

# Decentralizing Query Processing in Sensor Networks

Ross Rosemark, Wang-Chien Lee  
Department Of Computer Science and Engineering  
The Pennsylvania State University  
University Park, PA 16802, USA  
{rosemark, wlee}@cse.psu.edu

## Abstract

Recent research has led to the advent of software systems capable of performing query processing in sensor networks. They perform query processing in a sensor network by constructing a routing tree rooted at an access point (i.e., a base station), where the queries are submitted, parsed, and optimized. This approach to query processing is centralized in nature. Performing query optimization at a single node (base station) does not generate efficient query plans, and requires each node to report metadata to the access point. In addition, the routing tree infrastructure inefficiently aggregates data packets. To address these issues, this paper proposes several decentralized query processing systems that utilize sensor node's innate spatial and semantic characteristics. Experimental results conclude that decentralizing query processing significantly reduces energy costs. In addition, experimental results show that the spatial and semantic properties of nodes are influential in designing a decentralized query processing system.

## 1 Introduction

Recent research has led to the advent of software systems capable of performing query processing in sensor networks [11, 14, 10, 9]. Unlike other environments, query processing systems designed for sensor networks must incorporate energy awareness into the system to extend the lifetime of the sensor nodes and network. Existing query processors, e.g. TinyDB and Cougar, have met these requirements by pushing operations such as selection and aggregation within the sensor network in order to reduce communication costs [11]. Their approach is to construct a routing tree rooted at an access point (i.e., a base station), where queries are submitted, parsed, and optimized. The optimized execution plan obtained at the access point are then disseminated via the routing tree into the network for processing. Finally, the query results flow back to the access point also

via the tree. Queries for sensor networks typically specify a sample rate. There are usually different sensors in each sensor node. Query results are generated by sampling node's sensors in the order defined by the query plan. To facilitate query optimization, metadata periodically will be collected (via the routing tree) by the access point in aggregated form from all nodes within the sensor network. This kind of query processing systems is referred to as the Centralized Query Processing System (CQPS), since it performs query optimization in a centralized manner (i.e. at the access point). An example of the CQPS is illustrated in Figure 1. As shown, this systems's infrastructure is typically a routing tree.

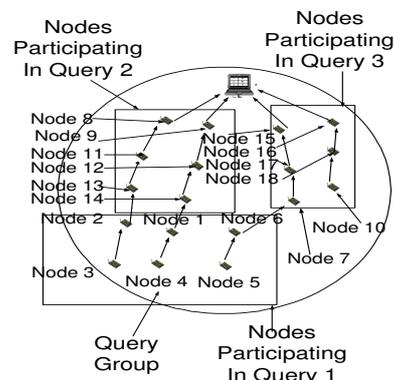


Figure 1. Centralized Query Processing System

In this paper, we argue that query processors in sensor networks should continue the trend of pushing expensive computations within the sensor network. The above mentioned centralized approach to query optimization and spanning tree construction has some deficiencies with respect to query optimization and routing. To derive an execution plan utilizing this approach, the access point must first collect metadata from all nodes within the sensor network requir-

ing excessive energy consumption (we argue that a subset of nodes in the sensor network can avoid sending their metadata). Second, the access point derives a single energy efficient query plan by utilizing the metadata collected from the sensor network. Such a centralized approach disallows two sets of sensor nodes that have different data profiles from executing separate query plans. This problem exists since the access point generates the query plan based on the aggregated metadata it has collected from all nodes in the sensor network. Since metadata (gathered by the access point) is in aggregated form, it may not precisely represent every nodes local metadata. As a result query plans generated by the access point will be in-efficient requiring nodes to utilize excessive energy sampling data from its sensors. We argue that multiple nodes should perform query optimization within the sensor network. Finally, the access point distributes the execution plan to all nodes within the sensor network, who will in turn execute the query and return the results to the access point via the routing tree. A routing tree is inefficient since aggregation can only transpire between two childrens data packets at an intersecting parent. For instance, in Figure 1, multiple paths are utilized to route data packets from nodes 1 and 2 to the AP (nodes 1 and 2 utilize paths 1,14,12,9 and 2,13,11,8 respectively). A better solution would perform aggregation near the nodes that generated the data packets. In this example a better solution is as follows: 1) node 1 is designated to perform aggregation for nodes 1-6; 2) node 2 routes data packets to node 1; 3) node 1 performs aggregation; 4) node 1 routes aggregated data to AP. The concept of designating nodes to perform aggregation has been discussed in the context of clustering research [15, 12]. This research has shown that clusters will alleviate the energy associated with routing. In this paper we expound on existing research to intelligently adapt the network infrastructure based on the nature of queries active in the sensor network.

This paper introduces a decentralized infrastructure to support query processing in sensor networks – our main contribution. This infrastructure will distribute query optimization and query routing within the sensor network in order to alleviate the energy consumption required to: 1) collect metadata; 2) perform execution plans at sensor nodes; 3) route results from sensor nodes to the access point.

In this paper, we will first introduce a fully decentralized query processing system that generates efficient query plans and alleviates the energy costs associated with collecting metadata. Second, we will present a query processing system that utilizes node’s innate spatial properties in order to alleviate the energy costs associated with routing data packets. Third, we will use the innate semantic characteristics of nodes to derive a query processing system that generates query plans that are more efficient in terms

of energy consumption (relative to the CQPS). Finally, this paper will merge the spatial-based and semantic-based approaches into a hybrid approach.

A performance evaluation comparing the CQPS to our proposed query processors on the basis of energy consumption is conducted through simulation. Experimental results conclude that decentralizing query processing significantly reduces energy costs. In addition, experimental results show that the spatial and semantic properties of nodes are influential in designing a decentralized query processing system.

