International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-1 Issue-6, August 2012

Energy Efficient Query Optimization in Wireless Sensor Networks

Ankita Bharaktya, S.G.Reddy

Abstract – A wireless sensor network (WSN) is a wireless network consisting of distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. A WSN Consists of ten to thousand of sensor nodes that communicate through wireless channels for information sharing and cooperative processing. In this our main focus is on base station energy-efficient queries optimization. Different from existing query optimization techniques that consider only query plans for extracting data from sensors at individual nodes, our approach takes into account both of the sensing and communication cost in query plans. When a new query is submitted to the base station, we check whether the new query can be evaluated using the result of currently running queries. If it is possible then we rewrites a new query using currently running queries at the base station without injecting it into the sensor network. Thus optimizing the query processing in Wireless Sensor Network. Simulation results show queries transmitted in merging, rewriting query plans and shows the query plan chosen in our approach consumes significantly less energy than an approach that optimizes on sensing cost only.

Index Terms — Energy Efficiency, Query Optimization, Query Sharing, Sensor Networks, Wireless Sensor Networks.

I. INTRODUCTION

Recent advances in computing technology have led to the production of a new class of computing devices: the wireless, battery-powered, smart sensor. Traditional sensors deployed throughout buildings, labs, and equipments are passive devices that simply modulate a voltage on the basis of some environmental parameter. These new sensors are active, full-fledged computers, capable of not only sampling real-world phenomena but also filtering, sharing, and combining sensor readings with each other and nearby Internet-equipped end points.

Wireless Sensor Network is the network is composed of many autonomous and compact devices called sensor nodes. Each sensor node is endowed with a limited amount of processing, but when coordinated with the information from other nodes, they have the ability to measure the given physical environment in great details or to execute a task with complex functions. The objective of this network is to collect data. Resources of sensor nodes are storage capacity and its computation power and communication system. Sensor networks are composed of thousands of resource constrained sensor nodes and also some resourced base stations are there. All nodes in a network communicate with each other via wireless communication. Moreover, the energy required to transmit a message is about twice as great as the energy needed to receive the same message. Query Processing is the method to get the sensor readings from sensor.

As the number of queries that are running in the sensor network increases, the energy consumption of the sensor network increases. This is because each sensor node has to send its readings towards the base station. Thus in order to reduce the energy consumption of the sensor network, it is important to reduce the number of monitoring queries injected into the network sensor.

II. PRELIMINARIES

A. Sensor Networks

A sensor network consists of a large number of sensor nodes. Individual sensor nodes (or short, nodes) are connected to other nodes in their vicinity through a wireless network, and they use a multihop routing protocol to communicate with nodes that are spatially distant. Sensor nodes also have limited computation and storage capabilities: a node has a general-purpose CPU to perform computation and a small amount of storage space to save program code and data. We will distinguish a special type of node called a gateway node. Gateway nodes are connected to components outside of the sensor network through long range communication (such as cables or satellite links), and all communication with users of the sensor network goes through the gateway node. Since sensors are usually not connected to a fixed infrastructure, they use batteries as their main power supply, and preservation of power is one of the main design considerations of a sensor network . This makes reduction of message traffic between sensors very important.

B. Sensor Data

A sensor node has one or more sensors attached that are connected to the physical world. Example sensors are temperature sensors, light sensors, or PIR sensors that can measure the occurrence of events (such as the appearance of an object) in their vicinity.

Thus each sensor is a separate data source that generates records with several fields such as the id and location of the sensor that generated the reading, a time stamp, the sensor type, and the value of the reading. Records of the same sensor type from different nodes have the same schema, and collectively form a distributed table. The sensor network can thus be considered a large distributed database system consisting of multiple tables of different types of sensors.

