RESEARCH PAPER
International Journal of Recent Trends in Engineering, Vol 1, No. 1, May 2009

Energy Efficient In-Network Data Processing in Sensor Networks

Prakash G L[1] , Thejaswini M[1] , S H Manjula[1] , K R Venugopal[1] , L M Patnaik[2]
[1] UVCE, Department of Computer Science and Engineering , Bangalore, India
Email: glprakash@yahoo.com
[2]Defence Institute of Advanced Technology(Deemed University), Pune, India.

Abstract—We assume a sensor network with data-centric storage, where sensor data is stored within the sensor network and adhoc queries are disseminated to and processed in the network. In such minimizing energy consumption has been a major objective at all levels in sensor networks. In this paper, we present an in-network aggregation scheme that maintains the user-specified quality of data requirement while significantly reducing the overall energy consumption. Specifically, since communication dominates power usage in sensor networks, in-network aggregation scheme exploits end-user temporal coherency tolerances to reduce the amount of information transmitted by individual nodes. Further, we show that in-network aggregation scheme, by using temporal coherency tolerances, can allow for better quality of data when the time given to perform readings is too short for all data to be propagated up through the network. We compare our proposed scheme against an existing in-network aggregation scheme with a local sensor cache. We present experimental results measuring both power savings and also the quality of data for group-by queries. These results show that in-network aggregation scheme can reduce power consumption without any loss in the quality of data and extend the life of the sensor network.

Keywords—Aggregation, Energy Consumption, Query Plan, Network Lifetime.

I. INTRODUCTION

A Wireless sensor network (WSN) consists of a large number of sensor nodes deployed over an area and integrated to collaborate through a wireless network. WSNs encourage several novel and existing applications such as environmental monitoring. Sensor nodes, such as the Berkeley MICA Mote which gathers data such as light and temperature, are getting smaller, cheaper, and able to perform more complex operations, including having mini operating systems embedded in the sensor. While these advances are improving the capabilities of sensor nodes, there are still many crucial problems with deploying sensor networks. Limited storage, limited network bandwidth, poor inter-node communication, limited computational ability, and limited power still persist. In this paper we focus on the problem of limited power in sensor nodes.

One way to help alleviate the problem of limited power is through in-network query processing rather than query processing at the base station. For example, assume a query which counts the number of sensors in the network. One way to implement this is to have each sensor count itself and send that count up the network to the base station, with intermediate nodes just routing packets up the network. Another way, with in-network query processing (or aggregation), would be for each node to send the count of itself and all of its children. In this way, only one packet needs to be sent per node and each intermediate node adds the count of itself to that of its children before sending information further up the network.

As the example shows, with in-network aggregation some of the computational work of the aggregation is performed within the sensor node before it sends the results out to the network. The reason why in-network aggregation reduces power consumption is that sensor power usage is dominated by transmission costs. Therefore, being able to transmit less data (the result of the aggregation over having to forward all the packets) results in reduced energy consumption at the sensor nodes.

Our approach is to use temporal coherency tolerances in addition to in-network aggregation to save energy in the sensor network while retaining the users specified quality of data requirement . Our scheme, which is Temporal coherency-aware in-Network Aggregation is independent of the underlying synchronization protocol used for sending and receiving data between the sensor nodes.

The basic idea underlying the temporal coherency tolerance is to send a reading from the sensor only if the reading differs from the last recorded reading by more than the stated tolerance. These tolerances are based on user preferences or can be dictated by the network in cases where the network cannot support the current tolerance level. Temporal coherency checks occur at the level of individual readings and are compared against the last reading available for a sensor node.

In order to specify temporal coherency tolerance, we introduce a new clause in the expression of continuous queries. While it is the WHERE clause that acts as a (input) result filter, this new clause acts as a (output) transmission filter. By being a transmission filter, in-network aggregation is able to save energy for two reasons. First, at the edge nodes (i.e., the leaf nodes of the routing tree), if a new reading falls inside the given tolerance the reading is not transmitted. Secondly, at the internal nodes, if aggregation eliminates values, transmitted messages have smaller size. For example, in group-by queries the length of the messages sent by a node depends on the number of groups existing in the routing tree rooted at that node. By not transmitting at the leaf level, there are cases where a group is no longer showing up at an internal node. This can propagate up the tree and result in additional savings for the sensor network.