The rest of this paper is organized as follows. In Section 2, we review existing approaches to support query processing, and discuss existing clustering algorithms. We then introduce in Section 3 multiple decentralized query processing systems. In Section 4, we conduct simulations to compare and analyze the energy consumption of the CQPS and our proposals.

## 2 Background

In this section, we review protocols to support query processing and clustering in sensor networks.

### 2.1 Query Processing

As briefly introduced, in the CQPS, queries are optimized at the access point. The access point will then disseminate the chosen query plans by broadcasting a *routing tree build request* (RTBR) to all adjacent nodes that are within the range of its radio. The RTBR message broadcast by the access point will contain the binary query plan, a level specifying the number of hops from the access point (initially the level is zero) and a node identifier specifying the sender of the message. Adjacent nodes hearing the build request will choose the sender as their parent, assign their level as one more than the level specified in the message, and save to memory this query plan if applicable. They will then continue to disseminate a message specifying the query plan, their level and identifier to all adjacent nodes, which will in turn continue this process until all nodes within the sensor network have been assigned a level in the routing tree. Utilizing this infrastructure, nodes send all query results to their selected parent, who in turn will forward the message to their parent. This process will continue until the message has been received by the access point. Ensuring that node failure does not break the connectivity of a path from a sensor node to the access point, the RTBR will be periodically re-broadcasted by the access point. Nodes participating in a query will periodically wake up from a sleep state, collect data specified by the query and route the query results to a adjacent node that is one hop closer to the access point and is within radio range. The sample rate in which query results are generated is defined in the user’s query.

Query results are generated by sampling node’s sensors in the order defined by the query plan. To facilitate query optimization, periodically metadata will be collected (via the routing tree) by the access point in aggregated form from all nodes within the sensor network.

## 2.2 Clustering

Recently many clustering algorithms have been proposed to create and maintain spatial clusters in a sensor network. This section reviews some well established algorithms along with the research issues they address. Spatial clustering algorithms are reviewed because aspects of these algorithms are utilized by our Spatial Query Processing Systems’s routing infrastructure (discussed later).

In [2], Bandyopadhyay et al. described a distributed hierarchical clustering algorithm that is able to reduce the communication costs associated with sensor nodes reporting gathered information to the access point.

Dimensions [5] creates a clustering hierarchy that utilizes wavelets to encode sensor data in order to reduce the energy costs associated with delivering results to users. Dimensions has done a preliminary analysis showing that if the data contained at nodes in the sensor network exhibit high spatial and temporal correlations the reconstruction error of the wavelet will be reduced.

In [4], Estrin et al. argues that global or distributed algorithms will not scale well in terms of energy consumption as the size of the sensor network increases. This paper proposes the utilization of clusters as a means of sensor nodes efficiently coordinating their local interactions in order to satisfy a global objective.

Leach [7] designs an approach that periodically reselects cluster heads in order to minimize the variance between sensor nodes time of death (the time a node fails).

In [15], a distributed clustering protocol capable of providing fairly uniformly distribution of cluster heads (in comparison to other clustering algorithms) is proposed. Ensuring that clusters meet these requirements is desirable since they uniformly distribute the workload (energy consumption) required by each cluster heads. Our proposals also mitigate the workload across multiple nodes.

In [12], Pattem et al. clusters nodes that exhibit a high degree of spatial correlation for a sensed phenomenon in order to improve the compression of data packets. Through analytical derivations and simulations they have ascertained the optimal size of a cluster for a wide range for spatial correlations. The idea of incorporating spatial correlations of data in the cluster formation provides motivation for our Semantic Query Processing System (discussed later).

## 3 Query Processing Infrastructures

In this section, we discuss distributed query processing systems. For each approach, we assume nodes represent each attribute’s metadata with a separate histogram and nodes are static with respect to movement. In addition, we assume that queries are injected at a single access point and are long running in nature. In respect to metadata, the characteristics of each node’s histograms (number of buckets, size of buckets, distribution of values within a bucket) will be the same for each node. For each histogram, the number of buckets is 30 and the distribution of the values within each bucket is assumed to be uniform. The size of a bucket is dependent on the attribute the histogram is representing. In this paper we assume all query processing protocols utilize the same representation for metadata. To disambiguate terminologies, when discussing our proposed approaches this paper refers to clusters as Query Groups (QG)<sup>1</sup>.

### 3.1 Fully Distributed Query Processing System

A simple modification to the CQPS is to duplicate the query optimizer at every node (create N QGs such that N is the number of nodes in the sensor network (see Figure 2)). This approach was proposed to examine the upper bound

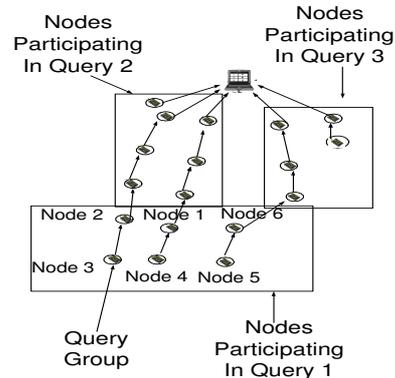


Figure 2. Fully Distributed Query Processing

on the number of nodes performing query optimization (the CQPS is the lower bound since 1 node performs query optimization). In this approach, queries received by the access point are disseminated throughout the sensor network. Nodes receiving the query will utilize their local query optimizer to generate a query plan. Each nodes query optimizer will determine the nodes query plan based only on the local metadata of the node. Each node will then perform the

<sup>1</sup>A QG is defined as a cluster of nodes in which the cluster heads performs query optimization.

query plan generated by its query optimizer at a predefined time and route the results to the access point via the routing tree. In this paper, we refer to this system as the Fully Distributed Query Processing System (FDQPS).

