Querying Sensor Networks: Techniques, Evaluation, and New Directions

Sayed Ahmed, M. Rasit Eskicioglu, and Sajid Hussain
Department of Computer Science
University of Manitoba, Canada
{sayed, rasit, sajid}@cs.umanitoba.ca

Abstract

Advances in sensor node hardware, which comprises sensors, embedded processors, and communication components, have made the large-scale deployment of sensor networks a reality. Various sensor network applications ranging from monitoring applications to military applications require sensor nodes to collect data over a continuous time period. The placement, management, and processing of sensor network data necessitates an effective data storage, management, and query processing policy. This paper attempts to identify the key query processing techniques used in sensor networks. Design goals and challenges for query processing techniques are identified. The techniques are evaluated in terms of efficiency, scalability, applicability, and reliability. The evaluation of the techniques is guided by the distinctive query processing features supported by both types of sensor networks: conventional and wide area. Moreover, we argued for the integration of conventional and wide area sensor networks and addressed the integration issues and design goals. In particular, a query processing architecture is proposed to meet the emerging needs of sensor networks. The architecture addressed the requirements for different layers of the integrated sensor network components: base stations, sensor nodes and wide area workstations. Additionally, the future research directions for query processing are outlined.

1. Introduction

Recent advances in technology have enabled the widespread deployment of sensor networks consisting of a large number of sensor nodes with sensing, computation, and communication capabilities. Sensor networks aim at providing an efficient and effective bridge between physical and computational worlds. Typical sensor network applications show a great deal of diversity including contaminant transport monitoring, marine microorganisms analysis, habitat sensing, seismic and home monitoring, and more. From small, low powered, and wireless sensors to webcams, microphones, and pressure gauges, pervasive sensors provide excellent new services for these applications. Querying and monitoring the physical world is one of these services bearing utmost importance. Consider for example, a Trip Planner service used for finding out the current traffic condition of streets leading to a users destination. The user enters into a PDA (or a car navigation system) his origin and destination and gets back information regarding the traffic condition of every possible street that leads to the destination. A threefold requirement is envisioned to create such a Trip Planner service. First, sensors are located in different locations on the street and can determine the current traffic rate. Second, a sensor database is needed that can store dynamic information such as the current traffic rate and the speed of vehicles along with some static information such as possible ways to reach the destination. Third, a web accessible query processing system is needed to provide replies to high-level user queries.

Unfortunately, the resource constraints related to sensor nodes such as computation, communication, power consumption, uncertainty in sensor readings have posed a numerous challenges in query processing for sensor networks. Therefore, research on efficient query processing for sensor networks has become an important activity. To this end, databases in sensor networks are recently attaining a great attention of researchers.

So far, reasonable work has been done on different query processing techniques in the context of both conventional and wide area sensor networks. Conventional sensor networks consist of very small, power-constrained, wireless sensors having very simple processors, little memory and limited sensing capabilities. The Directed diffusion [12], the Cougar approach [3], [21], [22], the TAG [15], the Fjords [14] and some other techniques have been devised for querying the conventional networks. These query processing techniques operate directly on continuous and never-ending streams of sensor data and take special care on using low power as the sensor nodes are power-constrained. Besides, wide area sensor networks can have both tiny and large (powered and high-bit-rate) sensors [10] spread over a wide geographically dispersed region. Databases used by these networks are also dispersed and contain data that correspond to the most recent updates by the sensors. Unfortunately, little research has been conducted on wide area sensor databases. The IrisNet (Internet-scale Resource-intensive Sensor Network Services) [8] project works with wide area sensor databases and develops a novel query processing technique that is applicable to hierarchical distributed databases created by any wide area sensor network.

Our overall strategic goal is to identify the key features adopted by current query processing techniques for sensor networks, evaluate them in terms of efficiency, scalability, applicability, and reliability and to outline the most challenging future research directions for the development of new techniques. We envision providing emphasis on:

- **Identifying the requirements and challenges for sensor networks queries:** The requirements for query processing may vary depending on the types of sensor networks and the applications being deployed. Nevertheless, a number of common requirements are shared among all sensor network applications regardless of the usage such as a simple query language, aggregation of query data, less query response time, and multi query management. At the same time, sensor networks pose a number of query processing challenges due to the resource constrained sensor nodes (low power, small size, and less storage) and the nature of sensor data.
- **Identifying the inter-operability and integration issues:** Query processing techniques should be compatible to both conventional and wide area sensor networks. Besides, integration of conventional and wide area sensor databases should be supported to provide a common query language interface.
- **Optimum design:** The first tendency could be to optimize each and every component to the maximal extent. Identifying the area where to put the main optimization effort is important. For example, supporting more in-node aggregation operations, filtering sensor data, and supporting the usage of intermediate result for different queries could be important design issues.
- **Identifying trade-offs due to resource constraint:** A number of possible trade-offs should be considered depending on the sensor node resources. For example, increasing node computation to minimize the amount of communication in order for the power consumption to be kept a minimum could be a design goal where the underlying network architecture supports sufficient local storage.

