Query Aggregation for Providing Efficient Data Services in Sensor Networks

Wei Yu, Thang Nam Le+, Dong Xuan+, and Wei Zhao

Computer Science Department,
Texas A&M University, College Station, TX 77843
+Department of Computer and Information Science,
The Ohio State University, Columbus, OH 43210
{weiyu, zhao@cs.tamu.edu}, {let, xuan@cis.ohio-state.edu}

Abstract—Providing efficient data services is one of the fundamental requirements for sensor networks. The data service paradigm requires that the application submit its requests as queries and the sensor network transmits the requested data to the application. While most existing work in this area focuses on data aggregation, not much attention has been paid to query aggregation. For many applications, especially ones with high query rates, query aggregation is very important. In this paper, we study a query aggregation-based approach for providing efficient data services-. In particular: 1) we propose a multi-layered overlay-based framework consisting of a query manager and access points (nodes), where the former provides the query aggregation plan and the latter executes the plan; 2) we design effective query aggregation algorithms to reduce the number of duplicate/overlapping queries and save overall energy consumption in the sensor network. Our performance evaluations show that by applying our query aggregation algorithm, the overall energy consumption can be significantly reduced and the sensor network lifetime can be prolonged correspondingly.

Keywords—Sensor Network, Query Aggregation

I. INTRODUCTION

In this paper, we study a query aggregation-based approach to provide efficient data services in wireless sensor networks.

Wireless sensor networks have gained much importance in a wide range of civil and military applications. The sensor networks are required to provide efficient data services. Here the application can submit its requests as queries and the sensor network transmits the requested data to the application. Due to the energy resource constraints in the sensor networks, it is important to place our work in context.

Advances in low-powered microprocessor technology, combined with low-cost sensing devices and radio frequency circuits have resulted in the feasibility of inexpensive wireless sensor networks. A vast number of applications like temperature and humidity measurement, habitat trajectory tracking, assembly line production sensing, and enemy movement detection have proven the significance of sensor networks in a wide range of areas in recent years.

Although different in nature, applications in sensor networks share a common property. They view a sensor network as a distributed database to provide the data service [1]. The applications and this distributed database will communicate through a set of queries, which is quite similar to the concept of SQL queries in the traditional database context. Query dissemination and sensed data transmission are therefore the most crucial tasks in the operation of any sensor network. An efficient implementation of these functions will reduce latency of data services and increase the operation time of the energy-constrained sensors and therefore prolong the lifetime of the whole network.

Even though query dissemination has been identified as an important topic in sensor network research, most existing research work so far has focused on data aggregation to provide efficient data transmission. The overhead of the query dissemination is generally ignored with the assumption that query transmission only consists of a small portion of overall data transmission in sensor networks. While this principle applies to a selected group of applications, it does not hold true for all the cases. For example, in applications where the query rate is high, and query attribute is simple (like temperature, noise level), the energy consumption spent on sending and routing queries may far exceed transmitting response data. This is due to following factors: the size of query message may be larger comparable to that of the replied data information; the broadcast nature of the sending and routing queries as opposed to the unicast nature of the data forwarding path, etc. For this class of applications, optimizing query dissemination is critical to improving the overall performance of the sensor network.

In the traditional query dissemination model, applications forward queries to the base station or root node, of the sensor networks. This base station processes the queries one by one and sends queries to proper regions of the sensor network
using the underlying routing infrastructure. This simple approach suffers from shortcomings; applications may pose duplicate queries (one query may contain information of others); overlapping queries (some queries may be interested in the data on the same physical area). In such cases, system resources (in terms of bandwidth or sensor node energy) are wasted because of too many redundant query transmissions in the sensor networks.

In this paper, we attempt address the issue of providing efficient data services in sensor networks using Query Aggregation. The highlights of this paper are:

1) A multi-layer overlay-based framework for the data services. We employ an Overlay-based approach to do the Query Aggregation. The overlay-based framework consists of two important components; Query Manager located at the base station and Access Points (nodes) distributed in the sensor network. This framework effectively leverages the resources/capabilities of components of the sensor network. The query manager receives queries from applications, consolidates and dispatches them to access nodes, that serve as the proxy between the base station and sensor nodes. The query manager mainly focuses on query aggregation plan by taking advantage of it having a global query view. The aggregated queries are then executed by access nodes distributed in the sensor network.

2) Effective query aggregation algorithm. We formally define the query aggregation problem and design an effective algorithm which is applied by the query manager to help consolidate the queries and reduce the overall energy consumption for the query dissemination and data transmission in the sensor network.

We conduct extensive performance evaluations on our proposed frameworks and algorithms. Our evaluation results show that by applying this multi-layered framework and the query aggregation strategy, we can significantly reduce overall energy consumption for both the query dissemination and data transmission phases.