The advantage of this approach is that it creates efficient query plans while removing the need to perform metadata collection. Query optimizers will generate efficient query plans since each node performs query optimization based only on its local metadata. In addition, since query optimization is performed at each sensor node, the energy cost to disseminate metadata packets is alleviated (i.e. each node can suppress sending their metadata). The disadvantages of this approach is that 1) every node consumes energy in query optimization; 2) it does not address aggregation of data packets since the infrastructure is still based on the routing tree (discussed above).

### 3.2 Spatial Query Processing System

An alternative approach to query processing is to distribute query optimization to  $K$  nodes, such that  $K \leq$  number of nodes in the sensor network (see Figure 3). This

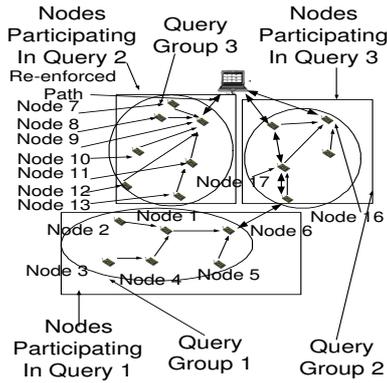


Figure 3. Spatial Query Processing System

approach was proposed to examine the query processing infrastructure when the number of nodes performing query optimization is between the lower bound (CQPS) and upper bound (FDQPS). In this approach, spatial clustering algorithms are utilized to generate QGs within the sensor network. The number of QGs derived by our query processing system is based on the radius of a QG. Radius refers to the maximum distance (in hops) between a non cluster head (NCH) node and its corresponding cluster head (CH) node. In this paper, we refer to this system as the Spatial Query Processing System (SQPS).

The pseudo-code in Algorithms 1 and 2 illustrate respectively how the SQPS's infrastructure is derived and how it processes queries. In this approach, the CH node in each

QG is responsible for collecting metadata from the sensor nodes in its QG along with performing query optimization and data aggregation. Our system strives to create a set of QGs that: 1) provide relatively uniform distribution of CH nodes; 2) choose CH nodes with high remaining energy; 3) choose CH nodes that are spatially close to the access point. Design aspects adapted from [4, 3, 2, 15] are utilized in our schemes. In addition, our query processing system provides means to perform intelligent adaptations of the query processing infrastructure based on the nature of queries active in the sensor network. In line 15 (Algorithms 1) we determine the QG a NCH node participates in based on the number of queries that both the NCH node and the QGs CH node are participating in. This increases the chances data packets generated by the NCH node are aggregated at its corresponding CH and ensures that query plans generated by a CH node is valid for a large set of nodes the CH nodes QG. In other words, given a NCH node  $N$  that is participating in a set of queries  $S_N$  and has heard from a set of CH nodes  $S_{CH}$ , node  $N$  will select the  $CH_i \in S_{CH}$  that has the greatest cardinality  $\|(S)_{CH_i} \cap (S)_N\|$ . If a NCH node participating in QG receives a message from a CH node with greater query pattern, the NCH node will join the new QG and cancel its participation in the old QG. To cancel participation in a QG, a NCH node will send a cancelParticipation message to the QGs CH node.

---

#### Algorithm 1 Pseudo-code to create the infrastructure (Spatial Query Processing System)

---

- 1: Set  $SN$  of length  $N$  represents the nodes in the sensor network
  - 2: The set  $QG_{1..NQG}, NQG \leq N$  represents the Query Groups in the sensor network.
  - 3: Network consists of a single access point  $AP_i$
  - 4:  $AP_i$  injects a Routing Tree Build Request (RTBR)
  - 5: The set of nodes  $AN \in N$  that hear the RTBR message will set a Promotion Timer (PT) based on their (current energy, level in routing tree, random timer).
  - 6: When node  $X_i$ 's  $\in AN$  PT expires the node  $X_i$  will:
  - 7: **if**  $X_i \in QG_{1..N}$  **then**
  - 8:   Maintain as a NCH node
  - 9: **else**
  - 10:   Create a new Query Group  $QG_{X_i}$  in which you are the CH node ( $X_i \in QG_{X_i}$ ).
  - 11:   Node  $X_i$  send a pathEnforeMsg to  $AP_i$  to re-enforce the path between it and the  $AP_i$ .
  - 12:   Node  $X_i$  send out a clusterHeadNotificationMsg to notify all nodes within  $M$  hops of node  $X_i$  that it is a CH node.
  - 13: **end if**
  - 14: The set of NCH node's  $P \in SN$  hearing node  $X_i$ 's clusterHeadNotificationMsg will:
  - 15: **if** The Query Pattern between node  $P_i$  and  $X_i$  is the greatest (By default all nodes have a Query Pattern of 0) **then**
  - 16:   Node  $P_i$  will join Query Group  $QG_{X_i}$
  - 17: **end if**
  - 18: To maintain the infrastructure periodically restart algorithm from step 4.
- 

The first advantage of this approach is that it alleviates the costs associated with collecting metadata (compared to

---

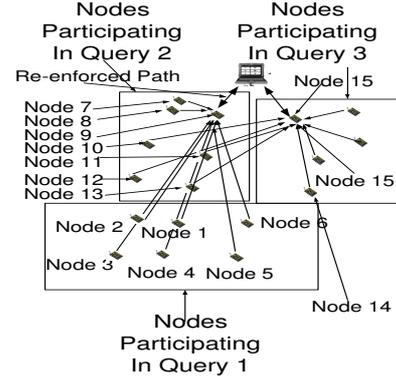
**Algorithm 2** Pseudo-code to process queries (Spatial Query Processing System)