The remainder of the paper is organized as follows. Section 2 outlines important query processing challenges. Section 3 presents different query processing techniques and evaluates them. Section 4 provides our proposed query processing architecture for sensor networks, section 5 provides the challenging research directions for the development of new querying techniques. Finally, the paper ends with conclusions in section 6.

2. Query Processing Challenges

In this section, an overview of research problems for query processing in sensor networks is presented in the context of both conventional and wide area sensor networks.

A. Conventional Sensor Network Databases

- **Streaming Data:** In traditional database systems, data is static and the query processor pulls data from the disk, organizes the movement and handling of data but sensor networks involve streaming data where data is continuously pushed to the query processor, which must react to the arriving data. The data arrival rate is sometimes extremely high or bursty, hence processing time or memory usage should be carefully handled. Since data is dynamic in nature, the data must be processed dynamically as it arrives and can be spooled to the disk in background mode [19].
- **Continuous Queries:** Monitoring applications are common sensor applications where queries remain continuously active. When new data comes or an event happens, data are supplied to the stored and continuously active queries whereas in a traditional database system, the arrival of queries initiates access to the static database. The streams pushed to continuous queries can be effectively infinite leading the query processors to include different query semantics, non-blocking query operators, and tolerance of failure [19]. Continuous queries can be extremely long lived, which makes them vulnerable to changes over time in performance or data characteristics and qualities.
- **Resource Constrained Sensor Nodes:** Sensor nodes are resource-constrained having very low power, limited memory, very simple processors, and limited sensing capabilities. Besides, wireless communications consume much battery power. Therefore, the query processor should be designed so as to overcome these limitations [21].
- **Aggregation:** Aggregation refers to collecting, filtering, and processing sensor data in the network, and supplying the result to the central nodes. Aggregation reduces wireless communication and hereby minimizes power usage. Hence, an efficient aggregation plan is needed considering the volatility of the underlying data [21].
- **Query Languages:** Technological development is leading to various sensor based applications driving to diverse requirements for the query language. Hence, the development of a user friendly, efficient, generalized query language is at a difficult point [21].
- **Query Optimization:** A complex query can consist of a lot of parameters and operators along with diversified user requirements. A great variety of optimization options are also available for such complex queries but it is the query processors responsibility to find a good plan within a short computational time to execute these queries [21].
- **Catalog Management:** To generate a good plan within a short time for user queries, the query optimizer needs to know the current status of the network. Data about the network topology and status of the network helps to calculate the costs and benefits of different query plans and make a good selection among them [21].
- **Multi Query Optimization:** Simultaneous multiple query support and optimization is another challenging issue [21].

B. Wide Area Sensor Network Databases

In addition to the above mentioned query processing challenges wide area sensor networks offer some unique query processing challenges not common to the conventional sensor networks. They are as follows

- **Data Partitioning and Placement:** Physical placement and keeping track of sensor data is an important issue due to the huge volume of data and multiple and geographically dispersed databases. The data is partitioned in such a way that query processor can find appropriate site for providing results within a short period [8].
- **Caching:** An efficient caching strategy can help a lot. If data is cached depending on the query and application in different sites they can provide dramatic decrease in query response time [8].
- **Query Processing and Query Language:** The query language should support diverse applications and at the same time should be efficient and user friendly. The query processing technique should be able to handle a large number of sensors, diverse queries, multiple concurrent queries, and large volumes of data [8].

3. Different Query Processing Techniques

This section presents different query processing techniques for both conventional and wide area sensor networks with their relative advantages and limitations.

