August 26, 2002 TR CS-02-12

Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks

Konstantinos Kalpakis[1,2], Koustuv Dasgupta[1], and Parag Namjoshi[1]

Computer Science and Electrical Engineering Department
University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, Maryland 21250

[a]Email: {kalpakis,dasgupta,nam1}@csee.umbc.edu

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

MAXIMUM LIFETIME DATA GATHERING AND AGGREGATION IN WIRELESS SENSOR NETWORKS

KONSTANTINOS KALPAKIS [c], KOUSTUV DASGUPTA, PARAG NAMJOSHI
Department of Computer Science and Electrical Engineering,
University of Maryland Baltimore County,
Baltimore, MD 21250, USA
E-mail: {kalpakis,dasgupta,nam1}@csee.umbc.edu

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. 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 polynomial–time algorithms to solve the data gathering problem, with and without data aggregation. Further, our experimental results demonstrate that the proposed algorithms significantly outperform previous methods, in terms of system lifetime.

1 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 [7,12,13]. 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 [6]. Such obstacles arise from the limited energy, computing capabilities and communication resources available to the sensors.

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

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, a lot of research has been directed towards energy–aware routing in the context of sensor networks [3,5].

Data fusion or aggregation has emerged as a useful paradigm 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 [8] 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). Most of the previous work [2,6,9,10] in the related area aims at reducing the energy expended by the sensors during the process of data gathering. Directed diffusion [2] 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 [6] uses meta–data negotiations between sensors to eliminate the transmission of redundant data through the network. In PEGASIS [9], sensors form chains so that each node transmits and receives from a nearby neighbor. Gathered data moves from node to node, gets aggregated and is eventually transmitted to the base station. In [10], the authors propose a hierarchical scheme based on PEGASIS that reduces the average energy and delay incurred in gathering the sensed data. 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. Instead of trying to minimize the energy consumed by each sensor, in this paper we directly address the performance objective of interest, that is to maximize the lifetime of the system. To this end, we propose novel algorithms to solve the maximum lifetime data gathering problem in distributed sensor networks when (a) sensors are allowed to perform in–network aggregation of data packets, and (b) no sensor is allowed to aggregate its data packets with those of another sensor. Our solutions are near–optimal and can be computed in polynomial time.

The rest of the paper is organized as follows. In section 2, we formulate the data gathering problem and describe our algorithms to solve the problem, both with and without data aggregation. In section 3, we present experimental results based on our algorithms. Finally, in section 4 we conclude the paper.

2 The Data Gathering Problem

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 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 [5]. 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,

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

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

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

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

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.

2.1 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 [2,5,8,9]. Such aggregation can be performed when the data from different sensors are highly correlated. As in previous work [2,5,8,9], we make the simplistic assumption that an intermediate sensor can aggregate multiple incoming packets into a single outgoing packet.

The 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, where sensors are permitted to aggregate incoming data packets, with maximum lifetime.

Consider a schedule 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 $i = 1, 2, \ldots, n$,

$$\sum_{j=1}^{n+1} f_{i,j}.T\mathrm{x}_{i,j} + \sum_{j=1}^{n} f_{j,i}.R\mathrm{x}_i \leq E_i. \tag{3}$$

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 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 $f_{i,j} > 0$.

Theorem 1 *Let S be a schedule with lifetime T, and let G be the flow network induced by 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.

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 (3) at all the sensors. Clearly what needs to be found are the capacities of the edges of G. 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.

Finding a near–optimal admissible flow network. An optimal admissible flow network can be found using the following 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 is given by,

$$\text{Maximize } T \tag{4}$$

subject to the energy constraint (3) and the constraints (5)–(8) below, for each $k = 1, 2, \ldots, n$,

$$\sum_{j=1}^{n} \pi_{j,i}^{(k)} = \sum_{j=1}^{n+1} \pi_{i,j}^{(k)}, \quad \text{for all } i = 1, 2, \ldots, n \text{ and } i \neq k, \tag{5}$$

$$T + \sum_{j=1}^{n} \pi_{j,k}^{(k)} = \sum_{j=1}^{n+1} \pi_{k,j}^{(k)}, \tag{6}$$

$$0 \leq \pi_{i,j}^{(k)} \leq f_{i,j}, \quad \text{for all } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, n+1 \tag{7}$$

$$\sum_{i=1}^{n} \pi_{i,n+1}^{(k)} = T, \tag{8}$$

where all the variables are required to take integer values. For each $k = 1, 2, \ldots, n$, constraints (5) and (6) enforce the flow conservation principle at a sensor; constraint (8) ensures that T flow from sensor k reaches the base station; and constraint (7) ensures that the capacity constraints on the edges of the flow network are respected. Moreover, constraint (3) is used to guarantee that the edge capacities of the flow network respect the sensor's available energy.