We have experimentally evaluated our proposed temporal coherency tolerance scheme using simulation. We have studied the effect of in-network aggregation on different group-by and aggregation type queries, as well as how in-network aggregation is affected by the rate that data changes. Additionally, we looked at in-network aggregation effects on the lifetime of the sensor network. Our results show that our method, by not sending and by decreasing the size of messages, provides large gains in power savings over previous methods of in-network aggregation while minimizing the impact on quality of data. Specifically, these results show that in-network aggregation can reduce power consumption and increases the quality of data compared to an existing in-network aggregation method.

In sensor networks, information or data may be described by using attributes. In order to integrate tightly with the information or data, a routing protocol may be designed according to data-centric techniques. A data-centric routing protocol requires attribute-based naming which is used to carry out queries by using the attributes of the phenomenon. In essence, the users are more interested in the data gathered by the sensor networks in the phenomenon rather than by an individual node. They query the sensor networks by using attributes of the phenomenon that they want to observe.

*Contribution :*In summary, our contribution in this paper is four-fold:

1) We propose materialized views in sensor networks which dynamically replicate query results and enable the sharing of query processing among similar queries.
2) We propose a query processing algorithm that utilizes the materialized views when processing similar queries.
3) We formulate the candidate selection problem, i.e., to an optimal subset of candidate sensor nodes to answer a query, as an optimization problem. We then design a greedy algorithm to solve the problem.
4) We conduct an extensive simulation study to evaluate the performance of our proposal, comparing it to query processing schemes for sensor networks.

Our simulation results demonstrate that aggregation reduces energy consumption on queries with negligible increase in response time, and without compromising the quality of data.

Organization : The remainder of this paper is organized as follows. In Section II we provide a general overview of sensor networks and present related work. Section III discusses the Problem definition. Detailed system design and in-network aggregation is described in Section IV. Our Simulation results can be found in Section V. We conclude in Section VI.

II. Related work

Two well known query processing engines for sensor networks are TinyDB and Cougar [1], [2]. In these systems, a user injects a query at the access point. Upon receiving the query, the access point collects meta data in an aggregated form from all nodes participating in the query. Based on the collected meta data, the access point generates a single query plan that defines the order a node samples its sensors. Next, the access point utilizes a minimum spanning tree to disseminate the chosen query plan to all nodes participating in the query. Nodes receiving the query execute the query plan and send the resulting data to the access point via the minimum spanning tree.

Many data dissemination and query processing architectures in sensor networks have been proposed, e.g., [3]. These proposals can be grouped into two categories based on the type of query supported, i.e., continuous or ad-hoc queries, and on the location where the data are stored, i.e., within or outside the sensor network.

In-network aggregation [4] could reduce energy consumption by aggregating data en route. However, it works typically for continuous aggregation queries such as SUM, MIN and MAX while it is not quite suitable for ad hoc queries. Besides, a setup phase is required such that queries are added from the base station to the network and then a routing tree rooted at the base station is built. If the duration of the query is not long enough, the setup phase will dominate the cost.

Existing research has proposed several parameterized query optimization systems which postpone part of the query optimization process until runtime [5], [6]. In sensor networks, each node is equipped with a query optimizer which chooses a set of query plans and the conditions utilized to select the appropriate query plan at runtime. By postponing parts of the query optimization process until runtime this approach is capable of gracefully re-optimizing to account for expected changes in the selectivity of the query plans predicates. The problem with this approach, is that it does not adapt well to unexpected changes because the set of query plans chosen by the query optimizer does not contain all possible query plans.

A more aggressive strategy proposed in Eddies [7] combines query optimization and query execution. In this approach, a query plan is defined at runtime on a tuple to tuple basis. Unlike traditional relational query optimizer, which utilize collected meta data, this approach utilizes dynamics and a lottery based ticketing scheme in order to determine at runtime the selectivity of the queries predicates.

III. Problem definition

A. Network Model

Given an application graph AG = <C, Q> is a directed acyclic graph (DAG) with vertex set C representing the set of computation tasks, and edge set Q representing the set of communication tasks. We often use tasks or activities to refer to both computation and communication tasks. Also, for a communication link, the terms ”sending node” and ”sender” are used interchangeably, as are the terms ”receiving node” and ”receiver”. In such minimizing energy consumption has been a major objective at all levels in sensor networks are

- To improve a data processing method to reduces the data size.
- To improve a communication model to lower the number of transmissions.
- To reduce energy by sending the data to be transferred to the base station.