- 1: Queries injected by the user at the access point will be forwarded to all CH nodes via the reinforced paths.
  - 2: Queries received by a CH node  $X_i$  will perform query optimization based on the metadata it has collected from nodes in  $QG_{X_i}$ .
  - 3: CH node  $X_i$  will disseminate the query plan to all nodes in  $QG_{X_i}$ .
  - 4: Nodes receiving the query plan will execute it at a predefined time and return aggregated results to their corresponding CH node.
  - 5: When all data packets have been received by a CH node, it will forward the received data packets in aggregated form to the access point
- 

the CQPS). Less energy is required to collect metadata since CH nodes are omitted from sending their metadata packets (CH nodes do not send metadata packets since they perform query optimization). The second advantage of this approach is that it alleviates on average the costs associated with performing query optimization (compared to the FDQPS). Query optimization is reduced on average since fewer nodes perform query optimization (only CH nodes perform query optimization). The third advantage of this approach is that it provides on average more efficient query plans (compared to the CQPS). Query plans on average are more efficient since a smaller subset of nodes define the metadata utilized by a query optimizer (discussed above). The final advantage of this approach is that it performs more efficient aggregation of data packets (compared to the CQPS and the FDQPS). To clarify, since each NCH node chooses a QG based on the greatest query pattern, data generated by the NCH has a high probability of being aggregated at the CH node. The disadvantages of this approach are 1) query plans are still not guaranteed to be efficient for all nodes in a QG 2) additional energy is required to create and maintain the routing infrastructure (routing tree in addition to QGs).

### 3.2.1 Semantic Query Processing System

Another approach to query processing is to utilize nodes innate semantic and temporal characteristics [16, 6] in the design of the query processing infrastructure. In this approach sensor nodes with similar metadata will be formed into  $K$  QGs ( $K \leq$  number of nodes in the sensor network). This approach was proposed to examine the energy consumption of the sensor network when semantic (rather than spatial) property of nodes are utilized to dictate the set of nodes to perform query optimization. The number of QGs derived by our clustering algorithm is based on the similarity (discussed later). In this paper, we refer to this system as the Semantic Query Processing System (SEQPS). An example of the infrastructure derived by the SEQPS is illustrated in Figure 4. Note that, in Figure 4, QGs are not denoted by a circle and each node is depicted as having direct communication with its corresponding CH. These changes are aesthetic in nature, and are made only to help illustrate the algorithm



**Figure 4. Semantic Query Processing System**

(i.e. this algorithm still employs multi-hop routing). In Figure 4, two QGs are defined, one consists of nodes 1-9 and the other consists of the remaining nodes, nodes 10 and 15 are CHs.

The pseudo-code in Algorithms 3 and 2 illustrate respectively how the SEQPS's infrastructure is derived and how it processes queries. Note that, the SEQPS utilizes the same protocol as the SQPS to process queries. Our algorithm to form QGs strives to choose CH nodes that: 1) have similar metadata to the largest subset of nodes in the sensor network; 2) have high remaining energy; 3) are spatially close to the access point. To our knowledge, we are the first to utilize sensor networks innate semantic and temporal characteristics in a query processing system.

---

**Algorithm 3** Pseudo-code to create the infrastructure (Semantic Query Processing System)

- 1: Lines 1, 2, 3 and 4 of Algorithm 1 are equivalent in this algorithm.
  - 2: The set of nodes  $AN \in N$  that hear the RTBR message will flood a packet containing their metadata referred to as the advertiseMsg.
  - 3: Each node  $N_i \in AN$  will set a timer to allow their advertiseMsg to reach all nodes  $AN$ .
  - 4: **if** Metadata in an advertiseMsg is similar to node  $N_i$ 's metadata **then**
  - 5:   Node  $N_i$  will save info about the node  $X_y$  that initiated the advertiseMsg.
  - 6: **end if**
  - 7: When a node  $N_i$ 's advertiseTimer expires it will set a Promotion Timer based on: it's current level in the routing tree, remaining energy, number of nodes it has similar metadata with, random timer.
  - 8: At this stage the algorithm will perform the same as the SQPS except a node's clusterHeadNotificationMsg will contain the node id of all nodes that have similar metadata along with span the entire sensor network rather than  $M$  hops. In addition NCH nodes receiving the clusterHeadNotificationMsg will only choose CH nodes that have similar metadata. A NCH node and CH node have similar metadata if the NCH node's id is specified in the CH's clusterHeadNotificationMsg.
- 

Two well established mathematical approaches can be utilized to perform the similarity function (line 4 in Algorithm 3): Euclidean distance and earth movers distance

[13, 8]. To dynamically re-adapt the query processing infrastructure, buckets of each attributes histograms will be weighted based on query frequency. This skews similarity to give a higher weight to the buckets that have the greatest impact in determining query plans. To determine the weight of buckets the query optimizer will increase the weight of a bucket every time the query optimizer consults the bucket for its frequency and decrease the weight of corresponding buckets when a query expires.