The rest of the paper is organized as follows: Section II introduces the network and query model. In Section III, we propose our query aggregation-based data service framework. In Section IV, we formalize the query aggregation problem and propose our algorithm for query aggregation. In Section V, performance evaluation and analysis results are given. In Section VI, we give more discussions about our proposed framework. We give a survey of related work in Section VII and conclude our study in Section VIII with scope for future work.

II. SENSOR NETWORK AND QUERY MODELS

In this section, we will briefly give some background about our sensor network and query models.

A. Network Model

A sensor network can be modeled as a graph $G(V,E)$, where each vertex represents a sensor node and each edge represents the edge between two nodes when they are within each other’s communication range. This network tracks the values of certain variables like temperature, humidity, etc. Let $V = \{V_1,V_2,\ldots, V_N\}$ be $N$ variables tracked by the network. Each sensor is equally likely to track one or multiple of these $N$ variables.

In this paper, we make the following assumptions: 1) a location-based routing scheme is supported by the sensor network, where each node is aware of its own location by using GPS or other techniques; 2) to simplify our analysis, we assume that the sensor network is laid out uniformly and all sensor have equal transmission ranges; 3) we do not consider failures of sensor nodes; 4) we assume that there is a centralized base station that connects to applications; 5) we assume that that multiple applications can simultaneously send a number of queries to the sensor network through the base station.

B. Query Model

Applications request information from a sensor network through queries. Depending on the nature of the application, many types of queries can be delivered on the sensor network. In general, these queries can be summarized by the following tuple:

$$Q = < S, V, T, F, D >,$$

where

$S$ = Spatial information, indicating the geographical locations that the application is interested in.

$V$ = Attribute information, indicating the list of attributes which the application is interested in.

$T$ = Temporal information, indicating the duration of the query.

$F$ = Frequency information, indicating the frequency at which the data should be reported.

$D$ = Deadline restriction information, indicating the urgency at which the data should be reported.

In the example: “Report regularly noise level and wind speed from region $S_1$, $S_2$ and $S_3$ from time $T_1$ to $T_2$ every second”, where $S = \{S_1, S_2\}$, $V = \{\text{noise level, wind speed}\}$, $T = \{T_1, T_2\}$, $F = 1$ second, and $D = \{\text{not urgent}\}$.

In order to conduct query aggregation, we make the following assumptions about the query model: 1) Query content: each query can ask for one or several attributes ($V$) and must include geographical information ($S$). The list of attributes and geographical locations will be referred to as $\{v_1, v_2, \ldots, v_n\}$ and $\{S_1, S_2, \ldots, S_n\}$ respectively in this paper. We assume that a majority of queries have geographical information. A query without geographical information can be processed through the traditional query processing techniques such as flooding, etc. 2) Query temporal information: we assume that the majority of queries are snap-shot queries, i.e. queries that ask for current value of the sensors as opposed to continuous queries, which asks for sensor values during a
period of time. 3) Query arrival rate: we assume that queries are coming at a relatively high rate, or in other words, the deadline restriction level (D) of queries is not high so that we can temporarily buffer queries for aggregation.

![Query Example 1](image)

**Figure 1: Query Example 1**

### III. QUERY AGGREGATION-BASED DATA SERVICE FRAMEWORKS

#### A. Motivation

We first identify problems with the current query dissemination schemes.

Generally, when the base station receives queries from applications asking for sensor data, it will directly forward them to the sensor network. The transmission of these queries may naively be flooding or it may follow some logic that the intermediate sensor nodes apply [2] [3]. Finally, when the queries are routed to proper sensors (i.e. sensors within the queried regions or satisfying some query conditions), the sensors will start sending data back to the base station, and the base station will deliver data to the applications accordingly. When there are multiple queries from applications, this process repeats until all the queries have been satisfied. At the base station or some intermediate nodes, some caching algorithm may be performed to avoid redundant query forwarding when the queried data is already available in the local cache.

Fig. 1 shows some of the problems of the above scheme on a sensor network, where queries can be location specific and contain multiple attributes. First, when two queries \( Q_1(v_1) \) and \( Q_2(v_1) \) come simultaneously and the queried information is not available in the base station local database, both queries will be transmitted to the network. In this case, the base station just needs to send \( Q_1 \). The information of \( Q_1 \) can be inferred from \( Q_2 \). In the second scenario, \( Q_1 \) and \( Q_2 \) ask for different attributes on the same region. Instead of sending two different queries, we can combine them into one query as \( \langle S, v_1 U v_2 \rangle \), where \( S \) denotes the location where both queries refer to. In the last case, when \( Q_1 \) and \( Q_2 \) ask for different attributes in nearby areas, combining into one complex query as \( \langle S, v_1 \rangle \), \( \langle S, v_2 \rangle \) will still be beneficial. It reduces the energy overhead of sending two separate query messages and more importantly, it avoids disturbing other intermediate nodes in the location-based routing process. When sending the queries to their corresponding regions, the combined query will be routed to a proper node (access node) in one of these regions only once. After that, the access node will separate the combined query. From here, the query part corresponding to the other region will be routed to its destination. Compared to the original case when we have to send 2 separate queries, the number of intermediate nodes involved in the routing process will therefore be reduced.