Manuscript published on 30 August 2012.
***** Correspondence Author (s)
Ankita Bharaktya*, Department Of Computer Science, Mewar University, Chittorgarh, India.
S. G. Reddy, Department Of Computer Science, CMR Engineering College, Hyderabad, India.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

SELECT {attributes aggregates}
FROM {Sensordata S}
WHERE {predicate}
GROUP BY {attributes}
HAVING {predicate}
DURATION time interval
EVERY time span e

Figure: Query Template

C. Queries

We believe that declarative queries are the preferred way of interacting with a sensor network. Rather than deploying application-specific procedural code expressed in a Turing-complete programming language, we believe that sensor network applications are naturally data-driven, and thus we can abstract the functionality of a large class of applications into a common interface of expressive queries.

Our query template has the obvious semantics: The SELECT clause specifies attributes and aggregates from sensor records, the FROM clause specifies the distributed relation of sensor type, the WHERE clause filters sensor records by a predicate, the GROUP BY clause classifies sensor records into different partitions according to some attributes, and the HAVING clause eliminates groups by a predicate. Note that it is possible to have join queries by specifying several relations in the FROM clause.

III. SIMULATION, DESIGN & DEVELOPMENT

A. Simulation Environment

Simulation can provide a way to study system design alternatives in a controlled environment.

Simulation based testing is a very convenient testing environment. A single PC can simulate hundred of nodes that can be easily deployed and redeployed through configuration files. Simulation also provides access to internal state of the nodes. But the challenge for simulation is the amount of interaction with node's environment. Wireless sensor applications often have tens of components (radio, timer, sensor etc.) that a simulator need to model. A WSN simulation model must contain the required entities for a correct representation. These include communication components (motes and gateways), the wireless network interconnecting these elements, sensor entities and, finally, the surrounding environment. All should e configurable, act accordingly and yield reliable results.

Simulation Model of Wireless Sensor Networks:

To simulate wireless sensor networks, the simulation model is shown below:

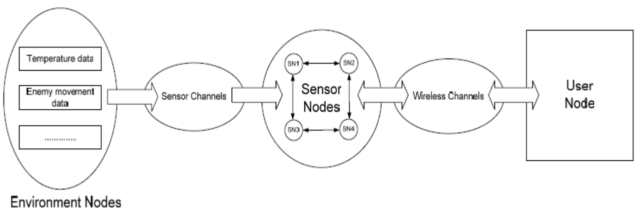

Fig. 3.1: The simulation model of wireless sensor networks

The simulation model of wireless sensor networks is generally separated into four main models; environment, sensor node, user node and communication models.

Environment model: This model uses environment nodes to define the physical environment in the simulation. For example, in the enemy surveillance application, this model defines environment nodes moving randomly as enemy moving in the battle field or, in the environment temperature detection application; this model defines static environment nodes to generate the environment temperature in each area.

IV. QUERY OPTIMIZATION

This section discusses our assumptions and provides a high level overview of our query optimizer. **Assumptions.** In this paper, we focus on optimization of a *single query* (*Q*) which contains *p* predicates PQ = *f*P1*; P*2*; :::; Ppg* and involves *n* nodes, NQ = *f*N1*;N*2*; :::;Nng*1. Similar to well-known query processing engines (e.g., TinyDB [14]), we assume that nodes are stable (i.e., they do not move or fail) and each node has *s* sensors, S = *f*S1*; S*2*; :::; Ssg*. For each sensor *Si 2* S, a node maintains a histogram to capture the distribution of reading values generated by *Si*.

Overview: In our study, the query optimizer generates a set of query plans with respect to *Q*, denoted as QQ = *fQP*1*;QP*2*; :::;QPmg*, where *m* is the total number of query plans generated. A query plan *QP 2* QQ specifies how to execute the query in the sensor network instead of just specifying how to sample sensors in a node. There are two stages in our query optimization process:

A. Classification Stage

Determine if metadata should be collected by evaluating each query plan within QQ. Based on this evaluation, the query optimizer chooses *QPmin 2* QQ that utilizes minimal energy. If *QPmin* requires no metadata collection, it indicates that the extra cost of metadata collection may not be justifiable for the energy saving obtainable from finding a better query plan. Thus, *QPmin* is distributed to each node *N 2* NQ, which in accordance with *QPmin* samples its sensors and routes the results back to the AP. On the other hand, if *QPmin* dictates that metadata collection is necessary, our query optimizer enters the second stage as described next.

