Modeling In-Network Processing and Aggregation in Sensor Networks

Ajay Mahimkar
mahimkar@ece.utexas.edu

Abstract

The rapid advances in processor, memory and radio technology have enabled the development of distributed networks of sensor nodes capable of sensing and communicating using wireless media. The basic operation in sensor networks is the systematic gathering and transmission of sensed data to the end-user. The severe energy constraints and limited computing capabilities of the sensors present major challenges to its design. In this paper, I propose two new protocols DEEPADS (Distributed Energy-efficient Protocol for Aggregation of Data in Sensor Networks) and C-DEEPADS (Clustered-DEEPADS) that maximize the lifetime of the sensor network. Simulation results show that the protocols perform better than the existing approaches: Directed diffusion, LEACH, PEDAP and PEDAP-PA. The two-tier clustering approach C-DEEPADS is optimal in terms of maximizing the system lifetime as well as reducing the end-to-end latency.

Index Terms

Sensor Networks, In-Network Processing, Data Aggregation, Energy-Efficient Algorithms

I. Introduction

A sensor network comprises of a large number of low-power wireless sensors spread across a geographical area that can be used to monitor and control the physical environment from remote locations. Each sensor node is battery powered and equipped with integrated sensors, data processing capabilities and short-range radio communications. The readings sensed by the sensors are routed to the end user by a multi-hop architecture through the base station. Potential applications of sensor networks include real-time traffic monitoring, battlefield surveillance and nuclear attack detection.

The main constraint of sensor nodes is their low finite battery energy, which limits the network lifetime. Network lifetime is the time at which the first node in the network completely exhausts its battery resources. The key challenge in such an environment is the design of communication protocols that maximize the network lifetime and reduce bandwidth requirements by using local collaboration among the nodes [6]. The primary focus of the paper is the design and evaluation of an energy-efficient protocol that maximizes the lifetime of the sensor network. The key idea is to combine data from different sensors to eliminate redundant transmissions, thereby leading to efficient use of the energy resources [7]. For example, in a reconnaissance-oriented sensor network, sensor readings indicate the detection of a target, while aggregation is used to track and identify the detected target.

A. Existing Approaches

In Directed diffusion [1], the sink disseminates a sensing task as an *interest* message. This sets up *gradients* within the network that point to the neighbor from which an interest was received. Directed diffusion does not consider the energy of the sensor nodes while routing data. Thus, it may route through nodes that are heavily energy-constrained, thereby decreasing the network lifetime.

LEACH [2] (Low Energy Adaptive Clustering Hierarchy), developed at MIT, is an elegant solution to the data aggregation problem where clusters are formed in a self-organized manner. A designated node in each cluster, called the *cluster-head* is responsible for collecting and aggregating the data from sensors in its cluster and eventually transmitting the result to the base station.

In PEGASIS [3] (Power-Efficient GAthering in Sensor Information Systems), data is aggregated at nodes organized as a chain. PEGASIS achieves reduction in energy consumption as compared to LEACH since only one designated node sends the combined data to the base station. PEDAP [4] (Power Efficient Data Gathering and Aggregation Protocol) outperforms LEACH and PEGASIS by routing data along the edges of the minimum spanning tree. PEDAP-PA (power-aware) further optimizes by distributing the load evenly among the nodes. The drawback of these approaches is that they optimize the energy of the node locally. Thus, data might be routed along paths that minimize the energy locally, however, consuming higher total system energy. If we globally optimize the total energy consumption from source to sink, instead of local optimization, we will have more benefits.

In this paper, I present two new new energy-efficient protocols for in-network processing and data aggregation in sensor networks. Rest of the paper is organized as follows. In Section 2, I present the model for our system. In Section 3, I give a formal description of my protocol DEEPADS, followed by the two-tier clustering approach C-DEEPADS in Section 4. In Section 5, I present simulation results. Finally, I conclude my paper in Section 6.

II. System Model

A. Model for a Sensor Node

The model for a sensor node is depicted in Fig. 1. The key blocks are :

1) Sensing - Sense the physical environment and report readings.
2) Receiving - If received data is a *query* message, broadcast it to remaining sensors. If it is a *reading*, aggregate the readings and forward it to the base station along the best available path.
3) Sending - Send data along the communication block.
4) Sleeping - Power off the sensor when the network does not require its services.

