flow network. Recall that a schedule is a collection of directed trees rooted at the base station that span all the sensors, with one such tree for each round. Each such tree specifies how data packets are gathered and transmitted to the base station. We call these trees *aggregation trees*. An aggregation tree may be used for one or more rounds; we indicate the number of rounds f, that an aggregation tree is used, by associating the value f with each one of its edges; we call f to be the lifetime of the aggregation tree.

Figure 1 provides an example of an admissible flow network G with lifetime $T = 100$ and two aggregation trees A_1 and A_2, with lifetimes 60 and 40 respectively. By looking at one of these trees, say A_1, we see that for each one of 60 rounds, sensors 2 and 3 transmit one data packet to sensor 1, which in turn aggregates the incoming packets with its own data packet, and then sends one data packet to the base station. Similarly, for each of the remaining 40 rounds (using A_2), sensors 1 and 2 transmit one data packet to sensor 3, which in aggregates the incoming packets with its own packet, and sends one data packet to the base station. We next describe an algorithm to construct aggregation trees from an admissible flow network G with lifetime T.

Definition 1 : Given an admissible flow network G with lifetime T and a directed tree A rooted at the base station t with lifetime f, we define the (A, f)*–reduction* G' *of* G to

[2]The reduction in the system lifetime achieved, w.r.t the fractional optimal lifetime, is at most the maximum cardinality of any min s–t cut.

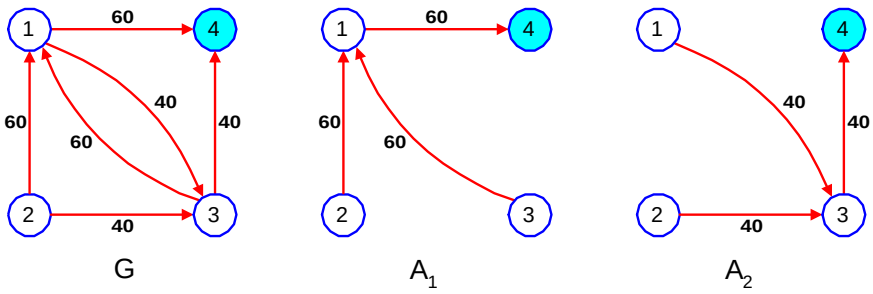

Fig. 1. An admissible flow network G with lifetime 100 rounds, and two aggregation trees A_1 and A_2 with lifetimes 60 and 40 rounds respectively.

be the flow network that results from G after reducing by f, the capacities of all of its edges that are also in A. We call G' the (A, f)–*reduced* G.

Definition 2 : An (A, f)–reduction G' of G is *feasible* if the maximum flow from v to the base station t in G' is $\geq T - f$ for each vertex v in G'.

Note that A does not have to span all the vertices of G, and thus it is not necessarily an aggregation tree. Moreover, if A is an aggregation tree, with lifetime f, for an admissible flow network G with lifetime T, and the (A, f)–reduction of G is feasible, then the (A, f)–reduced flow network G' of G is an admissible flow network with lifetime $T - f$. Therefore, we can devise a simple iterative procedure, to construct a schedule for an admissible flow network G with lifetime T, provided we can find such an aggregation tree A.

We use the GETTREE algorithm in Figure 2 to get an aggregation tree A with lifetime $f \leq T$ from an admissible flow network G with lifetime T. Throughout this algorithm, we maintain the invariant that A is a tree rooted at t and the (A, f)–reduction of G is feasible. Tree A is formed as follows. Initially A contains just the base station. While A does not span all the sensors, we find and add to A an edge $e = (i, j)$, where $i \notin A$ and $j \in A$, provided that the (A', f)–reduction of G is feasible–here A' is the tree A together with the edge e and f is the minimum of the capacities of the edges in A'. The running time of this algorithm is polynomial in the number of sensors. Finally, we can compute a collection of aggregation trees from an admissible flow network G with lifetime T by repeatedly invoking the GETTREE algorithm, until all T data packets from each of the sensors have been aggregated and transmitted to the base station t. Given a flow network G and base station t such that each sensor s has a minimum $s-t$ cut of size $\geq T$ (i.e. the maximum flow from s to t in G is $\geq T$), we can prove that it is always possible to find a collection of aggregation trees, via the GETTREE algorithm, which can be used to aggregate T data packets from each of the sensors. The proof of correctness is based on a powerful theorem in graph theory (Edmonds[5], Lovász[15]) and is omitted due to lack of space. We refer to the algorithm described in this section, for finding a maximum lifetime schedule with data aggregation, as the MLDA algorithm.