B. Refinement Stage

Collect metadata to re-evaluate each query plan in QQ choosing *QP0 min*. Finally, *QP0 min* is distributed to each node *N 2* NQ, which samples its sensors and routes back the desired tuples to the AP in accordance with *QP0min*.

V. QUERY OPTIMIZATION TECHNIQUES

Base station has the interface for the queries input in wireless sensor network. Thus we can use the base station as a filter to reduce duplicate data access from the sensor network. When a user submits a continuous monitoring query like "SELECT nodeid, temperature FROM sensors WHERE temperature>50 SAMPLE PERIOD 2s" to the base station, then query is injected into the sensor network and then each sensor node periodically sends its readings towards the base station if the readings satisfy the condition of the query. The sample period of a query specifies the time interval between the re-evaluation of the query. As the number of queries that are running in the sensor network increases, the energy consumption of the sensor network increases. This is because each sensor node has to send its readings towards the base station.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Thus in order to reduce the energy consumption of the sensor network, it is important to reduce the number of monitoring queries injected into the network. The query optimization techniques are:

A. *Merging of queries [10]*

In practice, monitoring queries often have similar expression among them like SELECT, FROM, WHERE etc. Instead of running each query independently, if we merge queries to answer all queries, the overall energy consumption of the sensor network can be reduces because duplicate data request will be eliminated.

Consider the following two queries:

i) "select nodeid, temp where 25<temp<80 epoch duration 2 sec"
ii) "select nodeid, temp where 100<temp<150 epoch duration 4 sec"

To ensure correctness all the data requested by Query (i) and Query (ii) must be requested by query Q" and the result of both queries (i) and (ii) should be obtainable from result of Q". The attributed and predicates of Q" should be union, while the epoch duration should be Greatest Common Divisor. Hence Q" should be: "select light where 25<temp<150 epoch duration 2 sec". Using the Greedy Query Insertion Algorithm [10] we can merge the two or more queries. In the case of Greedy query Insertion Algorithm Benefit [22] metrics is calculated. i.e. If Benefit > 0 then we merge the Queries.

B. *By using results of existing queries (Query Rewriting)*

If we can reuse the results of existing queries to answer other new queries, the overall energy consumption of the sensor network can be reduced because duplicate data requests can be eliminated. For example, suppose that the following two monitoring queries q_1 and q_2 are currently running in a sensor network.

q_1=SELECT nodeid, temp FROM sensors WHERE temp > 35 SAMPLE PERIOD 2s.

q_2= SELECT nodeid, light FROM sensors WHERE light > 200 SAMPLE PERIOD 4s.

If a new query q_{new} is submitted to the base station.

q_{new}= SELECT nodeid, temp, light FROM sensors WHERE temp >50 AND
light > 200 SAMPLE PERIOD 8s.

q" new = SELECT nodeid, temp, light FROM q1, q2 WHERE q1.nodeid = q2.nodeid AND temp > 50 AND light > 300 SAMPLE PERIOD 8s

We can answer q_{new} using the results of q_1 and q_2 without injecting q_{new} into the sensor network. In other words, we can rewrite q" new using q_1 and q_2 to reuse the results of those two queries.

VI. RESULTS, PERFORMANCE EVALUATION

A. *Performance Evaluation*

In this section, we evaluate the effectiveness of the query rewriting method. We simulated the sensor network comprised of 30 static nodes uniformly distributed in the fixed area (500m X500m).We have implemented the query rewriting method in TinyDB, which is one of the most famous query processing systems for sensor networks. We have simulated a sensor network using TOSSIM [9], an event-driven simulator for TinyOS based sensor networks. In the experiment, we used dummy data obtained from [17]. The data obtained from [17] contain about 2.3million sensor readings collected from 54 Mica 2 Dot sensor nodes deployed in the Intel Berkeley Research lab between February 28th and April 5th, 2004. Each sensor reading has the information about date, time, temperature, humidity, light, voltage etc.