A. Conventional Sensor Networks

- **Directed Diffusion:** Directed diffusion defines a data centric dissemination and co-ordination paradigm for delivering sensed data to the user. User queries or tasks are inserted as descriptive interest messages through a sink or entry node such as {Type = four footed animal, Interval = 20 ms Duration = 10s, Rectangle = [-100,100,200,400]}. The interest query traverses from the sink node to the destination nodes using broadcast/multicast to the neighbors. During the traverse, gradients are created to keep track of delivering nodes with some identification of the interest message. Target nodes and neighbors co-operate to sense an object and the results traverse back to the sink following the gradients to the sink [12].
  Diffusion, however is significantly lower-level, non-declarative interface and pays a considerable price in terms of usability and lacks latent for aggregate queries. However, in Directed Diffusion, limited data aggregation is performed on every node on the path from the source of the data to the source of the interest. Directed diffusion has the potential for significant energy efficiency; even with relatively un-optimized path selection it outperforms an idealized traditional data dissemination scheme like omniscient multicast. Directed diffusion can work against node failures by selecting new path and views the network as a data centric routing system (not as a database). This technique is totally centralized and uses limited caching and in-network processing.
- **Cougar Approach:** Cougar approach mainly focuses on in-network distributed query processing. In case of centralized traditional sensor networks, sensors are pre-programmed, and always send data to the central node for off-line querying and processing. In addition, the network and the applications are static. Users cannot change the system behavior on the fly. Besides, there is huge consumption of battery power due to wireless communications [3], [21], [22].
  Cougar approach solves these problems to a great extent. The main idea of Cougar Approach is in-network data processing. The sensors are capable of storing some data and perform processing like aggregating records or eliminating irrelevant records and send only the necessary data to the base station. Besides, a declarative, user friendly query language is developed. Hence, users are shielded from the internal structure of sensor networks. A query optimizer at the base station depending on the user query generates an efficient query plans for in-network query processing. An efficient query proxy layer in each sensor node is designed that resides between the network and the application layer. The query plan specifies data flow routes and exact computation plan such as which node will perform the computation. The plan is then delivered to all the related sensors of this plan. Also, control structures are needed to synchronize sensor behavior. Query optimizer assigns a leader for a group of sensors who will collect data and send data to the leader. Leader nodes perform computations while non-leader nodes have a scan operator to read sensor values periodically and to send them to the leader node. In addition, their plan contains an aggregation operator to aggregate data from other sensors. In essence, Cougar approach introduces aggregation, in-network-query processing, query proxy layer, a declarative query language in attention to using less battery-power.
- **Acquisitional Query Processing (ACQP):** ACQP works in the setting where the data to be processed does not exist at the sites when the query is issued but generates data actively by acquiring samples from special sensing devices. ACQP uses a set of techniques for deciding when and where to sample and which data to deliver with the combination of in-network aggregate processing like Cougar Approach. Based on these concepts, TinyDB, a DBMS for sensor networks is built. Unlike existing solutions for data processing, TinyDB does not require to write embedded C code for sensors. Instead, like traditional databases, TinyDB provides a simple, SQL-like interface to specify the query, along with additional parameters such as the rate at which data should be refreshed. Given a user query, TinyDB collects data from motes in the environment, filters it, aggregates it together, and routes to a PC. TinyDB does this using power-efficient in-network processing algorithms [13], [16].
- **Fjords Architecture:** The Fjords architecture [14] manages multiple queries over many sensors, and reduces sensor resource demands but provides high query throughput. The architecture addresses the low level infrastructure problem in a sensor stream query processor using two techniques. First, it combines proxies, non-blocking operators, and conventional query plans. Second, sensor proxies serve as a bridge between sensors and query

plans, using sensors to facilitate query processing while being sensitive to their power, processor, and communications limitations. The Fjords architecture provides support for integrating streaming data that is pushed into the system with disk-based data, which is pulled by traditional operators. Fjords allow combining multiple queries into a single plan and explicitly handle operators with multiple inputs and outputs. Previous database architectures are not capable of combining streaming and static data. The hybrid approach adopted by Fjords is an essential part of any data processing system which claims to be able to compute over streams. These solutions are an important part of the Telegraph Query Processing System [19], which seeks to extend traditional query processing capabilities to a variety of nontraditional data sources. Telegraph when enhanced with the Fjords sensor stream processing techniques, enables query processing over networks of wireless, battery powered devices that cannot be queried through traditional means.