IV. CMLDA : CLUSTERING–BASED MLDA HEURISTIC

Given the location of n sensors and a base station t, we can find a (near–optimal) maximum lifetime data gathering schedule using the MLDA algorithm. However, it involves solving a linear program (in Table I) with $O(n^3)$ variables and constraints. For large sensor networks, i.e. for large values of n, this can be computationally expensive[3]. In order to solve the data gathering problem efficiently for large networks, we next describe a heuristic based on the MLDA algorithm.

Consider the set of n sensors $1, 2, \ldots, n$ and a base station t labeled as $n + 1$. Let the sensors be partitioned into m clusters $\phi_1, \ldots, \phi_m$ each consisting of at most c sensors, i.e. $|\phi_i| \leq c$, for $i = 1, 2, \ldots, m$ and an appropriate constant c. We refer to each cluster as a *super–sensor*. Such a partitioning of the sensors can be achieved using a proximity–based clustering algorithm. Our approach is to compute a maximum lifetime schedule for the super–sensors $\phi_1, \ldots, \phi_m$ with the base station ϕ_{m+1}, and then use this schedule to construct aggregation trees for the sensors. Figure 5 gives a high level view of the cluster–based MLDA (CMLDA) heuristic. In the first phase, we assign the initial energy of each super–sensor ϕ_i $(i = 1, 2, \ldots, m)$ to be the sum of the initial energies of the sensors within it, i.e. $\mathcal{E}_{\phi_i} = \mathcal{E} \cdot |\phi_i|$. The distance between two super–sensors ϕ_i and ϕ_j is assigned to be the maximum distance between any two nodes (sensor or base station) u and v, such that $u \in \phi_i$ and $v \in \phi_j$. Having set up the initial energies and the distances between the super–sensors, we can find a maximum lifetime schedule for the super–sensors $\phi_1, \ldots, \phi_m$ with the base station as ϕ_{m+1}, using the MLDA algorithm. Recall that such a schedule consists of a collection of directed trees $\mathcal{T}_1, \ldots, \mathcal{T}_k$, each rooted at ϕ_{m+1} and spanning over all the super–sensors. To distinguish it from an aggregation tree for the sensors, we refer to each such tree as an aggregation super–tree (or simply an *AS–tree*). Next, we use the BUILD–TREE procedure (in figure 4) to construct an aggregation tree A for the sensors from an AS–tree $\mathcal{T}_k$. Observe that A is a directed tree rooted at t that is used to aggregate one data packet from each sensor. We denote $\mathcal{E}^r[i]$ to be the residual energy at sensor i. Initially, $\mathcal{E}^r[i] = \mathcal{E}$ for each sensor i in the network. Our objective is to construct (one or more) aggregation trees such that the *minimum residual energy* among the n sensors is maximized, thereby maximizing the lifetime of the corresponding data gathering schedule.

Initially, aggregation tree A contains only the base station t. We perform a (pre–order) traversal of the AS–tree $\mathcal{T}_k$. For each visited super–sensor ϕ, we add the sensors in ϕ to the

[3]For example, it takes approximately 60 seconds to solve a problem instance with 20 sensors and up to 5 hours for 100 sensors.

GETTREE (Flow Network G, Lifetime T, Base Station t)

```
1   initialize f ← 1
2   let A = (V_o, E_o) where V_o = {t} and E_o = ∅
3   while A does not span all the nodes of G do
4       for each edge e = (i, j) ∈ G such that i ∉ V_o and j ∈ V_o do
5           let A' be A together with the edge e
6           // check if the (A', 1)–reduction of G is feasible
7           let G_r be the (A', 1)–reduction of G
8           if MAXFLOW(v, t, G_r) ≥ T − 1 for all nodes v of G
9               // replace A with A'
10              V_o ← V_o ∪ {i}, E_o ← E_o ∪ {e}
11              break
12  let c_min be the minimum capacity of the edges in A
13  let G_r be the (A, c_min)–reduction of G
14  if MAXFLOW(v, t, G_r) ≥ T − c_min for all nodes v of G
15      f ← c_min
16  replace G with the (A, f)–reduction of G
17  return f, G, A
```

Fig. 2. Constructing aggregation tree A with lifetime f from an admissible flow network G with lifetime T.