With this setting, we have compared the performance of the following three methods for running multiple monitoring queries in the sensor network:

(1) Independent Queries: each query is independently executed in the sensor network
(2) Merge: queries are optimized using the merge-based method [4].

(3) Query Rewriting and Merge: query rewriting method is used together with Merge. In this method, if a new query can be rewritten using currently running queries, the new query is rewritten by Query Rewriting and evaluated at the base station. If not, the new query is optimized using merge. For the performance measure, we have counted the number of sensor readings transmitted from sensor nodes to the base station. For each query injected into the sensor network, each sensor node sends its sensor reading towards the base station whenever its sensor reading satisfies the condition of the query. Note that the energy consumption of the sensor network increases as the number of transmissions increases. Since the performance of our method does not depend on network conditions such as network topology, network size or routing protocols, we have not considered these factors in the experiments.

Fig.6.1. shows the number of sensor readings transmitted from sensor nodes to the base station when Query-Set is executed using the three methods respectively. In this experiment, we have counted the number of sensor readings transmitted as the number of sensor readings produced increases. In the case of Query-Set, combination of Query Rewriting and Merge significantly reduced the number of transmissions compared to the other methods. The number of transmissions was reduced in combination of Query rewriting and Merge compared to the other methods.

When a total of 150000 sensor readings were produced, 414346 and 209322 sensor readings were transmitted in Independent and Merge respectively, while only 178786 sensor readings were transmitted in Combination of Query Rewriting and Merge. In this case, Combination of Query Rewriting and Merge reduced the number of transmissions by 56.86% and 14.6% compared to Independent and Merge, respectively. This means that even when there are only a small number of queries rewritten, Combination of Query Rewriting and Merge can benefit from Merge. From the experimental results, we can confirm that the simulated method can help reduce the number of transmissions for running multiple monitoring queries. Obviously, the performance gain of the simulated method will increase as the number of queries rewritten increases.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

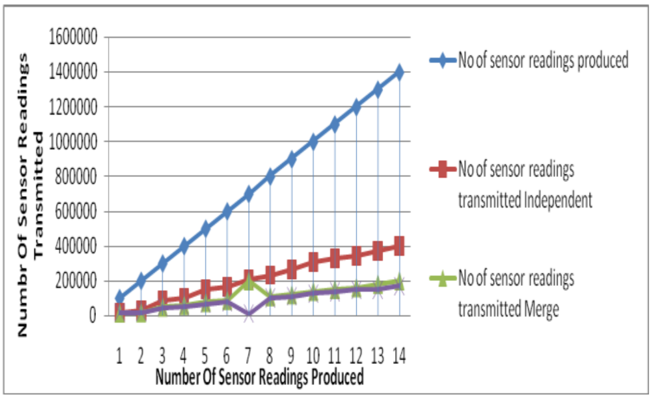

Fig. 6.1.No of sensor readings transmitted and received

B. Performance Evaluation of Cost (Energy Consumption)

Effectiveness of Classification: The classification component in our query optimizer determines whether on-demand metadata collection is needed. While up-to-date metadata is necessary to accurately estimate the energy cost of query plans, there is energy overhead for collecting metadata. Thus, only *merging, rewriting queries* which consume significant energy performing sensor sampling and result collection, may trade the overhead of metadata collection for an actual energy saving. There is not an absolute threshold to determine whether a query is heavy or light. For a given query, the following factors significantly impact its total energy consumption: *the number of reporting's* and *the unit energy cost of each sensor sampling*.

On the other hand, the higher unit energy cost of each sensor sampling the higher the total cost. In addition to the above two factors, *the freshness of metadata* a query optimizer currently has also plays a role on classification. If the metadata is fresh, the estimates of query plans are more accurate and thus may not require on-demand metadata collection. Therefore, for each of the three above-mentioned factors, we conducted an experiment to determine its impact on our query optimizer.