As illustrated by this example, our motivation in this paper is to build a query dissemination system that 1) performs the query aggregation efficiently to minimize the number of queries sent out, 2) dispatches the aggregated queries to proper regions so that the routing process will disturb a minimal number of intermediate nodes, and 3) prevents data transmission of sensor nodes in the same region from happening multiple times. By achieving these objectives, we can reduce the overall energy overhead for both query transmission and data delivery. Thus, the lifetime of the sensor network can be prolonged.

#### B. Frameworks

As mentioned above, to perform query dissemination efficiently, we have to solve two major problems: aggregating the queries and routing them efficiently to proper regions. We discuss 3 frameworks to solve these problems:

1) Purely sensor network-based framework (PSNF). In this approach, the base station does not conduct query aggregation decision and just simply forwards queries to the sensor network. All the query aggregation intelligence is pushed to the sensor network itself. Each sensor node or sensor region takes care of the queries in overlapping regions. Obviously, this approach is not optimal because intermediate sensor nodes do not have a global view of the whole network. Therefore, sensor nodes in an overlapping region may have to send the same data multiple times to reply for different queries asking for the same attribute.

2) Purely base station-oriented framework (PBSOF): just like a normal centralized system, all intelligence is conducted at the base station, which makes the query aggregation decision based on the input query information. As a result, a number of queries which are not sharing any common regions are generated. The queries are sent to corresponding regions and executed locally. Data from sensors will be sent back to the base station by each region. No query or data forwarding between regions is implemented. The main advantage of this scheme is its simplicity and ease in implementation. However, it can generate a larger number of new queries and disturb more sensor nodes in both the query dissemination and data transmission process.
3) Integrated query aggregation framework (IQAF). To alleviate the above problems, we propose an overlay-based framework, which leverages the resources/ capabilities of system components in the following ways: we consider the fact that base station has a global picture of all input queries and optimal query aggregation plan can be achieved. Besides this, we also consider that the sensor network can take certain roles to execute the aggregated query plan. Thus, a number of sensor nodes as access nodes are selected as the query proxy to actively execute the aggregated queries in a distributed fashion in the sensor network. As a result, an overlay is constructed and includes all query destination sensor nodes, access nodes for different aggregated queries, and query manager. By the cooperation of all components in this overlay, energy consumption of both the query transmission and data delivery can be effectively reduced. The benefit of this approach is that it is highly scalable and energy efficient.

![Figure 2: Multi-layer Overlay-based Query Aggregation](image)

Fig. 2 depicts our multi-layer overlay-based query aggregation framework. It consists of several components: query manager, access points, and normal sensor nodes. The function of each component as well as the workflow of the whole system is described below.

**Query Manager:** The main functionality of the query manager is to conduct the query translation and consolidation. All applications provide their application queries to the query manager, which translates the application queries to a format understandable by the sensor network. Generally, the query manager is acting as a middleware with the following functionalities: i) Syntax check and verification of application queries: checking syntax of the queries and verifying if the query attribute values are proper. ii) Local database lookup: if the data is already in the local cache (database), it will generate the aggregated queries whenever possible. iii) Query aggregation: combining separate queries into a more complex one whenever possible. The purpose is to reduce query dissemination and data transmission overhead. More details about the aggregation algorithm will be presented in the following section. iv) Data delivery: conducting certain mapping logic and delivering the received query data to corresponding applications.

**Designated access points (nodes):** The outputs of the query manager are generally complex queries, which may include several attributes or regions. The designated access points are the "central point" of the queried regions where the queries will be routed using any location-based routing protocols. The sensor node closest to the access point (access node) will receive the complex queries, decompose and reroute them to proper regions. When data is returned from the sensor nodes, the reverse process applies. Note that access points can be multi-layered, i.e. one access point can select several sub-access points for sub-regions if it feels necessary.

To demonstrate its functionality, consider an example shown in Fig. 3. Assume that a designated access node $D$ is selected by the query manager. After $D$ receives the aggregated query asking for $S_1 \cup S_2$, it will decompose the queries and generate 3 separate queries for the sub-regions - $S_1 \cap S_2$, $S_1$, $S_2$, accordingly. The sensors in the overlapping areas are therefore disturbed only once.