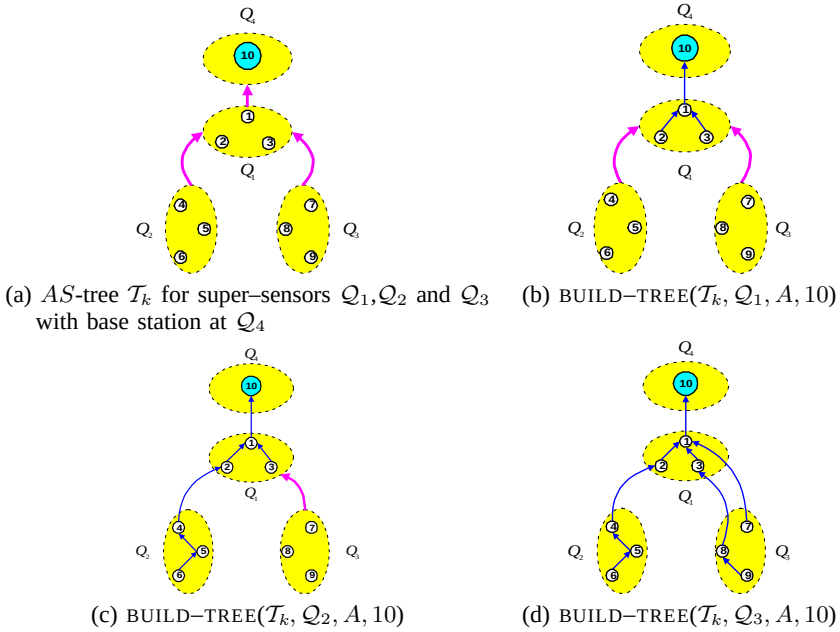

Fig. 3. Illustration of the BUILD–TREE procedure for an AS–tree $\mathcal{T}_k$. Sensors $1, 2, \ldots, 9$ are partitioned among super–sensors $\mathcal{Q}_1$, $\mathcal{Q}_2$, and $\mathcal{Q}_3$. Super–sensor $\mathcal{Q}_4$ contains the base station 10. The aggregation tree A is used for collecting one data packet from each sensor and transmitting to the base station.

current aggregation tree A. Let $\phi - A$ denote the set of sensors in ϕ that are not included in A. We define the *residual energy* of a pair (i, j) as $\min\{\mathcal{E}^r[i] - T\mathrm{x}_{i,j},\ \mathcal{E}^r[j] - R\mathrm{x}_j\}$, where $i \in \phi - A$ and $j \in A$. Intuitively, on adding a directed edge (i, j) to A, the residual energy at sensor i is reduced by the energy consumed in transmitting a data packet from i to j. Moreover, if j is not the base station, its residual energy is reduced by the energy consumed in receiving a data packet. Among all pairs (i, j), such that $i \in \phi - A$ and $j \in A$, the BUILD–TREE procedure chooses one with the maximum residual energy and includes the edge (i, j) in A. The process is repeated until all sensors in ϕ are included in A, upon which it continues with the next super–sensor in $\mathcal{T}_k$. Figure 4 gives an illustration of the BUILD-TREE procedure. The running time of the procedure is $O(n^3)$, where n is the number of sensors in the network. Finally, observe that a maximum lifetime schedule for the super–sensors could possibly consist of one or more AS-trees. In this case, we choose (in step 9 of figure 5) the AS–trees in decreasing order of their respective lifetimes; while constructing no more than f_k aggregation trees from a particular AS-tree $\mathcal{T}_k$, where f_k is the lifetime of $\mathcal{T}_k$.

V. EXPERIMENTS

In this section, we compare the data gathering schedule given by the CMLDA heuristic with that obtained from a chain-based 3–level hierarchical protocol proposed by Lindsey, Raghavendra and Sivalingam [14]. For brevity, we refer to this protocol as the LRS protocol. We choose this protocol since

BUILD–TREE(AS–Tree $\mathcal{T}_k$, Super–Sensor ϕ, Aggregation Tree A, Base Station t)

```
1   while A does not contain all the sensors in φ do
2       find a pair (i, j), where i ∈ φ − A and j ∈ A, with maximum residual energy
3       add the edge (i, j) to A
4       update the residual energy of sensor i as E^r[i] ← E^r[i] − T·Tx_{i,j}
5       if j ≠ t then update the residual energy of j as E^r[j] ← E^r[j] − R·Rx_j
6   foreach child φ' of φ in T_k do
7       A ← BUILD–TREE(T_k, φ', A, t)
8   return A
```

Fig. 4. Constructing an aggregation tree A for the sensors from an AS–tree $\mathcal{T}$.