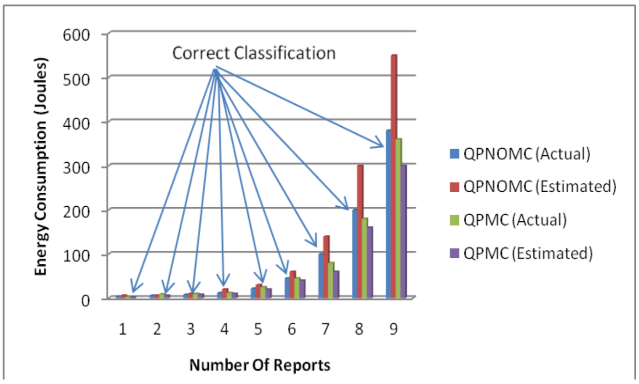

Fig.6.2 Impact of Reporting

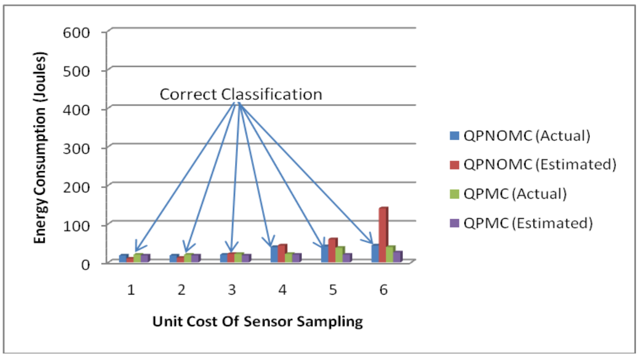

Fig.6.3 Impact of Sensor Sampling

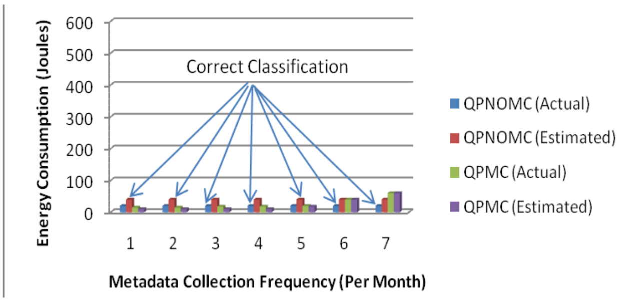

Fig.6.4 Impact of Metadata Freshness

Effectiveness of Query Optimizer: In this section, we conducted experiments to evaluate if our query optimizer chooses one of the most energy-efficient query plan for execution. To achieve this, we modified our simulator to execute each query plan in the query space of a given query *Q*. For each query plan, we obtain both of the estimated and actual energy costs.