![Figure 3: Access Points (Nodes)](image)

**Normal sensor nodes:** The main tasks of normal sensor node are to verify whether its information matches the received query, sense the corresponding data based on matched query requests, and forward the data to the direct up-layer access point.

In general, application queries will go through the following steps:

1) Query manager (QM) receives queries from the applications. If the queried data is not in the local cache (database), it will generate the aggregated queries whenever possible.

2) Based on the input query information, query manager divides the sensor network into zones and calculates the access point corresponding to each zone.

3) Query manager assigns each aggregated query to a proper zone.

4) Using the underlying network routing protocol, the query manager delivers the aggregated queries to corresponding access points.

5) The access node corresponding to each access point analyzes the combined query, decomposes it, generates different queries corresponding to different regions and routes them to proper destinations. It may also assign sub-access points for sub-regions if necessary.
6) Each region or sub-region designated access node conducts the query search, aggregates data if possible and forwards the data to its upper layer designated access node.

7) The designated access node conducts data aggregation if possible and forwards the data to the query manager.

8) Query manager delivers queried data to the corresponding applications.

Generally, our multi-layer overlay-based query aggregation architecture provides an intelligent environment, which can enable all components in the system to effectively manage their resources, complement each other, hence making the system scalable to efficiently support data services in the sensor networks. In the following section, we will study the most important issue – an efficient algorithm to conduct the query aggregation.

IV. QUERY AGGREGATION

A. Problem Definition

There are \( N \) queries: \( Q_1, ..., Q_N \) denoted by set \( Q \). The overall objective of query aggregation is to reduce the overall energy consumption in both query transmission and data delivery by eliminating and merging queries whenever possible. For the query aggregation operation, we define the aggregation function \( \zeta_i \in \zeta \) (with cardinality \( |\zeta| \)): \( \{Q_1, ..., Q_N\} \rightarrow \{Q'_1, ..., Q'_M\} \), where each \( Q'_i \) is merged by \( t_i \) (1\( \leq \)i\( \leq \)N) original queries from set \( Q \) and \( Q'_i \cap Q'_j = 0, Q'_i \cup ... \cup Q'_N = Q'_1 \cup ... \cup Q'_M \). The energy consumption for \( \zeta_i \) with output queries \( \{Q'_1, ..., Q'_M\} \) is defined by

\[
E_{\zeta_i} = \sum_{i=1}^{M} [F(Q'_i) + G(Q'_i) + H(Q'_i)]
\]

where \( F(Q'_i) \) represents the energy consumption for the query transmission phase from the base station to the \( i^{th} \) designated access point, \( G(Q'_i) \) represents the energy consumption for the query dissemination in each aggregated region, and \( H(Q'_i) \) represents the energy consumption for the sensed data delivery from each sensor node to the base station through access point nodes. Our objective is to find a function \( \zeta \in \zeta \) to achieve \( E_{\zeta_i} \leq E_{\zeta_j} \), where \( 1 \leq j \leq |\zeta| \). The system without query aggregation is just one special case of the mapping function, i.e., \( Q'_i = Q_i \) for all \( 1 \leq i \leq N \).

We can easily prove that the above query aggregation problem is \( NP \)-complete by showing that this problem can be reduced to the set cover problem, which is a well-known \( NP \)-complete problem [4]. In the next section, we study the algorithms in detail.

B. Algorithms

The exhaustive-based approach to solve above problem is to find the whole aggregation mapping function space \( |\zeta| \) and select one with minimal energy overhead \( E_{\zeta_i} \). It is easy to verify that

\[
|\zeta| = N \times N \times ... \times N = N^k.
\]

When the query number \( N \) increases, the computation complexity is huge and impractical for real applications.

Referring to the huge computation complexity of the exhaustive algorithm, the reason is that it exhaustively lists all the possibilities of query merge orders. By observing that the key issue is to efficiently find the good query merge order, we propose following weighted zone-based algorithm. The basic idea of our algorithm is listed as following: As the query manager has the information of all query information, it can globally calculate all overlapping regions in the whole query space and find the zones with the most heavy overlapping (calculated by as weight). Based on the overlapping weight, the queries located near to the most heavy overlapping weighted zone need to be merged together with higher priority. Thus, the efficient query merge order can be easily achieved with reasonable performance. In the following, we demonstrate our idea by following example shown in Fig. 4 with five queries.

