CS848 Presentation

Query Processing for Sensor Network

Present by:
Yang Sun
University of Waterloo
March 05, 2006

1 Background of Sensor Network:

1.1 Sensors and Sensor Network

Sensor network is made up of a network of sensor nodes. The purpose of the sensor network is to use multiple sensors to cover an geographic area to automatically monitor, or detect changes in the environment.

Each sensor node is a small device that has the ability to sense the environment. Examples of sensor could be a camera, an earthquake detector, or sensor that senses temperature, moisture etc. Additionally, sensors also possess the computations and communication capabilities. There are many different types of sensors. For example, some sensors are permanently attached to power sources, while most other sensors rely on a battery. Furthermore, some sensors are connected to the LAN network, while other sensors use radio communications. This paper focuses mainly on issues related to sensor that are powered by battery and uses multi-hop radio communication.

1.2 Resources Constraints of Sensor Node

Communication:

The wireless network provides very limited quality of communications service, has latency with high variance, has limited bandwidth, and frequently drops packets.

Power Consumption:

Sensor nodes considered in this paper has limited power supply. Since a lot of sensor networks needs to run a long period of time with little human maintenance, it is necessary for the sensor node to conserve power.

Computation:

Sensor nodes have limited computing power and memory sizes. This restricts the amount of intermediate result a node can hold, also the type of data processing algorithm on a sensor node.

Uncertainty in sensor readings:

Signals detected at physical sensors might have errors. Malfunction sensors might repeatedly generate false signals, also there could be bias caused by the placement of the sensor.

1.3 Goals of Query Processing for Sensor Network

The paper designed a query layer for wireless sensor network. There are three goals for the query processing. First, the paper chooses declarative queries for sensor network interactions. This allows user to issue queries without knowing how the results are generated and processed. The declarative query essentially acts as a ninterface. The user is able to access the data with the interface, while at the same time, the catalog management, query optimization, and query processing techniques abstract the user from the physical details of contacting the relevant sensors, processing the sensor data, and sending the results to the user.

The second goal is to address the resource constraints of the sensor network, for example, the energy and bandwidth at each node. One observation is that for a sensor node, the cost of sending data is always magnitude larger than data computation; therefore, to minimize energy consumption, it is essential to minimize the number of the packets sent through the network. A natural approach to the energy consumption problem is to simply perform aggregation the records at the node level, or eliminate irrelevant records.

The third goal is that the query processing must be able to dynamically adapt with requirements of different types of sensor network. As sensor networks might have different priority, the query processing must be able to address those priorities accordingly.

2. Simple Query Processing in a Sensor Network

2.1 Sensor nodes

A sensor network consists of a large number of sensors, or nodes. Each node is only wirelessly connected with its neighbor nodes. The nodes use a multihop routing protocol to communicate with distant node.

A special type of node is called a gateway node. Gateway node is connected to component outside of a sensor network through long range communication such as LAN. All communication with the user within the sensor network goes through the gateway node.

2.2 Data in a Sensor Network

Recall that all data in a sensor network is coming from a sensor. Therefore, each sensor is essentially a data source with its own id, location, and sensor type, A data record generated by a sensor has the information of the sensor, as well as value of the reading. Sensor type specifies the schema of a record. Therefore, records from different sensors with the same sensor type form a distributed database table. A sensor network is then considered a distributed database with multiple tables, each formed by a different type of sensors.

As mentioned above, each sensor contains noise. One way to reduce the noise is by fusing the results of multiple data. For example, instead of asking a particular sensor the value it senses, asks a group of sensors in the region to compute the average value of the reading.

2.3 Typical Query

The paper uses a descriptive query to interact with the network. A descriptive query is that the query is not given in terms of how to find data, but give criteria for the desired data. SQL is an example of descriptive query language.

The query in the sensor networks is very similar to SQL, with a noticeable exception. The sensor networks needs to support long running, periodic queries. To accommodate this requirement, the query includes a duration clause, which is used to specify the lifetime of the query. For example, a query might be given a life time from 2 o'clock to 5 o'clock. Another clause, Every clause, is often used in conjunction with the Duration clause. Every clause specifies the rate of the answer. So every 10 could specify the sensors report its finding once every 10 seconds, until the maximum life time of the query.