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 (5)–(8) (without requiring anymore that the flows are integers, since a solution with integer flows can always be found). [d]

[d] 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.

Constructing a schedule from an admissible flow network G. 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 an aggregation tree is used by associating the value f with each one of its edges; we call f as the lifetime of the aggregation tree. Further, we define the *depth* of a sensor v to be the average of its depths in each of the aggregation trees, and the depth of the schedule to be $\max\{depth(v) : v \in V\}$.

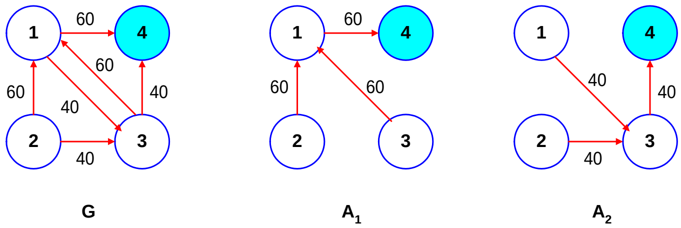

Figure 1: An admissible flow network G with lifetime 100 rounds, and two aggregation trees A_1 and A_2 with lifetimes 60 and 40 rounds respectively. The depth of the schedule with aggregation trees A_1 and A_2 is 2.

Figure 1 shows 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, sensor 2 transmits one packet to sensor 1, which in turn aggregates it with it's own data packet and then sends one data packet to the base station. 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 be the flow network that results from G after reducing the capacities of all of its edges, that are also in A, by f. We call G' the (A, f)*–reduced* G. 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 algorithm, to construct a schedule for an admissible flow network G with lifetime T, provided we can find such an aggregation tree A.

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

```
initialize f ← 1
let A = (V_o, E_o) where V_o = {t} and E_o = ∅
while A does not span all the nodes of G do
    for each edge e = (i, j) ∈ G such that i ∉ V_o and j ∈ V_o do
        let A′ be A together with the edge e
        // check if the (A′, 1)–reduction of G is feasible
        let G_r be the (A′, 1)–reduction of G
        if MAXFLOW(v, t, G_r) ≥ T − 1 for all nodes v of G
            // replace A with A′
            V_o ← V_o ∪ {i}, E_o ← E_o ∪ {e}
            break
let c_min be the minimum capacity of the edges in A
let G_r be the (A, c_min)–reduction of G
if MAXFLOW(v, t, G_r) ≥ T − c_min for all nodes v of G
    f ← c_min
replace G with the (A, f)–reduction of G
return f, G, A
```

Figure 2: Constructing an aggregation tree A with lifetime f from an admissible flow network G with lifetime T, such that the (A, f)–reduction of G is feasible.

We use the GetTree algorithm in Figure 2 to get an aggregation tree A with lifetime f from an admissible flow network G with lifetime $T \geq f$. Throughout this routine, 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. 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 sequence of aggregation trees, via the GetTree algorithm, that 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 [4,11] and is omitted due to lack of space. We refer to the approach described in this section, for finding a maximum lifetime schedule with data aggregation, as the MLDA approach.

2.2 Data Gathering without Aggregation

Data aggregation, while being a useful paradigm, is not applicable in all sensing environments. Imagine a scenario where the data being transmitted by the nodes are completely different (no redundancy) e.g. video images from distant regions of a battlefield. In such situations, it might not be feasible to fuse data packets from different sensors into a single data packet, in any meaningful way. This implies that the number and size of transmissions will increase, thereby draining the sensor energies much faster. The problem is finding an efficient schedule to collect and transmit the data to the base station, such that the system lifetime T is maximized. We call this variation of the data gathering problem as the **Maximum Lifetime Data Routing (MLDR)** problem.