B. Example

For example, the users may send out a query such as, find the locations of areas where the temperature is over 70^{o}F. Furthermore, a data-centric routing protocol should also utilize the design principle of data aggregation a technique used to solve the implosion and overlap problems in data-centric routing .

As shown in Figure 1, the sink queries the sensor network to observe the ambient condition of the phenomenon. The sensor network used to gather the information can be perceived as a reverse multicast tree, where the nodes within the area of the phenomenon send the collected data toward the sink. Data coming from multiple sensor nodes are aggregated as if they are about the same attribute of the phenomenon when they reach the same routing node on the way back to the sink.
For example, sensor node E aggregates the data from sensor nodes A and B while sensor node F aggregates the data from sensor nodes C and D in Figure 1.

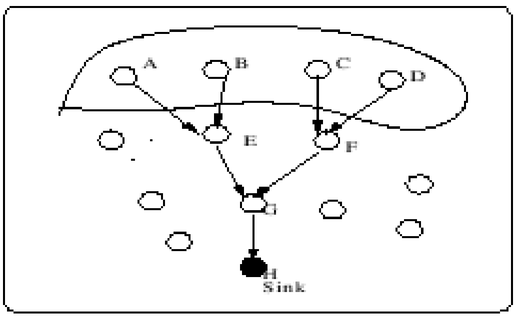

Fig. 1: Data Aggregation

C. Assumptions

1) A query issued in an environment typically specifies sensing types(photo, light, temperature, location, acceleration, magnitude), source node, set of predicates and sample period.
2) Every node holds a symmetric connectivity list of its neighbours.
3) Every node maintains a black list of neighbours of insufficient connectivity. All packets from or to a black node are dropped.
4) Every node holds an interest cache and a data list.
5) All nodes have similar capability and equal significance.
6) Each of the node is battery operated and fixed residual energy level.

IV. System design

A. Query Model

The order in which a node samples its sensors conventionally referred to as a query plan, This can be a crucial factor affecting the energy consumed by the sensor network. Such orderings for the nodes involved in a query are an essential part of query plan. The data collected by the sink node can be used to determine energy-efficient query plans for the nodes participating in the query. It is important to note that the cost of determining the optimal query for a node depends on the complexity of the query. While for simple queries, a node may itself be able to derive the optimal query plan by spending a small amount of energy or memory, for complex queries, it might be desirable to delegate this task to the energy or memory rich sink node.

The goal of the In-network query workload design is to reveal the performance characteristics of in-network query processing techniques. The query workload as follows.

Q1: Duplicate-Insensitive Simple Aggregation

```
SELECT MAX(photo)
  FROM sensors
```

Q1 tests the performance of the aggregation schemes for duplicate-insensitive aggregates. All nodes in the network participate in the aggregation process.

Q2: Duplicate-Sensitive Simple Aggregation

```
SELECT SUM(photo)
  FROM sensors
```

Q2 tests the performance of the aggregation schemes for duplicate-sensitive aggregates. The duplicate-sensitivity of the aggregate requires extra effort in multi-path routing in order to ensure the correctness of query results.

Q3: Aggregation with Sensory Attribute Selection

```
SELECT AVG(photo)
  FROM sensors
  WHERE photo ≥ C
```

In comparison with Q1 and Q2, Q3 adds a selection predicate on the aggregation attribute. The predicate selects a subset of the nodes in the network to participate in the aggregation and this subset may change over epochs of the query depending on the data.

B. Energy Cost Model

In our experiments we assume the commonly used energy consumption model for wireless sensor networks in which there is a base station (BS) that can provide access to the network of sensors. The base station is assumed to have unlimited energy. All of the other sensor nodes in the network have limited energy and each sensor node consumes energy in four main activities: transmitting, listening, processing, and sampling. That is, the total energy consumption at each node is captured by:

$$T_{cost} = E_{transmitting} + E_{processing} \quad (1)$$

We focused on transmission power since the amount of time spent listening using the DAG synchronization is the same for all sensor nodes and sampling is dependent on the synchronization and tree building methods, of which our scheme is independent. We did not include energy required for processing because it is negligible compared to that needed for communication as was observed in [3]. The cost for transmission has to take into account not only the packet size s, but also the distance between the sender and receiver d. This makes the cost of the sender to be

$$E_{transmitting} = s \quad E_{Tx} + s \quad E_{Amp} \quad d^2 \quad (2)$$

where E_{Tx} is the cost for using the transmitter (i.e., the bit cost for the transmitter electronics) and for the amplifier cost (i.e., the the per bit cost per square meter amplifier cost).