2.4 Simple Aggregate Query Processing

Definition of Aggregate Query

A simple aggregate query is an aggregate query without Group By, or Having clause. Simple aggregate query is a very popular class of queries in sensor networks. The following paragraphs describe the query processing strategies for the simple aggregate query.

Leader Node

To process a simple aggregate query, namely the delivery of records from distributed nodes and the computation of aggregation. Since all data resides on the nodes, all the data has to go to a central destination node for aggregation. This central node is called the leader node.

An important issue is to conserve energy of nodes. To do so, it is essential to reduce the communication cost, or equivalently, reduce the number of packets send. The following three approaches summarize different type of communications.

Direct Delivery

This is the do nothing approach. Each source sensor node sends the data record in a packet to the leader. The packet is forward in a multi-hop routing protocol. Computation is only performed at the leader node.

Packet Merging:

The cost of sending multiple small packets is much larger than the cost of sending a single big packet. This is because for each send, there is overhead such as reserve the channel and headers of multiple packets. One observation is that the size of the packets send are usually small, therefore, it is much more efficient to merge several records into a larger record.

Partial Aggregation:

For distributive and algebraic aggregation operators such as min and max, the computation of the aggregation can be pushed to the intermittent nodes. For example, intermediate node receiving multiple records can simply compute and forward the result to the destination.

Issues introduced by Packet Merging and Partial Aggregation:

Now the nodes need to be coordinated within a query plan, so that a node n can decide that which nodes (if any) are going to route data packet through it, i.e. the routing problem. Another issue is synchronization issue. Node n must know how long to wait for the nodes that route through it before sending the message to the next hop.

Routing problem

Depends on whether the aggregate operator is duplicate sensitive

For duplicate sensitive aggregator operator, such as sum and average, the data packet must be sent only once. A simple solution is a spanning tree, which guarantees a loop free, single path routing. On the other hand, if the aggregate operator is duplicate insensitive such as Max and Min, it is possible to send the same record multiple times through different paths. A communication structure for this case is DAG rooted at the leader.

Synchronization problem:
Naïve approach:
Beginning of a round, each sensor has a timer, and waits for a special waiting time for data packets from its children. For node n, the length of the timer equals depth(n)*timeslot, where timeslot is a predefined number.

Problem:
Hard to find a appropriate time slot, hard to recomputed time out value once a failure occurs that changes the depth of the network, expensive to notify large number of nodes. Time can never be completely synchronized among different node.

Approach 2:

- Use history data to infer future behavior.
- if p receives data from q, it expects to receive it from future
- if p does not receive it in the future and p recovers from timer by timing out
- q can also notify p if its parent prediction is wrong

3 General Query Plan

Previous sections have mentioned methods that minimize energy consumption, as well as synchronization and routing for simple aggregate query. In this section, we introduce the idea of general query plan.

Query Plan:

- can have multiple level of aggregation, i.e first compute average value for each group, then find the group with the minimum average value
- decide how much computation is pushed into the network i.e partial aggregation
- how to execute the query
- how to coordinates the relevant sensors
- made up of multiple flow blocks

Flow Block:

- task is to collect data from relevant sensor nodes
- perform computation at the destination or internal nodes
- specified by a set of source nodes, a leader selection policy, the routing structure, computation the block should perform

Query Optimizer:

- determine the exact number of flow blocks and interactions between them

Query Optimizer for Complex Aggregate Query with Group By and Having Clauses

Group By and Having

- plan 1, create a flow block for each group
- plan 2, create a flow block that is shared by multiple group

Plan decision

- depends on the overlap of the distribution of the physical location of the sensors that belong to different groups
- if different groups does not overlap and sensors within a group is close by, then use plan 1
- if sensors from different groups are spatially interspersed, then plan 2

Join Operator

- plan depends on the reduction of result size
- if reduction increases result size, then it is better for the leader to simply forward
- the data
- if reduction decreases the result size, then it is better to perform the join at the leader node

References:

Yong Yao, JUohannes Gehrke, Query Processing for Sensor Networks.

Jeremy Elson and Deborah Estrin, Sensor Network: A bridge to the physical world