B. Energy Model

We assume that each node in the network has a unique identifier. The energy consumed due to transmission from node i to node j is represented by E_{ij}^{TX}. Receiving traffic consumes energy, denoted by E^{RX}. The cost metric used for the protocols is given below:

$$C_{ij} = \frac{1}{(BC_i - E^{RX}) + (BC_j - E_{ji}^{TX})} \qquad i \in V \text{ is a sink node or base station}$$

$$= \frac{1}{(\frac{1}{C_{ki}} - E^{RX}) + (BC_j - E_{ji}^{TX})} \qquad i \text{ is an intermediate node and } k \text{ is the predecessor of } i$$

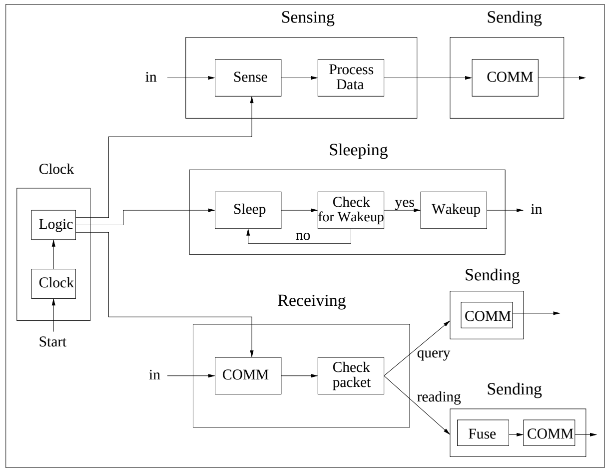

Fig. 1. **Model for a Sensor Node**

This metric reflects the cost incurred by node i if it were to receive a packet from node j. BC_i is the available battery capacity at node i. A lower value of the metric implies a higher value of residual energy. Our goal is to minimize this cost, which clearly depicts the selection of nodes with higher residual energies along the paths with minimum energy consumption.

III. DEEPADS Protocol Description

In this section, I will give a formal description of DEEPADS (**D**istributed **E**nergy-**E**fficient **P**rotocol for **A**ggregation of **D**ata in Wireless **S**ensor Networks).

A. Intuition

The approach used in query dissemination and data aggregation is similar to that used in directed diffusion [1], except that the control information will carry the energy cost metric with it. The protocol operation is as follows: The end user queries the network through either a base station or a sensor node. This node is referred to as the *sink*. The sink floods the *interest* message through the sensor network. The format of the *interest* message is $< sink\_id, seq\#, predecessor, cost\_metric >$ where $sink\_id$ is the sink node identity, $seq\#$ is the sequence number of the interest message to indicate its freshness, $predecessor$ is the 1-hop neighbor from which the *interest* message was received and $cost\_metric$ is the cost of reaching the current node from the sink. Each node also maintains a single entry for the same structure that reflects the reverse path to the sink. This is similar to setting up the *gradients*. However, unlike directed diffusion, DEEPADS maintains a single gradient that points to the predecessor along the least cost path in terms of the energy metric. Fig. 2 illustrates the operation of DEEPADS.

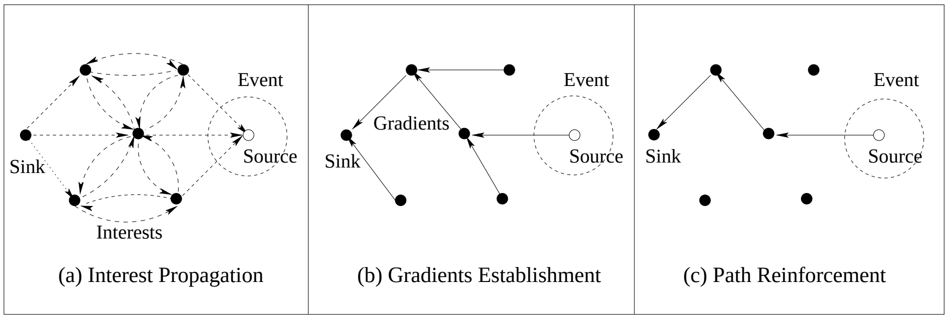

Fig. 2. **Schematic for DEEPADS. (a) Interest Messages are flooded through the network. (b) Each node sets up a single reverse gradient along the least cost path. (c) Source Node transmits data along the least cost path**