This approach has the advantages of the SQPS along with generates more energy efficient query plans. Since QGs are comprised of nodes with similar metadata, the variance between the difference in nodes metadata within the QG will be reduced. As a result, query plans will be more energy efficient since the metadata utilized by a CH to generate a query plan will be consistent with the metadata of nodes in its QG. The disadvantages of this approach is 1) the energy cost to create and maintain the infrastructure is greater (compared to the SQPS); 2) aggregation is not performed as efficient (compared to the SQPS). The cost to create and maintain the infrastructure is greater since: 1) initially each node must send out an advertiseMsg to ascertain nodes with similar metadata; 2) advertiseMsgs and promotionMsgs must flood the entire network rather than M hops (M hops is the radius of a QG). Not limiting the spatial size of a QG also increases the distance between a NCH node and its corresponding CH resulting in an increase in the energy cost associated with collecting metadata and routing data packets. For instance, Figure 4, illustrates the infrastructure derived by the SEQPS. Assume, in this example, nodes 2, 1, 6 route data to CH 9 utilizing the independent paths [14,15,9], [13,11,9], [12,10,9] respectively. Utilizing this approach nodes 10,11,13,14,15 have to route additional packets (i.e. data packets from nodes 1, 2 and 6). A better approach would utilize both node's spatial and semantic properties. For example, create a QG that consists of nodes 1-6 and another that consists of nodes 7-9, assume node 1 and 9 are chosen as CH nodes. In this case, a data packet generated by nodes 1-6 will be sent to 1, which in turn aggregates the data packet with it's local data and sends it to the AP. This approach is favorable since nodes 14,15,9,12,10,9 do not route additional data packets.

### 3.2.2 Hybrid Query Processing System

The final approach to query processing (see Figure 5) is a hybrid of the SQPS and SEQPSs. In this approach, QGs will be comprised of nodes that have similar metadata, and that are spatially close to their corresponding CH node. This algorithm was proposed to examine the energy consumption of the sensor network as the combination of nodes semantic and spatial characteristics are utilized to determine the set of nodes to perform query optimization. In this paper, we

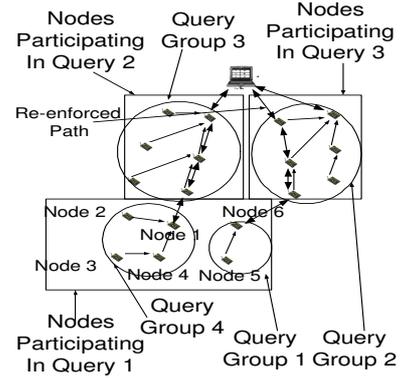


Figure 5. Hybrid Query Processing System

refer to this system as the Hybrid Query Processing System (HQPS).

Our algorithm to form QGs strives to choose CH nodes that: 1) have similar metadata to the largest subset of nodes within the CHs spatial vicinity; 2) have high remaining energy; 3) are spatially close to the access point; 4) are uniformly distributed within the sensor network. To achieve these goals the SEQPS algorithm will be modified to limit the range of each nodes advertiseMsg and clusterHeadNotificationMsg message to the subset of nodes that are within M (M is the radius of a QG) hops of the CH. Note that, the HQPS utilizes the same algorithm as the SQPS and SEQPS to perform query processing. This algorithm is defined in Algorithm 2. The advantages of this approach is that it provides both the benefits of the SEQPS and SQPS. This algorithms disadvantage is that it requires (like the SEQPS) additional energy to create and maintain the infrastructure.

## 4 Performance Evaluation

To evaluate our proposed approaches, experiments utilizing NS-2 network simulator were conducted. The goal of our experiments were to: 1) dictate that decentralizing query processing significantly reduces energy costs; 2) dictate that the spatial and semantic properties of nodes are influential in designing a decentralized query processing system.

### 4.1 Experimental Settings

In all experiments, the sensor network consisted of 50 static nodes distributed in a fixed area (900m \* 900m). The position of a node was dictated by 1) uniform distribution; 2) the coordinates (scaled accordingly) of a subset of nodes given by The Joint Institute for the Study of the Atmosphere and Ocean (JISAO) [1]. Data obtained from JISAO represents a node's metadata in our experiments.

In all experiments, simulations ran for 50000 milliseconds and were repeated 25 times to obtain the average results. Infrastructure maintenance is performed every 30000 milliseconds and metadata is collected every 10000 milliseconds. At the beginning of the simulation, multiple queries were injected at the access point. Each query would specify the type of query (MIN, MAX, AVERAGE, MEDIAN, SUM) along with a specification of data acquisition. In our experiments, each query specified sensor nodes to perform data acquisition by continuously polling and aggregating results from its sensor table every 50 milliseconds (event based queries are left for future work) [11]. The queries we consider are of the form [11]:

```
SELECT {aggExpr, attrs} FROM Sensors
WHERE {selPreds}
Group By {Attrs}
HAVING {havingPreds}
EPOCH DURATION i
```

When performing acquisition, our query optimizer will choose from a fixed set (typically 6) of query plans based on the selectivity of the query predicates and the energy cost associated with sensing the phenomenon. We assume nodes utilize 0.075 Joules to sample each sensor. We argue, that future sensors will utilize more energy because they will be more complex in nature (i.e. video cameras, thermal imaging units).

The metric chosen to evaluate the performance is **average dissipated energy**. This metric measures the ratio of the average energy consumed by all sensor nodes to the number of data packets received by the access point (data packets refer to packets that contain query results). Utilizing this metric we can obtain the energy query processing consumes at each sensor node, allowing us to indirectly extrapolate a sensor node's lifetime. Note that, that a small changes in average dissipated energy is significant since it only measures the energy required to deliver 1 data packet to the AP. In long running queries multiple packets (in excess of 1000) are generated and routed by each node. As such, a small change in average dissipated energy will amount to significant energy savings.

## 4.2 Simulation Results

In this section, we conduct experiments to compare the CQPS to our proposed query processing systems. First we perform experiments to obtain the optimal similarity of the SEQPS (under our experimental settings). Second, we perform experiments to obtain the optimal radius of the SQPS (under our experimental settings). Third, we conduct experiments to determine the optimal radius and similarity of the HQPS (under our experimental settings). Fourth, we

conduct experiments to determine the effects of query frequency. Finally, we perform experiments to evaluate the scalability of our approaches.