INPUT:
Location of n sensors $1, 2, \ldots, n$ and a base station t.
Initial energy $\mathcal{E}$ in each sensor.

OUTPUT:
A data gathering schedule $\mathcal{S}$, i.e. a collection of aggregation trees, with lifetime T.

ALGORITHM:

PHASE I:

1. partition the sensors into m super–sensors $\phi_1, \ldots, \phi_m$
2. let super–sensor ϕ_{m+1} consist only of the base station t
3. let the energy of each super–sensor ϕ_i, $i = 1, 2, \ldots, m$, be $\mathcal{E}_{\phi_i} \leftarrow \mathcal{E} \cdot |\phi_i|$
4. let the distance between any two super–sensors ϕ_i and ϕ_j be $d_{\phi_i,\phi_j} \leftarrow \max\{d_{u,v} \ : \ u \in \phi_i, v \in \phi_j\}$
5. find an admissible flow network G for the super–sensors $\phi_1, \ldots, \phi_m$ with base station ϕ_{m+1}, and compute a schedule $\mathcal{T} \leftarrow \{\mathcal{T}_i, \ldots, \mathcal{T}_k\}$ from G

PHASE II:

6. initialize the schedule $\mathcal{S} \leftarrow \emptyset$ and lifetime $T \leftarrow 0$
7. let the residual energy of each sensor $i = 1, 2, \ldots, n$ be $\mathcal{E}^r[i] = \mathcal{E}$
8. while $\min\{\mathcal{E}^r[i] \ : \ i = 1, 2, \ldots, n\} > 0$ do
9. choose an AS–tree $\mathcal{T}_k$ from $\mathcal{T}$
10. initialize A to contain only the base station t
11. compute an aggregation tree $A \leftarrow$ BUILD–TREE$(\mathcal{T}_k, \phi_{m+1}, A, t)$
12. update the schedule $\mathcal{S} \leftarrow \mathcal{S} \cup A$ and lifetime $T \leftarrow T + 1$

Fig. 5. A high level description of the CMLDA heuristic.

it significantly outperforms other competitive protocols (e.g. LEACH [6]) in terms of system lifetime.

LRS protocol for constructing a data gathering schedule: In this protocol, sensor nodes are initially grouped into clusters based on their distances from the base station. 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. Thus, for gathering data in each round, each sensor transmits to a close neighbor in a given level of the hierarchy. This occurs at every level, the only difference being that nodes that are receiving at each level are the only nodes that rise to the next level in the hierarchy. Finally at the top level, there is a single leader node transmitting to the base station. To increase the lifetime of the system, the leader in each chain is chosen in a round-robin manner in each round. Observe that, the protocol naturally defines aggregation tree(s) for each round of data gathering.

For the initial set of experimental results, we consider a network of sensors randomly distributed in a 50m $\times$ 50m field. The number of sensors in the network, i.e. the network size n, is varied to be 40, 50, 60, 80 and 100 respectively. Each sensor has an initial energy of 1J and the base station is located at (25m, 150m). Each sensor generates packets of size 1000 bits. The energy model for the sensors is based on the first order radio model described in section II.

Each experiment corresponds to a random placement of the sensors, for a particular network size. In each experiment, we measure the lifetime T, i.e. the number of rounds before the first sensor is drained of its energy, for the data gathering schedule given by the LRS protocol. For the same placement of sensors, we measure the lifetime of the data gathering schedules obtained from MLDA and CMLDA. We define the *performance ratio* R as the ratio of the system lifetime achieved using CMLDA to the lifetime given by the LRS

protocol. Recall that, the (integral) solution given by MLDA is an approximation of the optimal fractional solution. We denote OPT to be the optimal system lifetime for any particular experiment.