![Figure 4: Example for Weighted Zone-based Query Aggregation Algorithm](image)

In Fig. 4, we can lay out five queries in the network and create six overlapping zones represented by \( O_i \). Each \( O_i \) has two properties: query overlapping number \( O_{in} \) and zone size \( O_{zn} \). \( O_{in} \) defines the number of queries contributing the overlapping region \( O_{zn} \), i.e., \( O_{zn} \)'s overlapping number is 3 (generated by \( Q_5, Q_3, \text{and } Q_1 \)) and zone size \( O_{zn} \) represents the actual size of that overlapped zone. We sort these \( O_i \) by weight \( O_{zn} \) * \( O_{in} \) i.e., the sorted zone set \( S = \{O_3, O_6, O_5, O_4, O_2, O_1\} \). As \( O_1, O_2 \) and \( O_3 \) are neighbors of \( O_2 \), we remove them from set \( S \) and create a new zone \( O' = O_1 \cup O_2 \cup O_3 \). Then, the set \( S = \{O', O_5, O_4\} \). Based on the weight value of zones, we can assign \( \{Q_6, Q_5, Q_3\} \) into the zone \( O' \) and \( \{Q_2, Q_3\} \) with zone \( O_4 \) (detail is given below). In summary, two aggregated queries are generated: one is \( \{Q_6, Q_5, Q_3\} \) and the other is \( \{Q_2, Q_3\} \).

Before we introduce the detailed algorithm, we discuss the schemes for access point selection and query zone assignment.
(the former will also be used by the latter for the query zone assignment decision) which will be used in our algorithms:

**Access point selection:** we select the geographical central point as the location of designated access point. The detailed calculation is as follows: an aggregated query zone \( O \) includes \( K \) sub-queries \( Q_1, ..., Q_K \) and each query \( Q_i \) is assigned by a weight \( S_i \) (size of the query region) and each query \( Q_i \) has its local central access point represented by \((X_i, Y_i), ..., (X_K, Y_K)\), respectively. The location of designated access point \((X, Y)\) for zone \( O \) can be calculated by the following formula:

\[
X = \frac{S_1X_1 + S_2X_2 + ... + S_KX_K}{S_1 + S_2 + ... + S_K} \quad 3) \\
Y = \frac{S_1Y_1 + S_2Y_2 + ... + S_KY_K}{S_1 + S_2 + ... + S_K}
\]

When the queries are routed towards this access point, the sensor node closest to the access point (access nodes) will take responsibility of decomposing, regenerating, and rerouting the queries.

**Query zone assignment rule:** for the query \( Q \) matching with multiple overlapping zones, care must be taken to conduct proper zone assignment to minimize the energy consumption incurred in the transmission of the query. For example, suppose there are two overlapping zones \( O_1 \) and \( O_2 \) as the selection candidate for the query \( Q \). If \( O_1 \) is merged to \( O_1 \), an overlapping zone is generated with size \( S_{10} \) and the distance between the central designated access point of \( O_1 \) and the new overlapping zone is \( S_{12} \). With given sensor network sensor deployment distribution and topology information, the total distance for the query dissemination in zone with size \( S_{10} \) is represented by \( f(S_{10}) \). Therefore, we have following formula:

\[
d = f(S_{10}) - S_{10} - (f(S_{20}) - S_{20}) \quad 4)
\]

From the above formula, we have: if \( d > 0 \) and \( f(S_{10}) - S_{10} > 0 \), \( Q \) will be merged into \( O_1 \); if \( d < 0 \) and \( f(S_{20}) - S_{20} > 0 \), \( Q \) will be merged into \( O_2 \).

The detail of the algorithm is given below:

**Parameters:**
- \( Q \): set of input queries with cardinality \(|Q|\), each query \( Q_i \in Q \) is denoted by \(<S_i, V_i, B_i>\), where \( S_i \) represents the query region and \( V_i \) represents the query attributes, \( B_i \) is the flag to represent whether query being assigned (true=assigned, false = unsigned)
- \( Y \): set of output consolidated queries by aggregation algorithm with number \(|Y|\), each \( Y_i \) includes \( t \) (1 \( \leq t \leq N \)) number of original queries from set \( Q \)
- \( O \): set of overlapping zones for all queries in \( Q \). The size of \( O \) is \(|O|\) and \( O_1 \) is denoted by tuple \(<S'_i, Q'_i, W'_i, C'_i>\), where \( S'_i \) is the union of a number of query regions, \( Q'_i \) includes a number of queries from set \( O \), \( W'_i \) is the weight of the overlapping zone, and \( C'_i \) is the location of access point for all queries in \( Q'_i \).