As mentioned before, a sensor node will send its data to the root through its assigned parent. A parent node is one hop away from its child, and one hop closer to the root than its child. So every node sends its data exactly one hop away, all of which are the same distance from one another. This allows us to assume a uniform/constant amplifier cost of transmitting data. In essence, this makes the sender cost above to only be . Therefore, the energy $s \quad (E_{Tx} + E_{Amp})$ consumed in transmitting depends only on the aggregate size and the number of messages containing this aggregate.

C. In-Network Data Processing

Data stored in sensor networks can be viewed as local, external and data centric. In local storage, data is stored on nodes locally; to retrieve data a query floods the network. In external storage, data is sent to sink node without waiting for the query. In data centric storage all communication is for named data.

1) Broadcasting Query Message: This is the simplest scheme. Sink node broadcast query message(BQ). Each source sensor node sends a data packet consisting of a record towards the sink. Computation will only happen at the sink after all the records have been received. This may consume more power to communicate with far nodes and computation at sink node.

2) Processing Data Locally: Instead of sending all the data to the sink node, send the locally processed data to the sink which will optimize the power consumption and communication radio energy, e.g., instead of sending all the raw temperature readings, we send partially aggregated(PA) data such as average of every seven readings from intermediate node and send it to the sink for further processing.

3) Packet Merging: In Packet Merging(PM), instead of sending each sensor readings separately in a packet we can merge several records into large packet, consisting of many readings. Packet merging is the only way to reduce the number of bytes transmitted. This will save power consumption of source node and reduces the computation cost of sink node.

V. PERFORMANCE EVALUATION

Simulation results performed on a test bed using TOSSIM simulator for TinyOS. Using PowerTOSSIM to estimate the total energy consumption of in-network data processing approaches. To estimate the power consumption of the mica2 sensor node mica2 energy mode is used. Suppose a sensor is operating at 3 Volts and capable of transmitting data at a rate of 40 Kbps at 0.012 Amp transmit current draw. Hence, the energy cost of transmitting (TE_{ctrans}) one bit in Joules is computed as $TE_{ctrans} = 3 \quad 0.012 \quad (1/40,000) = 0.9\mu Joules$.

A. Simulation Setup

In this section, simulation studies are compare the per- performance of the packet broadcasting, packet merging with packet aggregation with respect to its lifetime using TOSSIM simulator in windows operating system. Different number of Sensors are randomly distributed in a query region over 100m x 100m area. The Simulation is run for different network size. The simulation parameters for query processing a are listed in Table I.

TABLE I: SIMULATION PARAMETERS FOR QUERY PLAN

Parameter Type	Test Value
Number of nodes	5,20,50,65,75,85,100
Sink node	Mote 0
Radio model	Lossy
Distance scaling factor	1.0 with empirical radius
Simulator hardcoded	4Mhz
Epoch Period	1000ms-10000ms
Aggregate operations	SUM,AVG,MAX
Sensor type	Photo sensor, Temperature sensor, Demo sensor, Accelerometer sensor, Magnetometer sensor

B. Performance Analysis

From the simulation results, Figure 2, illustrates the performance analysis of a simple query(SQ) of sensing photo reading above some threshold value and increased workload query of detecting photo, temperature, accelerometer and magnetometer in x and y directions, and all readings above some threshold values which influences the performance metrics such as lifetime of the network. Energy consumption for sparse networks is increases linearly and for dense networks simple query increases faster than workload query.