- **Continuous Queries:** Recently, there have been a lot of interests in streaming data and continuous query processing. TelegraphCQ [5] is a recent project to handle continuous queries which focuses on the extreme adaptability and it has the required novel infrastructure. NiagaraCQ [6], an XML-based engine builds static plans for the different continuous queries in the systems, and allows two queries to share a module if they have the same input. Babu and Widom [2] deal with STREAM, which focuses on computing approximate results, analyzing the memory requirements of posed queries and running queries in a bounded amount of memory. The Aurora [4] system works with quality-of-service for queries to meet the need for heterogeneous user requirements. Gehrke et al. [9] consider the problem of computing correlated aggregate queries over streams, and presents techniques for obtaining approximate answers in a single pass. Yang et al. [11] discuss data structures for computing and maintaining aggregates over streams to support fast lookup of aggregate results based on time line. Sadri et al. [18] propose SQL-TS, an extension of the SQL language to express sequence queries over time-series data.

B. Wide Area Sensor Networks

There is little research on wide area sensor databases while the Intel IrisNet project has done a major work to address the challenges to support a large number of sensors and high query volumes in wide area sensor databases. The sensor database is partitioned across multiple sites that operate in parallel. User queries are directed to the sites that contain the results. A site manager called organizing agent works in each site. Communications with the sensors are through sensor proxies called sensing agents, which are the powered and internet connected PCs. Sensor proxies collect, filter, and aggregate data then send update queries to the sites that own the data [8], [10].

To make queries flexible, easy to use and to support the growing trends to XML standard in the IrisNet XML database system is used. XPATH or XSLT for querying, and SixDML or XUpdate for updates are used. The distributed site hierarchy is represented as an XML document. The data in sensor networks is fairly diverse and heterogeneous, and hard to capture in a rigid data model. The schema of the data needs to be constantly adapted to support the growing and diversified needs of the users, and evolving capabilities of the sensors. Dynamic insertion and deletion of new fields is easy in XML. Besides, as sensors take data from a geographic location, it is quite natural to organize the sensor data into a geographic/political-boundary hierarchy (as depicted in Figure 1). Representing and making use of such a hierarchy is quite natural in XML, but very challenging in the relational, or an object-relational data model [8].

Sensor data is placed in any number of host nodes and near to the sensors but for efficient querying data are cached to anywhere depending on the needs of the queries and applications. An efficient partitioning and caching strategy supporting partial match caching is presented which ensures that even partial matches on cached data can be exploited and that correct answers are returned. A dynamic self starting distributed query is supported where the queries based on the XPATH query statement and the XML document find the least common ancestor (LCA) of the hierarchical database fragments where query results exist and a DNS like server finds the IP address of the LCA. Moreover, a novel query-evaluate-gather based on XSLT determines which data of the local database is part of the query result and how to collect the missing data [8].

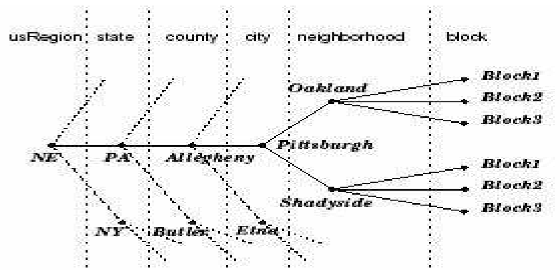

Fig. 1: Site hierarchy [8]

The IrisNet has similar structure with DNS [17] and LDAP [20] but DNS, and LDAP support a small set of lookup queries whereas IrisNet supports rich query languages. Latest works on peer-to-peer databases [7], [11] are quite similar to the IrisNet works. The IrisNet is mostly hierarchical but for performance reasons, the participating sites are treated as co-operating peers. The IrisNet differs from peer-to-peer databases because of its query processing part, caching, and partitioning strategies.

4. PROPOSED ARCHITECTURE:

Based on our study and the above discussion, we propose an architecture for query processing in future sensor networks.

The components of the proposed architecture and the functionalities of the components are provided as below:

- Data aggregation and filtering capability in each sensor node as well as in the base stations to reduce communication as well as power consumption.
- A query proxy layer at each sensor node helping the query plan and the query optimization generated at the base stations.
- A Query optimizer/parser at the base stations performing semantic analysis, normalization, and optimization of the queries posed at the base stations. Also at each workstation in the wide area sensor networks that run sensor applications a simple parser (depending on the application) will be placed. This parser will use the XML document, the fragmentation strategy, and the DNS like system to generate an efficient query plan in addition to doing semantic analysis, normalization, and optimization.
- At the base stations, gateway components to make necessary transformations (XPATH to SQL or SQL to XPATH or a traditional query into a continuous/aggregate query) for the inter-operability and integration between wide and conventional sensor networks.
- Security enabled services at each work station in wide area, and also in the base stations along with a simple security module at each sensor node. Security is not essential for each type of application so the security modules will be developed depending on the applications.
- Caching, partitioning, and site finding components at the workstations in wide area sensor network.
- Catalog Management and Metadata Management at the base stations. However, in sensor nodes protocols are needed to help them and support them.
- Data structures, algorithms and architectural supports (i.e. multiple gateways) are needed to support multiple queries and share the common intermediate results among the queries.
- An appropriate transport layer protocol at the sensor nodes. TCP is not suitable for end to end connection for conventional sensor networks as sensor nodes are data centric attribute based and addressless. The base stations can work as bridges between transport protocols for wide (TCP) and conventional sensor networks [1].
- Energy efficient and data centric routing algorithms in all the components.

The architecture described above is provided in Figure 2.

5. Future Directions

The analysis of current querying techniques for sensor networks opens a plethora of new research directions at the boundary of sensor database systems and networking:

- We find that conventional sensor networks generally use SQL like declarative language while their wide area counterpart uses XML and XPATH/XSLT query language. So, coordination is needed between them to work together. Some proxy layer can work in the terminal base stations to convert XML to SQL. Also XML query can be utilized in the power constrained sensor networks in future. We envision one step further integrating conventional and wide area sensor databases to provide a common query language interface so that no proxy layer/converter is needed. Besides, very changing nature of sensor data can well fit into XML representation. So, we suggest XML and XPATH queries in both wide and local area sensor networks. However, proper experiment and analysis would give the right decision.

Aggregation Support	Security Proxy	Application Layer for Sensor Nodes
Data Structure for Multiple Queries	Query Proxy	
XPATH/SQL Converter	Security Proxy	Application Layer For Base
Query Parser, Planner, Optimizer	Power Control	
Meta Data Management	Catalog Mgmt.	
Aggregation/ACQP Support		
Continuous/Streaming Query Support		
Query Parser	Caching Comp.	Application Layer For Wide Area
Data Partitioning Logic	Site Finding	
Application Dependent Security		
Wide Area Network to sensor node communication support		Transport Layer
Energy Efficient Data Centric Routing		Network Layer
Protocols for Catalog Mgmt.		

Fig. 2: Proposed Query Processing Architecture

- We see sensor network applications treat sensor data as a stream and continuous. Besides, base station database can be treated as hybrid where some static data is kept and continuously updated data are coming. Also the static data may expire after sometimes. In some applications queries are continuous and long-lived. Continuous queries can be active all the time or can act only in the presence of an event. Supporting aggregate queries which reduce much communication energy is also important. Hence, a generalized DBMS is required that can handle diversified types of applications, situations and user needs. Since the sensor nodes support limited storage, the code size should be small. Alternatively, the base/central node can load appropriate modules to the sensor nodes depending on the application. This on-demand loading of modules can use the concept of swapping/paging in the base station hard drive, which requires communication causing much power consumption. Hence, the algorithms need to be power efficient.
- In wide area sensor networks, different data mining approaches can also be utilized over historical sensor data to find historical trends. Historical trends can be used to provide better answers to user queries. For example, data mining approaches can be used to find the trends how fast a parking space is filled or when the parking space becomes busy. So in addition to mentioning the available

free space count data mining approaches will help the user know about the characteristics of the parking lot and take proper decision [8]. The statistical or data mining algorithms can be integrated within each database so that XPATH query using the XML document can retrieve and use the information from appropriate site.

- None of the current querying techniques for sensor networks are concerned with database security. Integration of security mechanisms inside database can prevent unauthorized user from querying. We propose to introduce a security layer in application layer of each sensor node. In the gateway node, the application layer can add some identification of the authority to the query and the application layer of the other sensor nodes will permit only the queries from valid authority. Besides, encryption techniques in query request and reply can be introduced but encrypting can use more energy leading to much research to utilize appropriate encryption algorithm. Justification of the need of the security and the encryption by the application is also required.
- Continuous queries that are very frequent and natural to conventional sensor networks must also be supported and handled in wide area sensor databases to make the networks practically useful. Both architectural support and query language support also efficient optimization and query processing plans are needed to pose continuous queries over wide area sensor networks. Yet the Intel IrisNet has not implemented the support but they hope to support continuous queries where the various data structures that the IrisNet used to collect information from different organizing agents will be utilized for this purpose [8]. Continuous query processing supports are provided in several works mentioned previously in section 3; however most of them are for conventional sensor networks and are centralized. However, we argue for continuous queries in both types of networks and also where the networks are integrated.
- Although the Intel IrisNet project supports aggregate queries; the caching infrastructure is not sufficient for caching results from aggregate queries. Consequently more sophisticated caching strategy is needed. View based semantic caching can help in this respect [8]. Also sophisticated algorithms for evicting of cached data are required.