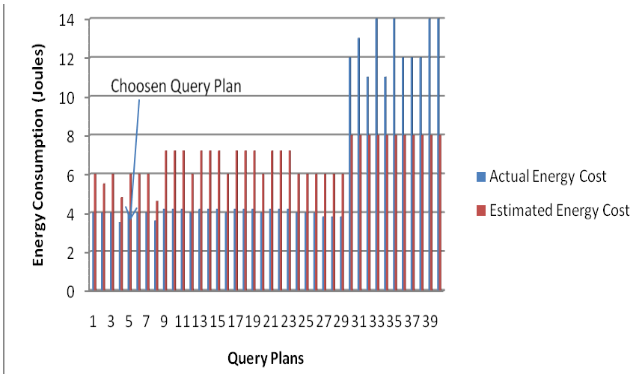

Fig.6.5 Simple Query

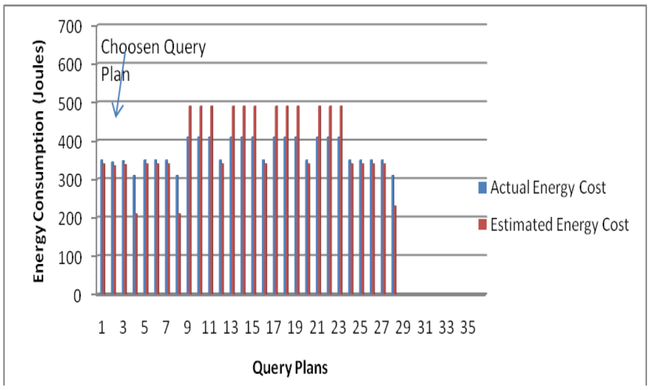

Fig.6.6 Rewriting Queries

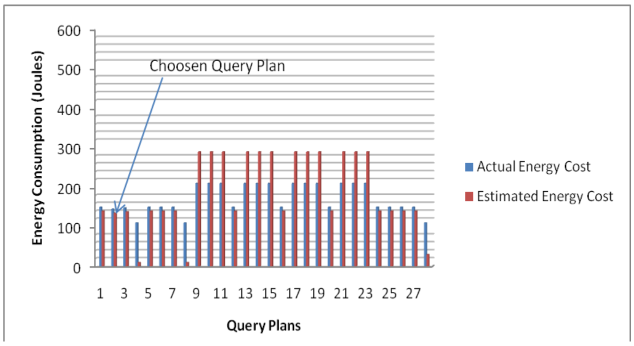

Fig.6.7 Merging Queries

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

VII. CONCLUSION AND FUTURE WORK

We examine the design of a query optimizer for WSNs. Since sensor nodes have limited energy capacity, it is important to minimize the energy consumption of sensor nodes to prolong the lifetime of a sensor network. Unlike existing approaches which only define different sensor sampling orders, our approach optimizes on routing, sensing and data/ metadata collection. A salient feature in our approach is the estimation and tradeoff of the metadata collection cost for the potential gain in more accurately estimated query plans. In query rewriting algorithm when a new monitoring query is submitted to the base station, the rewriting query method rewrites the new query using currently running queries if possible. The rewritten query is then evaluated at the base station using the results of other queries without being injected into the sensor network. As a result, the energy consumption of the sensor network is reduced. The simulated method shows that this method reduces the number of sensor readings transmitted from sensor nodes to the base station, resulting in lower energy consumption as compared to independent and Merge.

In this work base station optimization is performed. For the future work network optimization can be done at network level, multiple queries simultaneously. In further we can use individual sensor nodes for aggregation of data which can reduce the individual data packets.

REFERENCES

[1] Crossbow Technology Inc., http://www.xbow.com.
[2] Sirish Chandrasekaran Michael J. Franklin. Streaming Queries over Streaming Data. IEEE
[3] Madden, S., Franklin, M.: Fjording the stream: architecture for queries over streaming sensor data. In ICDE. (2002)
[4] Goel, A., Estrin, D.: Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. In SODA.(2003)
[5] A.Deshpande and S.Madden. MauveDB: supporting model-based user views in database systems. In SIGMOD, pages 73–84, 2006.
[6] S. Das, C. Perkins, and E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In INFOCOM 2000, pages 3–12. IEEE.
[7] A. Silberstein and J. Yang. Many-to-many aggregation for sensor networks.ICDE_2007.
[8] X. Yang, H. B. Lim, T. Ozsu, and K.-L. Tan. In-network execution of monitoring queries in sensor networks. In SIDMOD, 2007.
[9] P. Bonnet, J. Gehrke, and P. Seshadri. Towards sensor database systems. In Proceeding of the International Conference on Mobile Data Management, 2001. Perkins, C.: Ad hoc on demand distance vector (aodv) routing. (Internet Draft 1999, http://www.ietf.org/internet-drafts/draft-ietf-manetaodv-04.txt)
[10] Perkins, C., Bhagwat, P.: Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In SIGCOMM. (1994) 234–244
[11] Xu, Y., Bien, S., Mori, Y., Heidemann, J., Estrin, D.: Topology control protocols to conserve energy in wireless ad hoc networks. In TR6, UCLA/CENS (2003)
[12] Xu, Y., Heidemann, J., Estrin, D.: Geography-informed energy conservation for ad hoc routing. In MOBICOM. (2001) 70–84
[13] Ye, W., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In INFOCOM. (2002) 1567–157610.
[14] Madden, S., Hellerstein, J.: Distributing queries over low-power wireless sensor networks. In SIGMOD. (2002)
[15] Madden, S., Szewczyk, R., Franklin, M., Culler, D.: Supporting aggregate queries over ad-hoc sensor networks. In WMCSA. (2002)

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)