B. Formal Description of the Protocol

Each node in the network maintains a table T that comprises of entries of the form $< sink\_id, seq\#, predecessor, cost\_metric >$. On receiving the *interest* message, the node updates the table depending upon the cost metric. The notations used are: t is a 4-tuple entry in the table T. m is the interest message. It is a 4-tuple, same as t. $m.predecessor$ is the node from which the interest message was received. $chan[1..K]$ is an array of K 1-hop immediate neighbors. $chan[k]$ corresponds to the identity of the 1-hop immediate neighbor.

DEEPADS protocol is executed at each node in a distributed fashion. As the nodes receive the interest message m, they check to see if there exists a shorter path (in terms of smaller energy metric). If there exists one, they modify the predecessor in their table T and unicast this to their other neighbors. When the algorithm converges, each node will have a pointer to its predecessor along the least cost path to the sink node. As data traverses along the intermediate nodes to the sink node, it is aggregated along the path.

Formal Description of DEEPADS Protocol for Node i

```
On receiving interest message m from node j
begin
   if m.sink_id not found in table T then  add m to the table T;
   else
      begin
         find tuple t for the same sink_id as in m;
         if m.seq# > t.seq# then  replace entry for t in T by m;
        else
           if m.seq# = t.seq# and m.cost_metric < t.cost_metric then
                begin
                   t.predecessor := j;
                   t.cost_metric := m.cost_metric;
                   for k = 1 to K do
                      begin /* Build the interest message m' */
                         m' := m;  /* sink_id, seq#, predecessor are same as that of m */
                         m'.cost_metric := 1 / (1/m.cost_metric - E^{RX} + BC_{chan[k]} - E^{TX}_{chan[k] i});
                         Unicast this m' to node chan[k];
                      end
                end
              else  Drop the message; /* m.seq# < t.seq#*/
        end
 end
```

IV. Extensions to DEEPADS : A Two-Tier Clustering Approach

DEEPADS reduces the number of transmissions when there are multiple source nodes and thereby increases the system lifetime. However, each node has to wait for aggregation till it gets data from the node, for whom it is the predecessor. This increases latency, which may be undesirable in some applications.

In this section, I extend DEEPADS by developing a two-tier clustering approach. Given, a network topology, divide it into clusters, each cluster represented by a *cluster-head*. For communication within clusters, DEEPADS approach is used with sink node as the cluster-head. Data aggregation is done at the *cluster-head*. We denote cluster-head communication as tier-1 and communication within the clusters as tier-2. The two-tier approach is referred to as C-DEEPADS (Clustered-DEEPADS).The end-to-end latency is reduced since, aggregation at different cluster-heads can occur simultaneously.

A. Cluster Formation and Maintenance Algorithm

After the nodes are deployed, the base station divides the geographical region into clusters, such that each sensor belongs to only one cluster. It picks one node from each cluster to be the cluster-head and multicasts this information to each cluster. Periodically the role of the cluster-head is rotated to load balance the energy consumption. At the beginning of the interval, the current cluster-head builds a *CH-leader-elect* message and broadcasts it to all nodes in its cluster. On receiving the message, the nodes transmit a 2-tuple $< current\_battery\_capacity, node\_id >$ to the cluster-head. The cluster-head topologically sorts the 2-tuple in the decreasing order of $current\_battery\_capacity$ and elects the first node as the cluster-head for the next interval. This is broadcast to all nodes within the cluster.

V. Simulation Results

In this section, I will compare the performance of DEEPADS and C-DEEPADS with directed diffusion, LEACH, PEDAP and PEDAP-PA. The environment used for the experiments is Ptolemy-II, VisualSense [5] and Java. For the experiments, I have uniformly distributed 100 nodes over an area with dimension 100m x 100m. The experiments were run for different number of source nodes and one sink node. All nodes have equal initial battery energies (2 Joules). Each node is equipped with a single transceiver with range 50 meters.