For a data gathering schedule $\mathcal{S}$, we define the *depth* of a sensor v to be its average number of hops from the base station in the schedule, i.e. the average of its depths in each of the aggregation trees in $\mathcal{S}$. The depth of the schedule is defined as $\max\{\text{depth}(i)\}$, among all sensors i in the network. We measure the depth D of a schedule constructed using each of the MLDA, CMLDA and LRS algorithms. Note that, the depth of a data gathering schedule is an interesting metric since it gives an estimate of the (maximum) average delay that is incurred in sending data packets from any sensor to the base station.

Finally, for the CMLDA heuristic, we denote c to be the number of sensors in a cluster (super–sensor). Given the location of the sensors and the base station, we employ a greedy clustering algorithm similar to the the chain–forming algorithm used by the LRS protocol [14] – pick a sensor i farthest from the base station and form a cluster that includes i and its $c-1$ nearest neighbors; continue the process with the remaining sensors until all sensors have been included in some cluster. For a particular network size, we assign the size of a cluster in CMLDA to be identical to the size of a chain in the LRS protocol. By clustering the sensors in the above manner, we can efficiently compute a maximum lifetime schedule for the super–sensors (via the MLDA algorithm) even for large problem instances. Observe that, the MLDA algorithm and the CMLDA heuristic presented in this paper are essentially centralized in nature. This implies that the clustering of the sensors need to be pre–computed at the base station. Similarly, an appropriate data gathering schedule is pre–computed at the base station (which is less likely to be resource–constrained) and transmitted to the individual sensors. We take advantage of the fact that the base–station is aware of the locations of the sensors and have sufficient processing capabilities to compute efficient data–gathering schedule(s) for the sensors.

Table II summarizes our main results. Note that the presented values for lifetime and depth are averaged across 20 different experiments for each network size. Further, the MIN and MAX columns for R indicate the minimum and maximum performance ratios observed from those experiments. We make the following key observations:

- the lifetime of a schedule obtained using the MLDA algorithm is always within 1% of the optimal fractional solution.
- the lifetime of a schedule given by the CMLDA heuristic is always within 10% of the optimal fractional solution.
- the CMLDA heuristic significantly outperforms the LRS protocol in terms of system lifetime. In particular, the CMLDA heuristic performs 1.6 to 4.5 times better than LRS.
- the average depth of a data gathering schedule attained by the CMLDA heuristic is slightly higher than that of the LRS protocol. Note that the 3 level protocol in LRS is specifically devised to reduce the average depth of each sensor [14]. To this end, the CMLDA heuristic does quite well in attaining comparable sensor depths, while delivering significant improvements in system lifetime.

For our next set of experiments, we consider larger networks of sensors randomly distributed in a 100m $\times$ 100m field. The number of sensors in the network, i.e. the network size n, is varied to be 100, 200, 300 and 400 respectively. Each sensor has an initial energy of 1J and the base station is located at (50m, 300m). Once again, the presented values for lifetime and depth are averaged across 20 different experiments for each network size. Due to the high complexity of the algorithm, we do not include any results regarding the performance of MLDA for the large–scale networks. The clustering in CMLDA (chain formation in LRS) is done in the manner described above. We summarize our results in Table III.

We make the following observations:

- the CMLDA heuristic significantly outperforms the LRS protocol in terms of system lifetime. In particular, the CMLDA heuristic delivers system lifetimes that are 2.1 to 5.8 times larger than LRS.
- the average depth of a data gathering schedule attained by the CMLDA heuristic is only slightly higher than that of the LRS protocol.

In conclusion, our experimental results demonstrate that the CMLDA heuristic can achieve as much as a factor of 5.8 increase in system lifetime when compared to the LRS protocol, while incurring a small increase in the delay experienced by individual sensors.

INPUT		CMLDA		LRS		R	
n	c	T	D	T	D	MIN	MAX
100	10	2811	7.5	1228	6.9	2.1	3.9
200	10	4012	10.6	1412	9.6	2.4	4.2
300	15	6560	13.2	1356	12.1	3.0	4.9
400	20	9012	20.6	1621	19.6	3.6	5.8

TABLE III
EXPERIMENTAL RESULTS FOR A 100M $\times$ 100M SENSOR NETWORK.

VI. CONCLUSIONS

In this paper, we proposed a polynomial–time near–optimal algorithm (MLDA) for solving the maximum lifetime data gathering problem for sensor networks, when the sensors are allowed to perform in–network aggregation of data packets. Given the complexity of the MLDA algorithm, we next described efficient clustering–based heuristics to solve the maximum lifetime data aggregation problem in large sensor networks. Further, we presented experimental results demonstrating that the proposed methods attain significant improvements in system lifetime, when compared to existing protocols.