**Weighted Zone-based Query Aggregation Algorithm (WZQA)**

**Begin:**
- Process the input queries in set \( Q \) by filtering queries with full cover property
  - For queries \( Q_i, Q_j \in Q \), where \( i \neq j \),
    - If queries are overlapping, i.e., \( S_i \subseteq S_j \) or \( S_i \supseteq S_j \)
      \[
      Q = Q - \{Q_i, Q_j\}
      \]
      \[
      Q_i = S_i \cup S_j, \forall j \geq i
      \]
      \[
      Q = Q + Q_i
      \]
  - End
- // calculate the access point
  - For query \( Q_j \in Q \)
    \[
    Get C_j for Q_j by formula 3
    \]
  - End
- // calculate the overlapping zone and assign the weight
  - For queries \( Q_i, Q_j \in Q \), where \( i \neq j \),
    - If \( S_i \cap S_j \neq 0 \)
      \[
      W_i = 2S_y, \text{ where } S_y \text{ is size of zone } S_y = S_i \cap S_j
      \]
      \[
      O_i = S_i \cup Q_i, W_i >
      \]
      \[
      O = O + O_i
      \]
  - Else
    - //TH is the threshold value to identify whether all zones’ weights are similar
      - If \(|W_i - W_j| < TH\)
        Randomly select \( O_j \in O \)
      - Else
        Select \( O_j \in O, \exists W_i > W_j, i \neq j \)
      - For all \( Q_k \in O_j \), \text{ where } B_k(e \in Q_k) = false
        \[
        \text{If } Q_k \text{ is associated to multiple zones in } O, \text{ i.e.,}
        \]
        \[
        Q_k = Q_{k1} \text{ in } O_{k1} \text{ and } Q_k = Q_{k2} \text{ in } O_{k2}, \text{ where } t_1 \neq t_2
        \]
        \[
        \text{Get } W_{d1} \text{ and } W_{d2} \text{ by formula 4)
        \]
        \[
        \text{If } d_{d1} > 0 \text{ and } d_{d2} > d_{d2}
        \]

**End**
V. PERFORMANCE EVALUATION

In this section, we use simulations to evaluate the performance of the system that uses our proposed framework and algorithms. We will first describe the experimental model and then report performance results.

A. Experimental Model

1) Network Model

We run our simulation on a grid-topology network with size of 1500m x 1500m. The grid size is 5m x 5m and there is one sensor in each grid. We assume that there are N queries, each of which is m-bit long. The queries uniformly request data from the whole region. Each query covers an area of S m². When we combine the queries, we assume that we can achieve a compression ratio \( \alpha \) (by reducing message headers, some duplicate fields, etc.). For example, the combined message size of \( k \) query messages is \( \alpha \times k \times m \), where \( \alpha = 0.7 \) and \( S = 200 \) in our system.

2) Sensor Radio and Energy Model

We assume a simple model of radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics and power amplifier. We assume that each sensor works in free space mode with some experimental data introduced in [5]: the energy consumption of sending message is calculated by \( E_{\text{send}}(a, b) = E_{\text{elec}} \times b + E_{\text{amp}} \times b \times a^2 \) and the energy consumption of receiving a message is calculated by \( E_{\text{recv}}(a, b) = E_{\text{elec}} \times b \), where \( a \) is the message size and \( b \) is the message transmission distance between the sender and receiver. \( E_{\text{elec}} = 50 \text{nJ/\ bit} \), and \( E_{\text{amp}} = 100 \text{pJ/\ bit\ \ m}^2 \). Since the energy consumed for processing queries and sensing data consists of only a very small portion of the overall energy consumption (energy consumed to process 100 million instructions almost equals that to transfer 10 bits of data), we do not take it into account in our calculation.

3) Performance Metrics

We use overall energy consumption (OEC) to compare different approaches in our simulation. OEC is defined as the total energy consumption of the whole process from the time the query manager starts sending out the queries until it receives the replied data from the sensor network. The higher the OEC value is, the worse is the performance.

4) Evaluation System

We assume there is one query manager deployed at the base station which conducts the query aggregation. We define the Query region overlapping degree \( R \) as the ratio of overlapping region size and the original query region size. For example, two queries \( Q_1 \) (query region size \( S_1 \)) and \( Q_2 \) (query region size \( S_2 \)) have the overlapping region with size \( S_{12} \). In this case, \( R = S_{12} / (S_1 + S_2) \). The range of \( R \) is [0, 1], where 0 represents the case where the query regions do not overlap and 1 represents the case where the query regions are exactly the same.

In this part, the evaluation parameters are listed by the tuple \(<N, A, R, Z>\), where \( N \) is the number of total input queries submitted by applications, \( A \in \{\text{PSNF}, \text{PBSOF}, \text{IQAF}\} \) represents different query processing approaches defined in Section III (PSNF represents the Purely Sensor Network-based Framework, PBSOF represents the Purely Base Station-Oriented Framework, and IQAF represents our Integrated Query Aggregation-based Framework). \( R \) is the query region overlapping degree, and \( Z \) is the average query region size.

