

A New Storage Scheme for Approximate Location Queries in Object-Tracking Sensor Networks

Jianliang Xu, *Member, IEEE*, Xueyan Tang, *Member, IEEE*, and Wang-Chien Lee, *Member, IEEE*

Abstract—Energy efficiency is one of the most critical issues in the design of wireless sensor networks. Observing that many sensor applications for object tracking can tolerate a certain degree of imprecision in the location data of tracked objects, this paper studies precision-constrained approximate queries that trade answer precision for energy efficiency. We develop an Energy-conserving Approximate StorageE (EASE) scheme to efficiently answer approximate location queries by keeping error-bounded imprecise location data at some designated storage node. The data impreciseness is captured by a system parameter called the approximation radius. We derive the optimal setting of the approximation radius for our storage scheme based on the mobility pattern and devise an adaptive algorithm to adjust the setting when the mobility pattern is not available a priori or is dynamically changing. Simulation experiments are conducted to validate our theoretical analysis of the optimal approximation setting. The simulation results show that the proposed EASE scheme reduces the network traffic from a conventional approach by up to 96 percent and, in most cases, prolongs the network lifetime by a factor of 2-5.

Index Terms—Energy efficiency, data dissemination, data storage, location query, wireless sensor network.

1 INTRODUCTION

OWING to the rapid advances in sensing and wireless communication technologies, wireless sensor networks have emerged as a promising solution for a wide range of civil and military applications. In this paper, we consider object-tracking sensor networks, one of the most important classes of sensor networks. Example applications of object tracking include wildlife animal monitoring in remote areas and intrusion detection on military sites. Users in these applications are interested in *location queries*, which return the locations of tracked moving objects.

A sensor network is typically composed of a large number of tiny sensor nodes equipped with data processing, sensing, and communication capabilities. The sensor nodes usually operate in an unattended manner and are battery powered. However, replacing the battery is not only costly but also impossible in many situations (for example, in a hard-to-reach area). Thus, energy efficiency is a critical consideration in the design of large-scale sensor networks. There has been significant research on energy-conserving object-tracking sensor networks (for example, [9], [10], [35]). Most of these studies aimed at reducing the number of

sensor nodes activated for tracking an object and/or reducing the location update traffic in providing *accurate* answers to location queries.¹

Imprecision is an inherent property of object-tracking sensor networks. State-of-the-art location positioning technologies such as GPS and triangulation are not error-free. Moreover, many applications are willing to tolerate a certain degree of imprecision or error in the data due to either the application nature or the high resource constraints in sensor networks. As such, here, we take a different approach to improve energy efficiency by exploiting the trade-off between data quality and energy conservation. Instead of always feeding the most accurate answers to location queries, we investigate the problem of providing precision-constrained approximate locations based on user tolerances. In our model, an *approximate location query* is specified by an object identifier and a precision constraint. The sensor network responds with a location bounded by the required precision.

Inspired by [11] and [23], we develop an Energy-conserving Approximate StorageE (EASE) scheme to efficiently answer approximate location queries. Whereas most prior work assumed centralized/designated storage for data collection and query answering [9], [17], [32], EASE innovatively maintains two versions of object location data in the network. High-precision data are kept at some *local storage node* close to a moving object in order to reduce long-distance traffic resulting from *remote updates*. Meanwhile, the same data with a lower precision are replicated at some *designated storage node* that is known to users in order to reduce the query traffic. In the EASE scheme, the imprecision of location data at the designated storage node is

1. The accuracy is achieved by best effort since the object location cannot be 100 percent accurate due to network delay, discrete sampling instances, etc.

• J. Xu is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, KLN, Hong Kong.

E-mail: xujl@comp.hkbu.edu.hk.

• X. Tang is with the School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798.

E-mail: acxytang@ntu.edu.sg.

• W.-C. Lee is with the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802.

E-mail: wlee@cse.psu.edu.

Manuscript received 5 Dec. 2006; revised 26 Mar. 2007; accepted 13 June 2007; published online 6 July 2007.

Recommended for acceptance by J.-P. Sheu.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0393-1206.