6. Conclusions

The underlying query processing architecture for a sensor network is an important consideration affecting its performance, scalability, and applicability. In essence, current and future capabilities of sensor networks mostly depend on the query processing architecture. In this paper, we identify the design objectives and challenges related to query processing over sensor networks and to this end investigate the key querying techniques with their relative advantages and limitations. The analysis of these techniques has enabled us to set a plethora of future research directions. The emphasis is placed on interoperability and integration issues related to conventional and wide area sensor networks databases with an aim of pervasive deployment of sensor networks applications.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[2] S. Babu and J. Widom. Continuous queries over data streams. *ACM SIGMOD Record*, 30(3):109–120, Sep 2001.

[3] P. Bonnet, J. E. Gehrke, and P. Seshadri. Towards sensor database systems. In *Proc. of the Second Int'l Conference on Mobile Data Management*, pages 3–14, 2001.

[4] D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. B. Zdonik. Monitoring streams - a new class of data management applications. In *In Proc. of 28th conference of Very Large Databases(VLDB)*, pages 215–226, Aug. 2002.

[5] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. R. Madden, V. Raman, F. Reiss, , and M. A. Shah. Telegraphcq: Continuous dataflow processing for an uncertain world. In *CIDR Data Engineering Bulletin*, volume 26(1), Mar 2003.

[6] J. Chen, D. DeWitt, F. Tian, and Y. Wang. Niagaracq: A scalable continuous query system for internet databases. *ACM SIGMOD*, pages 379–390, May 2000.

[7] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Int'l Conference on Distributed Computing Systems*, pages 23–30, Jul. 2002.

[8] A. Deshpande, S. Nath, P. B. Gibbons, and S. Seshan. Cache-and-query for wide area sensor databases. *ACM SIGMOD*, pages 503–514, 2003.

[9] J. Gehrke, F. Korn, and D. Srivastava. On computing correlated aggregates over continual data streams. *ACM SIGMOD*, 30(2):13 – 24, 2001.

[10] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan. Irisnet: An architecture for a world-wide sensor web. *IEEE Pervasive Computing*, 2(4):22–33, Oct.-Dec. 2003.

[11] M. Harren, J. Hellerstein, R. Huebsch, B. Loo, S. Shenker, and I. Stoica. Complex queries in dht-based peer-to peer networks. In *Int'l Workshop on Peer-to-Peer Systems ACM Revised Papers from the First Int'l Workshop on Peer-to-Peer Systems*, pages 242–259, Mar. 2002.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.

[13] S. Madden. *The Design and Evaluation of a Query Processing Architecture for Sensor Networks*. PhD thesis, UC Berkeley, Fall 2003.

[14] S. Madden and M. J. Franklin. Fjording the stream: An architecture for queries over streaming sensor data. In *18th Int'l Conf. on Data Engineering*, pages 555–566, Feb. 2002.

[15] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[16] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *Proceedings of the 2003 ACM SIGMOD international conference on on Management of data*, pages 491–502, 2003.

[17] P. V. Mockapetris and K. J. Dunlap. Development of the domain name system. In *SIGCOMM*, volume 25(1), Jan 1995.

[18] R. Sadri, C. Zaniolo, A. Zarkesh, and J. Adibi. Optimization of sequence queries in database systems. In *PODS*, pages 71–81, May 2001.

[19] M. A. Shah, J. M. Hellerstein, S. Chandrasekaran, and M. J. Franklin. Flux: An adaptive partitioning operator for continuous query systems. In *ICDE*, pages 25–32, Mar 2003.

[20] M. Wahl, T. Howes, and S. Kille. Lightweight directory access protocol. RFC, Standards Track RFC 2251, M. Wahl, Critical Angle Inc., T. Howes, Netscape Communications Corp., S. Kille, Isode Limited, Dec 1997.

[21] Y. Yao and J. E. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM Sigmod Record*, 31(3), Sep 2002.

[22] Y. Yao and J. E. Gehrke. Query processing in sensor networks. In *First Biennial Conference on Innovative Data Systems Research (CIDR)*, Jan. 2003.