B. Performance Results

In this section, we report the performance results along with our observations. Due to the space limitations, we only present a limited number of cases here. However, the conclusions we draw generally hold for many other cases which we have evaluated.

1) The Sensitivity of Query Number

Fig. 5 shows the data on the sensitivity of energy performance for different input query numbers. The general system is represented as \(<*, *, 0.25, 50x50>\). In this Figure, \( X \) axis represents the total number of input queries and \( Y \) axis represents the total energy consumption. From this figure, we have the following observations: i) overall, our IQAF outperforms both PSNF and PBSOF frameworks. For example, with a large number of queries, i.e., from 90 to 120, the IQAF can achieve around 80%-120% performance improvement over the PSNF and PBSOF. The result matches our expectation because as IQAF consolidates the queries and pushes the query aggregation execution to the sensor network. The energy cost for both query transmission and data delivery has been significantly reduced. ii) PBSOF performs similarly
as the PSNF. The reason is because PBSOF conducts the query execution at the centralized base station. Although it can reduce the energy consumption by removing the redundant queries for the overlapping regions, the drawback of this approach is that more queries for overlapping regions are generated, which incurs more energy consumption. iii) The overall energy consumption is sensitive to the number of queries. A larger \( N \) generally implies more queries, and therefore, more energy consumption. However, in IQAF, this problem is alleviated because, when the number of queries increases, there will be more chances of overlaps between query regions, which can be effectively reduced by our query aggregation approach.

2) The Sensitivity of Query Region Size

Fig. 6 shows the data on the sensitivity of energy performance for different query region sizes. The general system is configured as <90, *, 0.25, *>. In this figure, the \( X \) axis represents the different query region sizes and \( Y \) axis represents the total energy consumption. As the query size is enlarged, the overall energy consumption also increases. This is because a larger region size means that more sensor nodes are involved, or more query/data transmissions are performed. Regardless of the query size, our IQAF performs better than the other 2 schemes with the same reasons given above.

3) The Sensitivity of Query Overlapping Degree

Fig. 7 shows the data on the sensitivity of energy performance for different query overlapping degrees. The general system is configured as <90, *, *, 50x50>. In this figure, the \( X \) axis represents the different query overlapping degrees and \( Y \) axis represents the total energy consumption. When the query regions are not overlapping, IQAF can not take advantage of query aggregation. Therefore, performance of IQAF is almost the same as other methods. However, as query regions become more and more overlapped, IQAF shows significant improvement. As shown in this figure, when the overlapping degree becomes 0.2, overall energy consumption of IQAF is less than half the other schemes. The behavior of PBSOF is explainable because, although PBSOF can prevent the duplicate queries for overlapping regions, the newly generated queries for overlapping regions require extra energy cost for the query/data transmission.

VI. DISCUSSIONS

A. Query Adaptation

In this paper, we assume a simple query model: a number of spatial snapshot-based queries arrive at the query manager in a short time. However, our framework can easily adapt to other query models. A detailed discussion follows:

1) Query content adaptation: A query without geographical information can be processed through the traditional query dissemination techniques such as direct diffusion [2]. In this case, the query does not have much space for the query aggregation, but query messages can still be consolidated based on the query contents in certain degree.

2) Query temporal adaptation: We assume that the majority of queries are snap-shot queries. Our framework can actually support continuous queries, as our proposed query aggregation approach optimizes the energy consumption for both query transmission and data delivery. To adapt to queries with different frequency attributes, our designated access node selection procedure can consider the query frequency attributes and assign each query with a weight according to query frequency attributes apart from using the query region size in our proposed algorithm. Thus, the access point selection can do better to fit this requirement.
3) Adapt to different query arrival rate pattern for query manager. In this paper, we assume that queries are coming at a relatively high rate and also deadline restriction of queries is not high so that we can temporarily buffer queries for the query aggregation. To adapt to the different query arrival rates in the temporal base, our query manager can also conduct adaptation in the following way. It can predict the query pattern based on historical query information and adapt correspondingly. If the query arrival rate is low and each query has tight deadline requirement, the query manager will treat the queries with deadlines with higher priority and immediately forwards such queries to the sensor network.

B. Routing Protocols

1) Underlying routing protocol for query message transmission and data delivery. In this paper, we assume that query message and data delivery can be handled by the underlying sensor routing protocols, i.e., geographical-based routing protocol. However, our framework is also compatible with other routing protocols such as, direct diffusion-based routing protocol [2], cluster-based routing protocol (LEACH) [6], etc. Although our work mainly focuses on the query aggregation, existing protocols for the data aggregation can also easily be adopted in our framework, i.e., multi-layered access points are good places to conduct the data aggregation.