Since no in–network aggregation is performed, the MLDR problem can be viewed as a maximum flow problem with energy constraints at the sensors, subject to integral flows. The MLDR program can be solved by the following integer program with linear constraints:

$$\text{maximize} \;\; T, \tag{9}$$

subject to energy constraint (3) for each sensor and the flow conservation constraints

$$T + \sum_{j=1}^{n} f_{i,j} = \sum_{j=1}^{n+1} f_{j,i}, \quad \text{for all } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, n+1, \tag{10}$$

where all variables T and $f_{i,j}$ are required to be non-negative integers. A near–optimal solution to the MLDR problem can be obtained as follows. First, solve the linear relaxation of the above integer program, by replacing the requirement that all the T and $f_{i,j}$ variables are non-negative integers, with the requirement that they are non-negative real numbers. Second, compute a solution to the linear program that consists of equations (9) and (10), by fixing the values of the $f_{i,j}$ variables to the floor of their values obtained in the previous step. The solution obtained in this second step is guaranteed to have integer values for all the variables, since it is a *max–flow problem* with integer capacities.

Observe that this solution provides us readily with a schedule for collecting the data packets without aggregation from all the sensors, during the lifetime of the system. A simple way to construct such a schedule would be to take the flow network obtained from the solution, and push T data packets from each sensor on one or more paths (with available capacities) to the base station. We define the *depth* of this schedule to be the maximum length of a path used by any sensor to transmit its data to the base station. Our solution is an approximate solution to the MLDR problem, and provides a near–optimal system lifetime that is efficiently computable. We refer to the approach described in this section, for finding a maximum lifetime schedule without data aggregation, as the MLDR approach.

3 Experiments

For the experimental results presented in this section, 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, is varied to be 10, 20, 30, 40, 50 and 60, respectively. Each sensor has an initial energy of 1J and the base station is located at (25, 150). 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 2. We compare the data gathering schedule given by the MLDA (MLDR) algorithm with that obtained from a chain-based hierarchical protocol proposed by Lindsey, Raghavendra and Sivalingam [10]. For brevity, we refer to this protocol as the LRS protocol. We choose this protocol since it significantly outperforms other competitive protocols (e.g. LEACH [5]) 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, but the only difference is 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 manner in which chain leaders are selected in each level of the hierarchy, naturally defines an aggregation tree, for each round of data gathering. For the sake of comparison, we implemented the LRS protocol to perform data gathering, with and without aggregation. In the case of no aggregation, sensors use the same chain-based hierarchy to transmit their packets to the base station. However, the packets are not aggregated when the data moves from node to node. In both cases, we fix the size of each chain at the lowest level of the hierarchy to 10 and adjust the number of levels based on the network size (with a maximum of 3 levels [10]).

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. We also compute the depth of each sensor by looking at its position in the hierarchy in each round, and the depth D of the schedule. For the same placement of sensors, we measure the lifetime and depth of the data gathering schedules obtained from the MLDA and MLDR algorithms. We define the *performance ratio* R as the ratio of the lifetime given by MLDA (MLDR) to the lifetime given by LRS with (without) aggregation, respectively. Recall that, the (integral) solution given by the MLDA (MLDR) algorithm is an approximation of the optimal (fractional) solution. In order to estimate the quality of approximation, we also measure the system lifetime given by the optimal fractional solutions (denoted as OPT) for the MLDA and (MLDR) algorithms.

n	*Without Data Aggregation*						
	OPT	**MLDR**		**LRS**		**R**	
	T	T	D	T	D	Min	Max
10	301.6	301	2.1	201	4.1	1.4	1.6
20	306.6	306	3.0	105	6.2	1.8	7.8
30	308.2	308	2.7	146	5.9	1.7	7.7
40	322.3	322	3.8	172	6.2	1.8	3.0
50	320.1	320	4.2	161	6.2	1.7	3.9
60	315.6	315	4.5	158	6.7	2.0	3.6

Table 1: MLDR results for a 50 × 50m sensor network.

n	*With Data Aggregation*						
	OPT	**MLDA**		**LRS**		**R**	
	T	T	D	T	D	Min	Max
10	5712.8	5712	4.5	5288	4.6	1.06	1.12
20	5809.2	5867	5.1	5185	4.3	1.11	1.16
30	6246.5	6265	5.0	5355	4.2	1.10	1.26
40	6611.8	6610	4.9	5592	4.4	1.15	1.45
50	6809.0	6808	5.8	5466	5.1	1.20	1.72
60	7176.2	7174	6.2	5872	5.2	1.16	1.64

Table 2: MLDA results for a 50 × 50m sensor network.