### 4.2.1 Similarity in SEQPS

The first experiment ascertained the optimal similarity of the SEQPS. As dictated by Figure 6 the optimal similarity is 0–9 for the uniform distribution and 10–19 for the JISAO distribution. Note that in the JISAO distribution only a fraction of energy reduction is obtained when the similarity is between 10–19. However, even a small variance in average dissipated energy is significant since it only reflects the energy required to deliver 1 packet to the AP. Also note, that in our settings when the similarity parameter is 0–9 the SEQPS (in addition to the HQPS) derives QGs that contain a single node (i.e. they derive the FDQPS). No QGs will be comprised of more than 1 node when the similarity parameter is 0–9 since no two nodes have similar metadata.

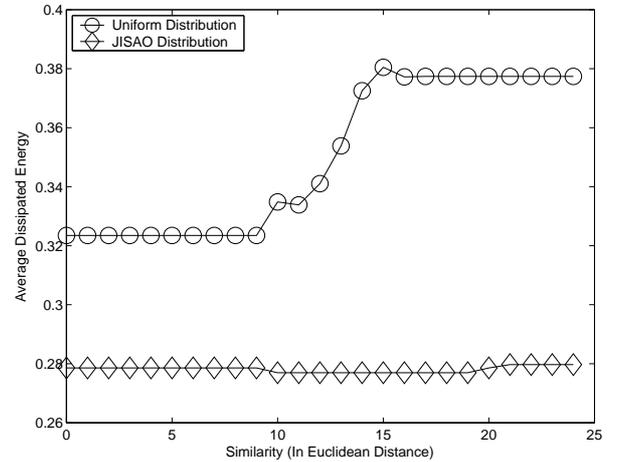


Figure 6. SEQPS optimal similarity

Again, in this experiment the optimal similarity is 0–9 and 10–18 for the uniform and JISAO distributions, respectively. The optimal similarity is 0–9 for the uniform distribution because it exhibits a low correlation between spatial distance and metadata similarity. As a result, nodes that have similar metadata may be spatially distant. To exemplify, the average distance (in hops) between two nodes that have similar metadata (i.e. similarity > 9) is 2.13 for the uniform distribution and 1.31 for the JISAO distribution. Decreasing the correlation between spatial distance and metadata similarity, increases the number of nodes on the path between a NCH and its corresponding CH. As discussed, this directly correlates to an increase in energy consumption as aggregation is performed in-inefficiently. Therefore, it can be inferred, that if nodes are uniformly distrib-

uted, the ability of the SEQPS to derive efficient query plans is outweighed by its inability to perform efficient aggregation. We will not further evaluate the SEQPS for the uniform distribution since optimally it derives the FDQPS. The SEQPS performs optimal in the JISAO distribution when the similarity parameter is 10–19 because it balances aggregation and query execution. For future experiments analyzing the SEQPS the optimal similarity (as dictated by Figure 6) is utilized.

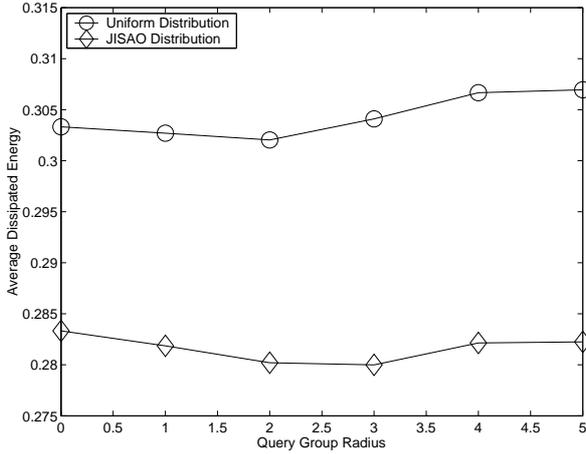
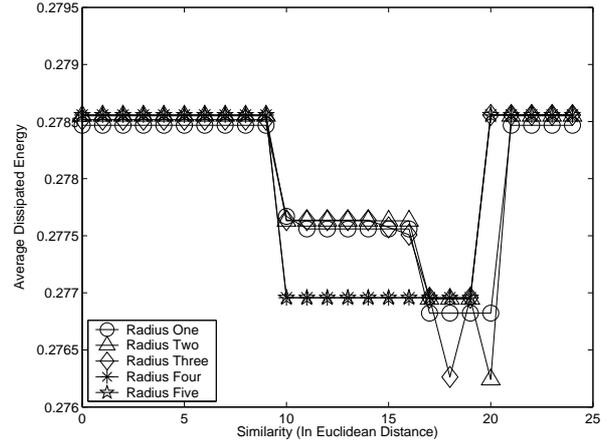


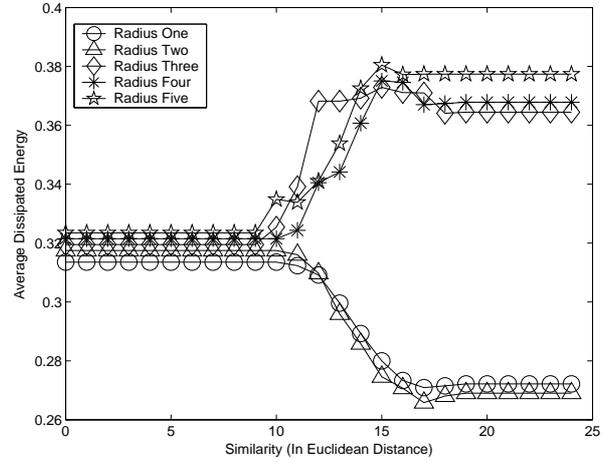
Figure 7. SQPS optimal radius

#### 4.2.2 Radius in SQPS

The second experiment ascertained the optimal radius of the SQPS. As dictated by Figure 7, the optimal radius is 2 and 3 for the uniform and JISAO distributions, respectively. In addition, this graph dictates that the SQPS utilizes more energy if the topology is uniformly distributed. The SQPS is not efficient when the topology is uniformly distributed because it does not derive efficient query plans. As discussed, nodes in the uniform distribution exhibit a low correlation between spatial distance and metadata similarity. A low correlation diversifies the data distribution of node's comprising a QG. As a result, the metadata (utilized by a CH to perform query optimization), does not accurately represent the metadata of the nodes in its QG (as discussed, this increases the energy associated with query execution). Therefore, it can be derived from this experiment that the Spatial Query Processing System will not be as efficient for topologies that exhibit a low correlation between spatial distance and metadata similarity. For future experiments analyzing the SQPS, the optimal radius (as dictated by Figure 7) will be utilized.