2) Overlay-based communication protocol for the query transmission and data delivery. In our framework, when the query aggregation plans are generated by the query manager, a tree-based overlay is constructed and includes all selected destination sensor nodes, designated access point nodes for different regions, and the query manager. In this overlay, the following three tasks need to be conducted: i) Overlay construction and maintenance: This task deals with how the overlay topology is constructed and its maintenance under network dynamics due to the failure of sensor nodes. ii) Query routing forwarding: This task deals with how to process the aggregated queries, forward query request in the overlay to achieve sub-region query delivery, and guarantee that destination node can receive the query. iii) Data delivery: This task deals with how to process the data and efficiently forward to the overlay root node (query manager) through the dedicated off-line constructed tree. Generally, we can borrow some existing overlay and Peer-to-Peer (P2P) research results here [7].

C. Energy Consideration of Access Point

In our query aggregation algorithm, the access point is selected as geographically central point. As the access point takes responsibility for query execution and data forwarding, the energy consumption could be an issue. As the access point selection is per query-based, different access points are highly likely to be selected. Even in certain cases, some access points may have been selected heavily and its usable energy capacity becomes low. In order to avoid this, our algorithm can rotate the access point selection to other nearby nodes based on information, i.e., the historical information of access point selection, current energy status for the sensor node if available.

VII. RELATED WORK

In this section, we review research work in the areas of sensor network and data query related to our study.

In the sensor network, much work has been focusing on energy efficient routing to maximize sensor network life time [8], effective target tracking for certain applications [9], etc.

From the sensor data services perspective, there are mainly two approaches: fixed sensor data collection and distributed on-demand sensor data collection approaches. Due to the energy limitation of sensor network, the distributed on-demand sensor data collection is becoming the more popular, which treats the sensor network as a large distributed database [1]. There are some fundamental differences between the sensor database and traditional database such as, traditional database query normally does not have much query search overhead [10]. Regarding to the sensor data services, much existing work focuses on the data aggregation such as, direction diffusion [2], aggregation modeling [11], multi-range sensor query [3]. No much work has been discussed about the framework to conduct the query aggregation, which is very critical for certain types of query application such as applications sending large number of snap-shot queries.

Our framework leverages the query information and resource at different components in the sensor network by introducing query manager as a middleware to conduct query consolidation and sensor network itself contribute possible resource to execute query aggregation in a distributed fashion. We list following work most related to us: 1) Although the query proxy concept has been wildly introduced in the traditional database system design and some initial work introduces the query proxy to support limited type of queries (continuously data) in the sensor network [12], our work is focuses more on designing query consolidation mechanisms to reduce the energy consumption in both query transmission and data delivery. 2) Regarding to the sensor network conduct the query execution for the complicated query, the work at [13] conducts the initial modeling analysis. However, this work just conducts some initial analysis about complex query search process without considering any query aggregation methodology to reduce the energy overhead.

In general, our approach extends existing studies to effectively support sensor data service by adopting a multi-layered overlay-based query framework and designing efficient query aggregation mechanisms. As discussed, our framework can also easily adapt to wide range of query models and also compatible with existing sensor routing protocols.
VIII. FINAL REMARKS

In this paper, we propose a new framework for efficient data service in sensor networks. To the best of our knowledge, this is the first study that leverages existing research work and address the issues in this aspect. In summary, our technologies include the following: 1) a multi-layer overlay-based framework for efficient sensor data service, 2) an effective query aggregation mechanism to reduce overall energy overhead for the data services in the sensor network. We conduct extensive performance evaluations on different frameworks and algorithms. Our evaluation results show that by applying our multi-layered overlay-based framework and the query aggregation strategy, we can significantly reduce the amount of query traffic and energy consumption for data services.

Our work has broad impacts. With a tremendous spurt in sensor network deployment demanded by sensor network applications, our approach can effectively deliver generic sensor information query and data dissemination services. There are several directions to extend our study. First, in the original model, we implicitly assume that the underlying architecture supports location-based routing. Extending our algorithm so that it can support other routing protocols would be one direction. Second, in the query aggregation algorithm, we construct our zones based on the input query zones and do not consider the existing topology and distribution of sensors in the network. Combining both dimensions (input query zones and network topology) in our algorithm will certainly produce better results. Finally, we wish to explore the functionality of our system in a more dynamic situation, where nodes can join and leave the network frequently.

ACKNOWLEDGEMENT

Dong Xuan’s work is supported in part by National Science Foundation under grant No. ACI-0329155. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES