Summary of "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks"

Kimmo Kinnunen, ktkinnun@cc.hut.fi

March 14, 2005

1 Introduction

Data aggregation is one of the basic concepts of sensor networks. The idea of data aggregation, or data fusion, is to combine data from different sensors in a way that eliminates redundant transmissions. This is done to prolong sensor network lifetime.

In their article "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks" Kalpakis, Dasgupta and Namjoshi [1] discuss algorithms to achieve data aggregation. This paper presents overview and summary of algorithms presented in aforementioned article.

2 Problem statement

The overall goal of the algorithms presented is to maximize the lifetime of a data gathering sensor network.

2.1 System model

The model of the sensor network is defined as follows.

The sensor network consists of n sensor nodes labelled $1, 2, .., n$ and a base station node t labelled $n + 1$. Locations of nodes in the network (sensors and the base station) are fixed, and full network topology is known by each network node.

Each sensor monitors its environment and produces one data packet of information per *time unit*, or per *round*. Information from all the sensors is gathered in each round and sent to the base station.

The sensors have limited amount of energy, $\mathscr{E}_i$. Reception of k bit data packet by sensor i consumes amount of energy given by

$$RX_i = \epsilon_{elec} k \tag{1}$$

where $\epsilon_{elec} = 50nJ/bit$. Transferring data packet to sensor j consumes energy given by

$$TX_{ij} = \epsilon_{elec} k + \epsilon_{amp} d_{ij}^2 k \tag{2}$$

where $\epsilon_{amp} = 150pJ/bit/m^2$. Base station doesn't have energy constraints.

Each of the nodes can reach every node in the network.

An intermediate node in the network can aggregate multiple incoming packets into a single packet per time unit.

2.2 Problem statement

Lifetime T of the system is the number of rounds until the first sensor is drained of its energy.

Data gathering schedule is collection of T directed trees. The schedule defines how the data packets are gathered from the sensors to the base station. The trees in the schedule are rooted at the base station and span all the sensors.

The problem is to maximize the system lifetime T. The goal is to find data gathering schedule which solves the problem.

3 MLDA algorithm

MLDA (maximum lifetime data gathering with aggregation) algorithm solves the problem with following steps:

1. Construct an admissible flow network from the sensor network.
2. Construct a schedule $\mathscr{S}$ by generating T aggregation trees from the admissible flow network.

An flow network is admissible if each sensor is able to send its results for each T rounds to the base station given its energy constraints.

3.1 Constructing an admissible flow network

To construct an admissible flow network, following integer program is given.

Goal:

$$\text{Maximize } T \tag{3}$$

Constraints:

$$\sum_{j=1}^{n+1} f_{i,j} TX_{i,j} + \sum_{j=1}^{n+1} f_{j,i} RX_{j,i} \le \mathscr{E}_i, \tag{4}$$

$$\sum_{j=1}^{n} \pi_{j,i}^{(k)} = \sum_{j=1}^{n+1} \pi_{i,j}^{(k)} \qquad \forall i = 1, 2, \dots, n \text{ and } i \neq k \tag{5}$$

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

$$0 \le \; pi_{i,j}^{(k)} \le f_{i,j}, \qquad \forall i = 1, \dots, n \text{ and } \forall j = 1, \dots, n+1 \tag{7}$$

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

where $f_{i,j}$ is the total number of packets that node i transmits to node j, $\mathscr{E}_i$ is the amount of energy node i has, $k = 1, \ldots, n$, $\pi_{i,j}^{(k)}$ is a flow variable indicating the flow that k sends to the base station over the edge (i, j) and all variables are required to be non-negative integers.

The constraint given by equation 4 means that any valid schedule must respect the energy constraints of each sensor. Equations 5 and 6 mean that a sensor generates as much flow as is its intake. Equation 7 conserves the capacity constraints on the edges of the flow network. Equation 8 makes sure that T flow from sensor k reaches the base station.

Solving the integer program yields an optimal admissible flow network. The approximation used in the algorihm consists of solving the program as a linear program and fixing the edge capacities, $f_{i,j}$, to their floor values obtained from the solution of the linear program. After fixing the edge capacities constraints 5 - 8 are solved resulting in near-optimal admissible flow network.

3.2 Constructing a schedule

An data aggregation schedule for an admissible flow network is constructed by GETSCHEDULE algorithm that does *feasible* (A, f)*-reductions* for the network. (A, f)-reduction of a flow network G reduces the capacities of edges of A in G by amount of f. A is a directed tree rooted at the base station node and spans the whole flow network. Feasible (A, f)-reduction is such that maximum flow from each node to the base station in the resulting network is $\geq T - f$.