(a) JISAO Distribution



(b) Uniform Distribution

Figure 8. HQPS optimal parameters

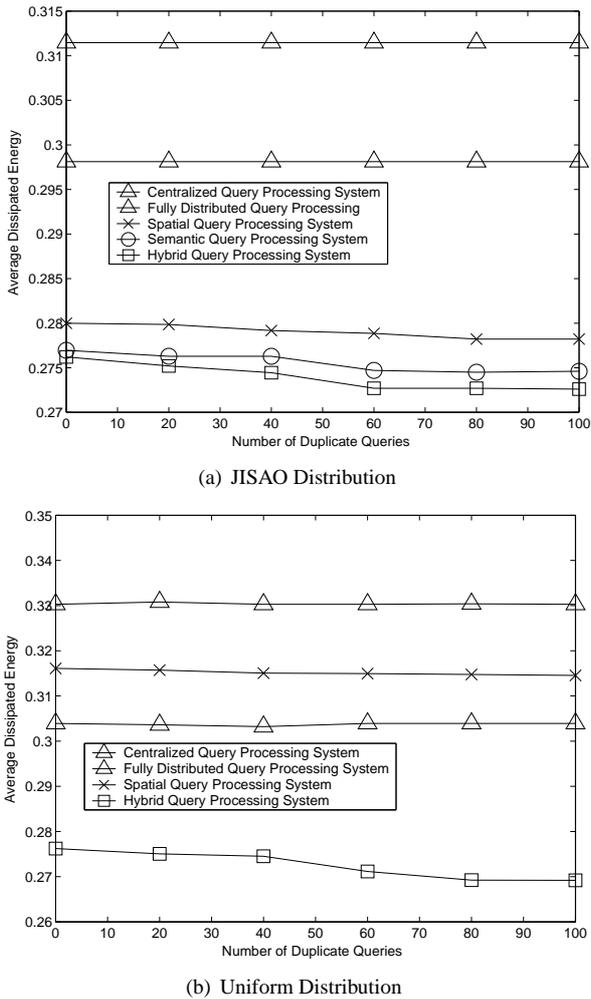
#### 4.2.3 Radius and Similarity in HQPS

The third experiment ascertained the optimal radius and similarity of the HQPS. As dictated by Figure 8, the HQPS performs optimal when QGs have certain spatial and semantic characteristics, radius of 2 and similarity of 17 for the uniform distribution and radius of 3 and similarity of 17 for the JISAO distribution. In addition, Figure 8(b) dictates that (unlike the SEQPS) the HQPS is efficient for topologies that exhibit a low correlation between spatial distance and metadata similarity. The HQPS is able to perform efficiently for these distributions since the QGs it derives are: 1) bounded in spatial size; 2) are comprised of nodes that exhibit similar metadata. First, bounding the spatial size of a QG alleviates the energy cost associated with routing packets (metadata and data) by alleviating the distance (in hops) from a NCH node to a CH node (discussed in the context of the SQPS).

Second, utilizing the similarity variable increases the efficiency of query plans by guaranteeing that nodes comprising a QG have similar metadata (discussed in the context of the SEQPS). For future experiments analyzing the HQPS the optimal radius and similarity (as dictated by Figure 8 will be utilized.

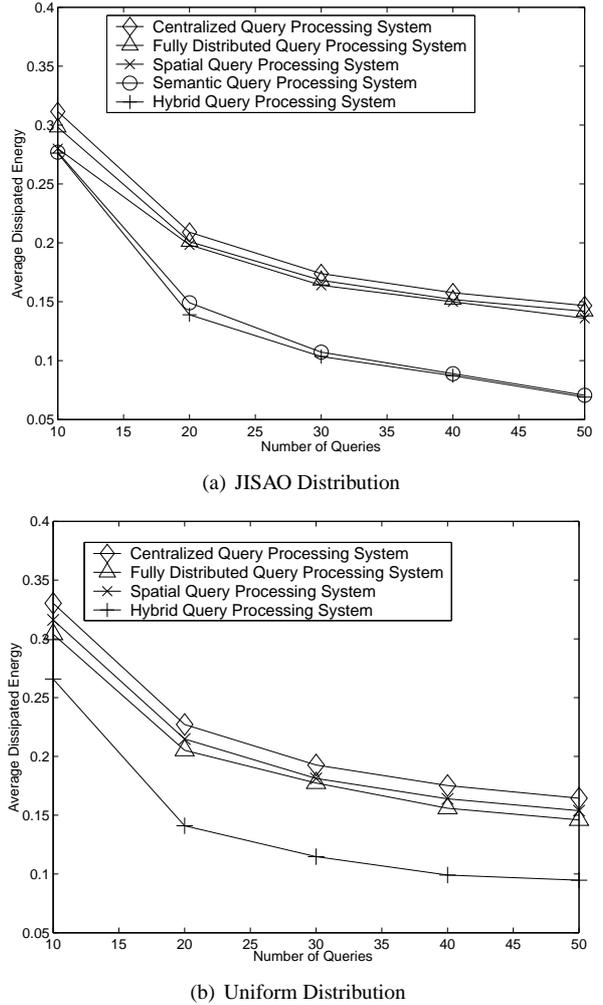
#### 4.2.4 The impact of query frequency