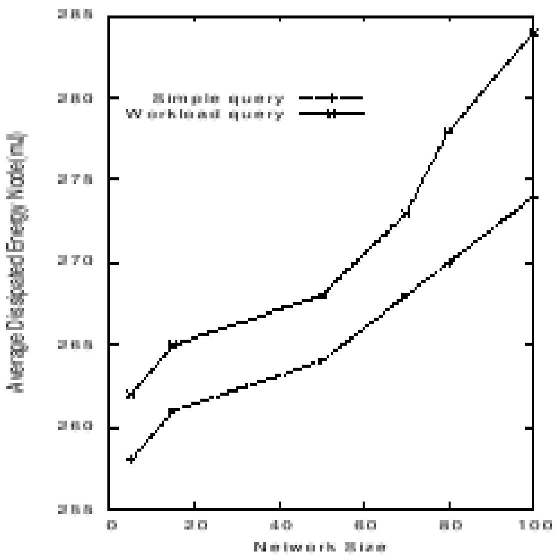

Fig. 2: Average Dissipated Energy versus Network Size for different query type

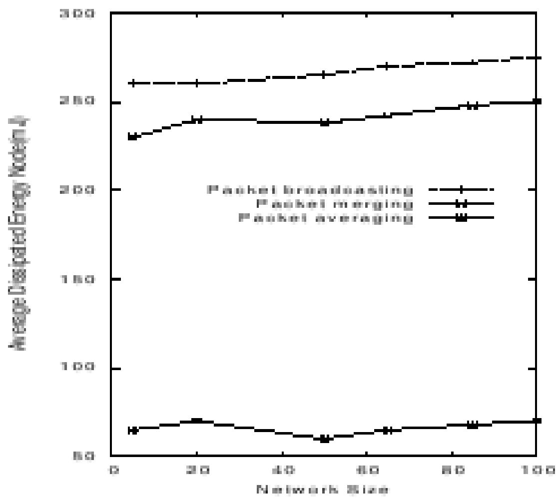

Fig. 3: Average Dissipated Energy for In-Networks Data Processing Techniques

Figure 3 illustrate the variation of average dissipated energy per node with different network size. This figure compares the energy dissipation of data processing techniques such as packet broadcasting messages, processing data locally that is partially aggregating values on local nodes, and packet merging. Without in-network data processing, each node has to send a data packet for each node whose route goes through n number of nodes, so energy consumption increases very fast. Packet broadcasting consists of all raw data, consumes more energy Packet merging consumes less energy than packet broadcasting as it consists of several sensor readings merged in a packet. Packet aggregation in in-network data processing method consumes less energy compared to other methods, it reduces redundancy in sensor readings.

VI. Conclusions

Recent advances in hardware development have enabled the use of wireless sensor networks in a myriad of monitoring applications. Being battery-powered, sensor network designs must be vigilant about energy conservation in order to increase the lifetime of the deployment. The contribution of this paper is a new scheme for doing temporal in-network aggregation, called in-network aggregation, which balances the trade-off between the quality of results returned to users and energy consumption. in-network aggregation extends current in-network aggregation methods by utilizing temporal coherency tolerance to minimize the size and number of transmitted messages. Since data transmission is the biggest energy-consuming activity in sensor nodes, using in-network aggregation results in significant energy savings.

Our experiments have shown large savings in energy over typical in-network aggregation methods without significant loss in quality of data. in-network aggregation has also been shown to increase quality of data in cases where the period to send is too short and can increase the lifetime of the network. In conclusion, in-network aggregation provides a good trade-off between decreasing energy versus decreasing quality of data in sensor networks. Currently, we are expanding in-network aggregation to consider spatial and topological redundancy.

REFERENCES

[1] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “The design of an acquisitional query processor for sensor networks,” *in 2003 ACM SIGMOD international conference on on Management of data,* pp. 491–502, June 2003.

[2] Y. Yao and J. Gehrke, “Query processing for sensor networks,” *in Proceedings of the First Biennial Conference on Innovative Data Systems Research, Asilomar, CA,* pp. 21– 32, 2003.

[3] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TAG: a tiny aggregation service for ad-hoc sensor networks,” i*n ACM SIGOPS Operating Systems Review*, vol. 36, pp. 131–146, 2002.

[4] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, “Synopsis diffusion for robust aggregation in sensor networks,” *In SenSys 04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems,* pp. 250–262, 2004.

[5] A. Deshpande, C. Guestrin, W. Hong, and S. Madden, “Exploiting correlated attributes in inquisitional query processing,” *in ICDE 05: Proceedings of the 21st International Conference on Data Engineering (ICDE05). Washington, DC, USA: IEEE Computer Society,* pp. 143–154, 2005.

[6] Y. E. Ioannidis, R. T. Ng, K. Shim, and T. K. Sellis, “Parametric query optimization,” i*n VLDB Journal: Very Large Data Bases,* vol. 6, pp. 132–151, 1997.

[7] S. Madden, M. Shah, J. M. Hellerstein, and V. Raman, “Continuously adaptive continuous queries over streams,” *in ACM SIGMOD Intl. Conf. on Management of Data, Madison, WI*, pp. 49–60, 2002.