There are a number of important issues related to the maximum lifetime data gathering problem that need to be investigated in the future. In the work presented in this paper, we make the simplistic assumption that a sensor can always aggregate its own data packets with those of any other sensor in the network. As part of our current research, we are exploring a more complex scenario where a sensor is permitted to aggregate its own packets with only certain sensors, while acting as a router for other incoming packets. In the future, we plan to investigate modifications to the MLDA algorithm

INPUT		MLDA		CMLDA		LRS		R	
n	c	OPT	T	T	D	T	D	MIN	MAX
40	5	6611.8	6610	6442	4.5	5592	4.4	1.10	1.52
50	5	6809.0	6808	6747	5.9	5466	5.1	1.20	1.60
60	5	7176.2	7174	7096	6.0	5872	5.2	1.15	2.05
80	10	7946.9	7945	7809	6.6	6008	6.1	1.21	2.24
100	10	8292.6	8290	8011	7.2	5526	6.6	1.38	2.64

TABLE II

EXPERIMENTAL RESULTS FOR A 50M $\times$ 50M SENSOR NETWORK.

that would allow sensors to be added to (removed from) the network, without having to re–compute the entire schedule. Further, we plan to study the data gathering problem with depth (delay) constraints for individual sensors, in order to attain desired tradeoffs between the delay experienced by the sensors and the lifetime achieved by the system.

REFERENCES

[1] M. Bhardwaj, T. Garnett, and A.P. Chandrakasan. Upper Bounds on the Lifetime of Sensor Networks. In *Proceedings of International Conference on Communications*, 2001.

[2] J.H. Chang and L. Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In *Proceedings of IEEE INFOCOM*, 2000.

[3] J.H. Chang and L. Tassiulas. Maximum Lifetime Routing in Wireless Sensor Networks. In *Proceedings of Advanced Telecommunications and Information Distribution Research Program, College Park, MD*, 2000.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Max–flow Min–cut Theorem. In *Introduction to Algorithms, MIT Press*, 1998.

[5] J. Edmonds. Edge –disjoint branchings. In *Combinatorial Algorithms, Academic Press*, 1973.

[6] W. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocols for Wireless Microsensor Networks. In *Proceedings of Hawaiian International Conference on Systems Science*, 2000.

[7] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proceedings of 5th ACM/IEEE Mobicom Conference*, 1999.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of 6th ACM/IEEE Mobicom Conference*, 2000.

[9] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Mobile Networking for Smart Dust. In *Proceedings of 5th ACM/IEEE Mobicom Conference*, 1999.

[10] K. Kalpakis, K. Dasgupta, and P. Namjoshi. Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks. *To Appear in Proceedings of IEEE Networks'02 Conference*, 2002.

[11] B. Krishnamachari, D. Estrin, and S. Wicker. Modelling Data-Centric Routing in Wireless Sensor Networks. In *Proceedings of IEEE Infocom*, 2002.

[12] X. Lin and I. Stojmenovic. Power–aware Routing in Ad Hoc Wireless Networks. In *University of Ottawa, TR-98-11*, 1998.

[13] S. Lindsey and C. S. Raghavendra. PEGASIS: Power Efficient GAthering in Sensor Information Systems. In *Proceedings of IEEE Aerospace Conference*, 2002.

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

[15] L. Lovász. On two minimax theorems in graph theory. In *Journal of Combinatorial Theory Series B, Vol. 21*, 1976.

[16] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler. Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks. In *Proceedings of 4th IEEE Workshop on Mobile Computing and Systems Applications*, 2002.

[17] R. Min, M. Bhardwaj, S.H. Cho, A. Sinha, E. Shih, A. Wang, and A.P. Chandrakasan. Low-Power Wireless Sensor Networks. In *VLSI Design*, 2001.

[18] J. Rabaey, J. Ammer, J.L. da Silva Jr, and D. Patel. PicoRadio: Ad-hoc Wireless Networking of Ubiquitous Low-Energy Sensor/Monitor Nodes. In *Proceedings of the IEEE Computer Society Annual Workshop on VLSI*, 2000.

[19] S. Singh, M. Woo, and C. Raghavendra. Power–aware Routing in Mobile Ad Hoc Networks. In *Proceedings of 4th ACM/IEEE Mobicom Conference*, 1998.