The fourth experiment was conducted in order to determine the effects of query frequency. In this experiment, the frequency that duplicate queries are injected into the sensor network is increased. As dictated by Figure 9 the average



**Figure 9. The impact query frequency has on energy consumption**

dissipated energy will decrease for the SQPS, SEQPS and HQPS as the number of duplicate queries increase. These



**Figure 10. Scalability**

algorithms reduce the average dissipated energy by dynamically adapting QGs based on the query pattern. When infrastructure maintenance is performed these algorithms will utilize query frequency to: 1) intelligently determine CH nodes; 2) weight the similarity parameter (SEQPS and HQPS). First, intelligently choosing CH nodes based on query frequency increases the number of duplicate queries that both a NCH and its corresponding CH are performing. This guarantees that data packets generated by the NCH node are aggregated at its corresponding CH node. Second, weighting metadata similarity based on the query pattern allows the similarity parameter to more accurately represent the true similarity between a NCH node and its corresponding CH. As a result, the data distribution of node's comprising a Query Group will be less diversified (results in more efficient query plans).

### 4.2.5 Scalability

The final experiment increased the number of queries in the sensor network in order to determine: 1) the scalability of each query processing approach; 2) the most energy efficient query processing approach. First in terms of scalability, each query processing approaches average dissipated energy will decrease asymptotically as the number of queries in the sensor network increase (see Figure 10). The average dissipated energy will decrease because the initial energy cost associated with deriving the infrastructure is amortized by a larger number of relatively low energy cost operations (i.e. routing data packets). Second, the HQPS has the lowest average dissipated energy (as expected). However, as dictated by Figure 10(a), the SEQPS average dissipated energy is only slightly greater than the HQPS for the JISAO distribution. Since the JISAO distribution exhibits a correlation between metadata similarity and spatial distance (as discussed), the SEQPS's similarity parameter will (like HQPS's radius parameter) bound the spatial distance between a NCH node and it's corresponding CH node. As a result, QGs derived by the SEQPS will be similar to those derived by the HQPS.

## 5 Conclusion

In this paper we showed that the centralized query processing system inefficiently collects metadata, does not generate efficient query plans and inefficiently aggregates data packets. In addition, this paper has shown that these problems are alleviated if query processing is decentralized. Finally, this paper proposed several approaches to decentralize query processing in sensor networks. These approaches (to our knowledge) are the first to decentralize query processing along with utilize sensor node's innate semantic characteristics. Analysis has dictated that: 1) decentralizing query processing is beneficial in terms of reducing energy consumption 2) both spatial and semantic characteristics should be utilized in the derivation of a distributed query processing system. In addition, analysis has shown that the infrastructure should adapt based on the query pattern.

In this research we assumed the radius of a QG (in the SQPS and HQPS) and the similarity parameter (in the SEQPS and HQPS) are innately known. Determining these parameters is an NP complete problem. In the future we plan on examining heuristics that will generate relatively efficient QGs. In addition, in this research we assumed all queries are injected at a single AP. Our future research will release this constraint to determine it's effect on our proposed query processing systems.

## 6 Acknowledgement

This research was supported in part by the National Science Foundation under Grant no. IIS-0328881.

## References

- [1] Joint institute for the study of the atmosphere and ocean. [http://tao.atmos.washington.edu/data\\_sets/](http://tao.atmos.washington.edu/data_sets/).
- [2] S. Bandyopadhyay and E. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, April 2003.
- [3] S. Banerjee and S. Khuller. A clustering scheme for hierarchical control in multi-hop wireless networks. In *IEEE INFOCOM 2001 - The Conference on Computer Communications*, pages 1028–1037, April 2001.
- [4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networks*, 1999.
- [5] D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: Why do we need a new data handling architecture for sensor networks. In *To appear in the First Workshop on Hot Topics in Networks (Hotnets-1)*, October 2002.
- [6] D. Ganesan, S. Ratnasamy, H. Wang, and D. Estrin. Coping with irregular spatio-temporal sampling in sensor networks. In *ACM SIGCOMM Computer Communication Review archive*, volume 34. ACM Press, 2004.
- [7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. In *IEEE Transactions on Wireless Communications*, volume 1, pages 660–670, New Orleans, LA, Oct 2002.
- [8] W. K. Leow and R. Li. The analysis and applications of adaptive-binning color histograms. *Comput. Vis. Image Underst.*, 94(1-3):67–91, 2004.
- [9] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. *ACM SIGOPS Operating Systems Review*, 36(SI):131–146, 2002.
- [10] S. Madden, R. Szewczyk, M. Franklin, and D. Culler. Supporting aggregate queries over ad-hoc wireless sensor networks. In *4th IEEE Workshop on Mobile Computing Systems and Applications*, Callicoon, New York, June 2002.
- [11] 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*, pages 491 – 502, San Diego, California, June 2003.
- [12] S. Pattem, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. In *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 28–35. ACM Press, 2004.
- [13] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision*, page 59. IEEE Computer Society, 1998.

- [14] Y. Yao and J. Gehrke. Query processing for sensor networks. In *Proceedings of the First Biennial Conference on Innovative Data Systems Research*, pages 21–32, Asilomar, CA, 2003.
- [15] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Infocom*, 2004.
- [16] Y. Yu, D. Ganesan, L. Girod, D. Estrin, and R. Govindan. Synthetic data generation to support irregular sampling in sensor networks. In *Geo Sensor Networks*, pages 9–11, 2003.