A. Comparison of System Lifetime

Figures 3 and 4 compare the system lifetime of the protocols when there are one and ten source nodes respectively. In each round, there is communication between the sink and source node. Fig. 3 (a) shows that DEEPADS and C-DEEPADS have a higher lifetime than the existing approaches. Fig. 3 (b) shows that the total energy consumed by DEEPADS and C-DEEPADS is lower than that of the existing approaches. Figures 4 (a) and (b) depict the system lifetime and the total energy consumption when ten source nodes transmit data in each round.

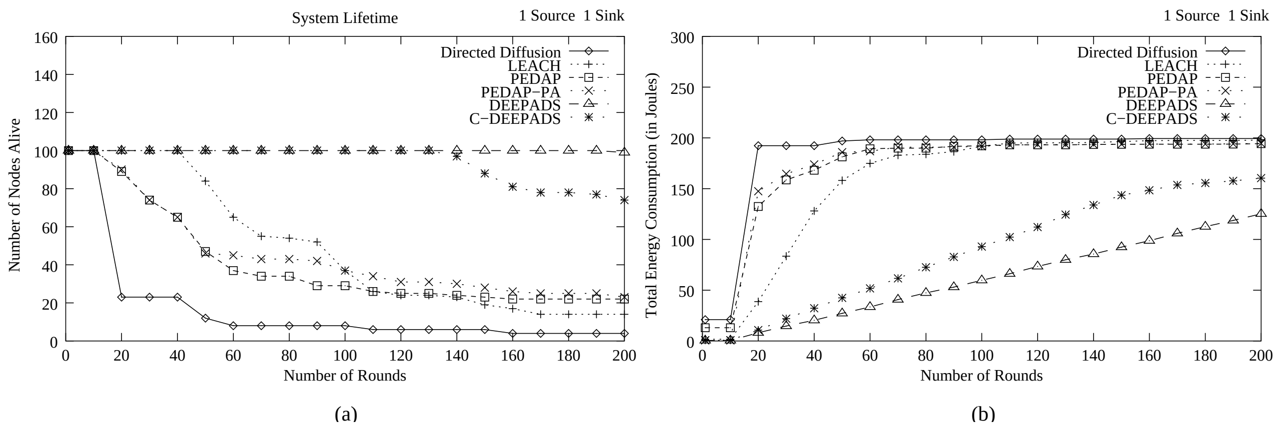

Fig. 3. **Simulation results with 1 source, 1 sink communication in each round. Fig. (a) depicts the system lifetime v.s. number of rounds and Fig. (b) depicts the total energy consumption (in Joules) v.s. number of rounds.**

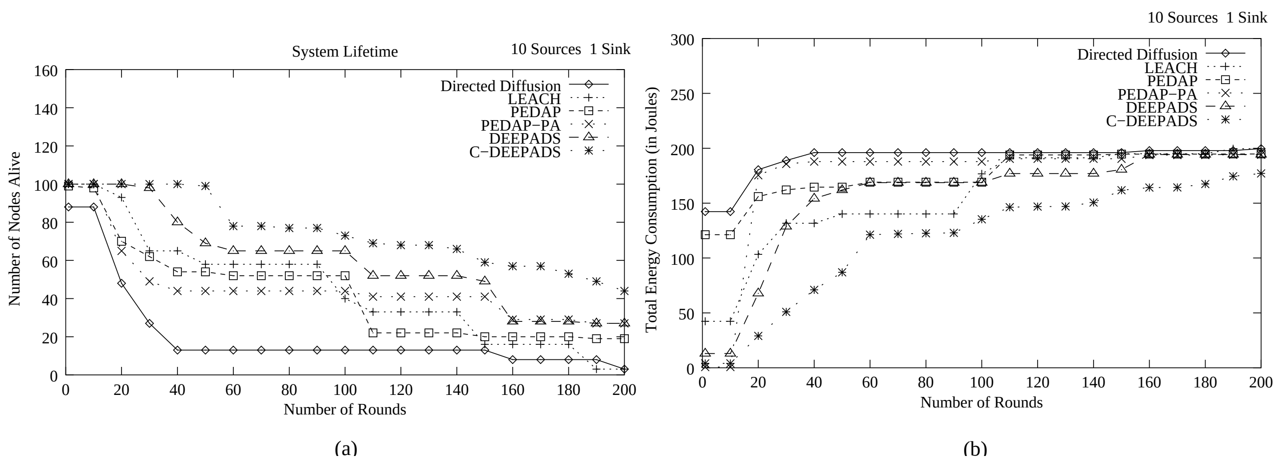

Fig. 4. **Simulation results with 10 Sources, 1 Sink communication in each round. Fig. (a) depicts the system lifetime v.s. number of rounds and Fig. (b) depicts the total energy consumption (in Joules) v.s. number of rounds.**