3.3 Analysis

Solving the problem using MLDA consists of solving a linear program twice. The linear program has $O(n^3)$ variables and coefficients. Complexity of this problem dominates the worst-case running time of MLDA, which is thus $O(n^{15} log(n))$. The complexity of the schedule construction step is $O(n^{20/3} log(n))$

4 CMLDA: clustering MLDA

MLDA requires solving linear program with $O(n^3)$ variables and constraints, which is computationally expensive for large sensor networks (n is the number of sensors in the network). The article describes two heuristics based on MLDA algorithm. Both algorihms work with clusters of sensors. Clusters $\phi_1, \ldots, \phi_m$ are called *super-sensors* and together contain all the sensors in the sensor network. Clustering is expected to be done with a proximity-based clustering algorithm.

4.1 Greedy CMLDA heuristic

Greedy CMLDA heuristic consists of following two phases.

1. Compute a schedule $\mathscr{T}$ for the super-sensors $\phi_1, \ldots, \phi_m$ using MLDA. Schedule consists of super-sensor aggregation trees $\mathscr{T}_k$.

2. Compute a schedule $\mathscr{S}$ for the whole network by applying BUILD-TREE algorithm for each super-sensor ϕ_i and aggregation tree $\mathscr{T}_k$ in $\mathscr{T}$.

In phase 1 the energy of each super-sensor is the sum of initial energies of the sensors in it and distance of two super-sensors is the maximum distance of the sensors in the super-sensors.

In phase 2 BUILD-TREE algorithm builds an aggregation tree A from super-sensor aggregation tree $\mathscr{T}_k$. The idea is to go through the nodes in each super-sensor, attaching them to a node in A that has *maximum residual energy.* This constructs an aggregation tree which maximizes the *minimum residual energy* among all sensors, thus maximizing the lifetime of the schedule.

4.1.1 Analysis

The worst-case running time of the phase 2 is $O(n^3)$. Thus greedy CMLDA has worst-case running time of $O(m^{15}log(m) + n^3)$ where n is the number of sensors, m is the number of super-sensors. The complexity of the heuristic can be adjusted by partitioning a sensor network according to worst-case estimation. For example, selecting $m = n^{3/16}$ results in worst-case running time of $O(n^3)$.

4.2 Incremental CMLDA

Incremental CMLDA heuristic consists of following phases.

1. Compute a flow network with lifetime T for super-sensors using the linear program in MLDA.
2. For each sensor s, calculate capacity provisions between s and sensors in the same super-sensor and between s and each super-sensor, minimizing $\mathscr{E}_{max}$, the maximum energy consumed by s.
3. Calculate capacities that need to be provisioned between individual sensors in different super-sensors minimizing $\mathscr{E}_{max}$.
4. Scale and floor the obtained capabilities by a factor of $\alpha = \mathscr{E}/\mathscr{E}_{max}$, obtaining admissible flow network with integer capacities and lifetime of $T_0 = \alpha T$. Use GETSCHEDULE of MLDA to calculate the schedule.

The linear program in phase 2 calculates the capacities that are needed for the sensors to ensure transmission of T packets. The capacities are between sensors in same super-sensor or provisioned capacities between a sensor and a super-sensor. This is done by minimizing the maximum energy consumed by individual sensors to ensure maximum lifetime of the network.

The linear program in phase 3 calculates the provisioned capacities of sensors in different super-sensors. This is done by likewise minimizing the maximum energy consumed.

The maximum energy consumed can be bigger than the available energy. Phase 4 scales the network lifetime and edge capacities in order to achieve an admissible flow network. Scaling factor is $\alpha = \mathscr{E}/\mathscr{E}_{max}$ where $\mathscr{E}$ is initial energy of each sensor and $\mathscr{E}_{max}$ is the maximum energy consumed by any sensor. This can be done because it is proven in the article that admissable flow network scales linearly with the energy of the sensors.

4.3 Analysis

The worst case running time of the algorithm is $O(m^{15}log(m))+O(c^5+m^{11}log(cm))+O(c^{10}log(c))+O(n^2)$ where n is the number of the sensors, m is the number of the super-sensors and c is the largest amount of sensors in a super-sensor.

5 Experiments and Results

The authors of the article [1] compare the results of algorihms to each other and also to a protocol proposed by Lindsey, Raghavendra and Sivalingam (LRS protocol).

Table 1 shows the performance of clustering MLDA algorithms compared to the optimum MLDA. Experiment simulations were conducted with network size $n = 40, 50, 60, 80, 100$ and area of 50m x 50m.

Table 1: CMLDA algorihms compared to optimum MLDA

	Greedy	Incremental
Network lifetime within	9%	3%
Average running time of	10%	33%

When the results are compared to LRS protocol, CMLDA heuristics are 1.20-2.61 times better in terms of network lifetime. The average depth of resulting schedules are slightly higher that of the LRS protocol.

The article also contains simulations for a larger networks, where both the area and network size are bigger. The results are similar to the smaller simulation with the distinction that optimum MLDA results were not presented due to the high complexity of the algorithm.

References

[1] K. Kalpakis, K. Dasgupta, P. Namjoshi, Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. Computer Networks 42 (2003), 697-716.