Tables 1 and 2 summarize our main results. Note that the results for each network size are averaged across 20 different experiments. 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:

- In case of no data aggregation, the lifetime of the schedule given by the MLDR algorithm is always significantly better than the lifetime given by the LRS protocol. In fact, the MLDR algorithm performs 1.4 to 7.8 times better than the LRS protocol in terms of system lifetime. In case of data aggregation, the lifetime of the schedule given by the MLDA algorithm significantly outperforms the lifetime given by the LRS protocol. In this case, the MLDA algorithm performs 1.06 to 1.72 times better than the LRS protocol in terms of system lifetime.

- The average depth of a data gathering schedule obtained from the MLDR algorithm is lower than that of the LRS protocol, while the depth of a schedule given by the MLDA algorithm is on an average slightly higher than that of the LRS protocol.

The depth D of a data gathering schedule is an interesting metric since it gives an estimate of the (maximum) average delay [e] that is incurred in sending data packets from any sensor to the base station. Note that the 3 level protocol in LRS is specifically devised to reduce the average depth of each sensor [10]. To this end, the MLDA (MLDR) algorithm does quite well in attaining comparable sensor depths, while delivering large improvements in system lifetime.

As mentioned before, the (integral) solution given by the MLDA (MLDR) algorithm is an approximation of the optimal (fractional) solution. We observed that, for the given set of experiments such an approximation leads to a reduction in the system lifetime by *no more* than 3 rounds. Based on the results in tables 1 and 2, we believe that a data gathering schedule given by the MLDA (MLDR) algorithm is near–optimal.

4 Concluding Remarks

In this paper, we described approaches to solve the maximum lifetime data gathering problem in sensor networks, with and without data aggregation. Further, we presented experimental results demonstrating that our solutions are near–optimal and attain significant improvements in system lifetime, when compared to previous protocols. In future work, we plan to investigate faster

[e] On a 1Mbps link, a 1000 bit message can incur a delay of 1 ms on each hop to the base station.

heuristics for solving the data gathering problem in large sensor networks. Further, we plan to study the data gathering problem with depth 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. R. Govindan C. Intanagonwiwat and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of 6th ACM/IEEE Mobicom Conference*, 2000.
3. J.H. Chang and L. Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In *Proceedings of IEEE Infocom*, 2000.
4. J. Edmonds. Edge –disjoint branchings. In *Combinatorial Algorithms, Academic Press*, 1973.
5. W. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocols for Wireless Microsensor Networks. In *Proc. of Hawaiian International Conference on Systems Science*, 2000.
6. 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.
7. R. H. Katz J. M. Kahn and K. S. J. Pister. Mobile Networking for Smart Dust. In *Proceedings of 5th ACM/IEEE Mobicom Conference*, 1999.
8. B. Krishnamachari, D. Estrin, and S. Wicker. Modelling Data-Centric Routing in Wireless Sensor Networks. In *Proc. of IEEE Infocom*, 2002.
9. S. Lindsey and C. S. Raghavendra. PEGASIS: Power Efficient GAthering in Sensor Information Systems. In *Proc. of IEEE Aerospace Conference*, 2002.
10. S. Lindsey, C. S. Raghavendra, and K. Sivalingam. Data Gathering in Sensor Networks using the Energy*Delay Metric. In *Proc. of IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, 2001.
11. L. Lovász. On two minimax theorems in graph theory. In *Journal of Combinatorial Theory Series B, Vol. 21*, 1976.
12. R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Wang, and A.P. Chandrakasan. Low-Power Wireless Sensor Networks,VLSI Design,2001.
13. J. Rabaey, J. Ammer, J.Silva, and D. Patel. PicoRadio: Ad-hoc Wireless Networking of Ubiquitous Low-Energy Sensor/Monitor Nodes. In *Proc. of IEEE Annual Workshop on VLSI*, 2000.