B. Latency Analysis

Latency is the time between the events when the sink node sends the interest message to the network and receives the aggregated reading from the network. For comparative evaluation, I performed experiments where all nodes want to transmit to the sink node. As seen in Table I, the two-tier approaches LEACH and C-DEEPADS have a lower latency than the flat ones. For LEACH and C-DEEPADS, as the number of clusters is increased from 8 to 16, latency reduces due to more parallelism of aggregation within clusters, but as you go from 16 to 24, latency increases. This is because with 24 clusters, there is more contention between cluster-heads at the media access leading to higher delay.

Cluster Size	Directed diffusion	LEACH	PEDAP	PEDAP-PA	DEEPADS	C-DEEPADS
8	92.5	63.8	88.5	87.5	87.4	59.6
16	92.5	61.8	88.5	87.5	87.4	57.4
24	92.5	74.3	88.5	87.5	87.4	63.3

TABLE I
END-TO-END LATENCY (VALUES ARE IN MILLISECONDS)

C. Overhead Analysis

There is an overhead involved for maintaining clusters in C-DEEPADS. The number of messages broadcast to elect a new cluster-head are of the order $O(N_i^2)$, where N_i is the number of nodes in cluster i. The complexity reduces to $O(log(N_i))$ by forwarding messages along a spanning tree.

VI. CONCLUSION

In this paper, I proposed two novel energy-efficient approaches DEEPADS and C-DEEPADS to in-network processing and data aggregation in wireless sensor networks. They use energy as the metric for routing data from source to sink, thereby leading to increased system lifetime. Simulation results show that DEEPADS and C-DEEPADS outperform the existing approaches. Using the two-tier approach, C-DEEPADS further reduces the end-to-end latency. The two-tier clustering approach C-DEEPADS is optimal in terms of maximizing the system lifetime as well as reducing the end-to-end latency.

REFERENCES

[1] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, “Directed Diffusion for Wireless Sensor Networking”, *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 2-16, Feb. 2003.

[2] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, “An Application-Specific Protocol Architecture for Wireless Microsensor Networks”, *IEEE Trans. Wireless Communications*, vol. 1, no. 4, pp. 660-670, Oct. 2002.

[3] S. Lindsey, C. Raghavendra, K. M. Sivalingam, “Data Gathering Algorithms in Sensor Networks using Energy Metrics”, *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924-935, Sept. 2002.

[4] H. O. Tan, I. Korpeoglu, “Power Efficient Data Gatherting and Aggregation in Wireless Sensor Networks”, *Proc. ACM Int. Conf. Management of Data (ACM SIGMOD)*, vol. 32, no. 4, pp. 66-71, Dec. 2003.

[5] P. Baldwin, S. Kohli, E. A. Lee, X. Liu, Y. Zhao, “Modeling of Sensor Nets in Ptolemy II”, *Proc. Information Processing in Sensor Networks*, pp. 359-368, Apr. 2004.

[6] D. E. Culler, J. Hill, P. Buonadonna, R. Szewczyk, A. Woo, “A Network-Centric Approach to Embedded Software for Tiny Devices”, *Proc. Int. Workshop Embedded Software (EMSOFT)*, pp. 114-130, Oct. 2001.

[7] G. Asada, A. Burstein, D. Chang, M. Dong, M. Fielding, E. Kruglick, J. Ho, F. Lin, T. H. Lin, H. Marcy, R. Mukai, P. Nelson, F. Newberg, K. S. J. Pister, G. Pottie, H. Sanchez, O. M. Stafsudd, S. Valoff, G. Yung, W. J. Kaiser, “Low power systems for wireless microsensors”, *Proc. IEEE Int. Symposium Low Power Electronics and Design*, pp. 17-21, Aug. 1996.