Digital Object Identifier no. 10.1109/TPDS.2007.70740.

bounded by an *approximation radius*, which specifies a geographical area in which the low-precision location data are considered valid. In other words, a location update due to object movement will not be sent to the designated storage node if the object remains within the approximation radius. Correspondingly, a query is answered by the designated storage node if its precision constraint is weaker than what is specified by the approximation radius. Otherwise, the query is forwarded to the local storage node for resolution. As such, the EASE scheme attempts to optimize the network performance (in terms of reducing network traffic and energy consumption) by balancing the update traffic and query traffic. This is achieved by properly setting the approximation radius. We derive the optimal setting of the approximation radius based on the mobility pattern and also devise an adaptive algorithm to adjust the setting on the fly when the mobility pattern is unknown or is dynamically changing. We show via simulation experiments that the EASE scheme, working together with the proposed methods for setting the approximation radius, reduces the network traffic by up to 96 percent from a conventional approach and, in most cases, prolongs the network lifetime by a factor of 2-5.

We summarize the contributions made in this study as follows:

- To the best of our knowledge, this is the first study on data dissemination in object-tracking sensor networks that attempts to address the issue of energy efficiency by exploiting the trade-off between data quality and energy conservation.
- An energy-efficient storage scheme, called EASE, is proposed to efficiently answer precision-constrained approximate location queries.
- The setting of the proposed storage scheme is analyzed and optimized through a theoretical study. The theoretical analysis is validated by simulation experiments.
- An adaptive algorithm is proposed to adjust the setting of the approximation radius for unknown/dynamic workloads.
- An extensive performance evaluation is conducted to evaluate the performance of the proposed EASE scheme and the methods for setting the approximation radius.

The rest of this paper is organized as follows: Section 2 reviews related work. The system model is described in Section 3. Section 4 presents the proposed EASE scheme in detail. We analyze the performance of EASE and investigate the setting of the approximation radius in Section 5. Section 6 presents the results of the performance evaluation. Finally, Section 7 concludes the paper.

2 RELATED WORK

2.1 Object-Tracking Sensor Networks

There are two research directions for improving the lifetime of an object-tracking sensor network. One is to reduce energy consumption in the sensing component (for example, [18] and [35]). The basic idea is to activate only the essential sensor nodes needed to track the moving objects while leaving the other nodes in a power-saving mode. In [31] and [33], the

sensor nodes are organized into a cluster-based architecture such that a cluster head calculates object locations based on signal readings from its slave nodes. Based on these studies, we assume that object locations can be obtained by cluster heads and only focus our task on where and how to store the location data in support of energy-efficient approximate location queries.

The other direction, aligned with ours, is to improve energy efficiency by reducing network traffic in disseminating location updates. Nevertheless, the focuses of the prior studies are different from ours. Goel and Imielinski [9] proposed a prediction-based approach. A base station collects sensor readings and periodically generates predictions to be sent back to the sensor nodes. A sensor node reports a location update only when its reading differs from the predicted one. Xu et al. [32] suggested a dual-prediction scheme where a fixed prediction model is deployed at both the base station and the sensor nodes. These studies complement our work in that prediction can be incorporated into our EASE scheme to further reduce remote update traffic. Kung and Vlah [17] investigated continuous location queries and proposed a publish-and-subscribe tracking method. In contrast, we leverage error tolerances to improve network performance for one-shot location queries. Although there has been research on the trade-off between energy conservation and tracking quality (for example, [10], [24]), the trade-off has not been investigated in the dissemination of location data, which is the topic of this paper.

Our work also bears some similarity to location management for mobile networks [4]. As the purpose of location management is to locate roaming users for call delivery, locations are managed at a fixed granularity (that is, cell). In contrast, our architecture is capable of adaptively storing location data at different degrees of accuracy to improve the efficiency of query processing.

2.2 Data Storage and Query Processing

A simple storage model is to have a centralized base station collect and store the sensed data. This approach is good for aggregate data collection (for example, sum, average, maximum, and median) [21], [28], where excessive sensed data can be pruned during aggregation along the routing path, and only short summaries are maintained at the base station. However, this is not efficient for nonaggregate data collection (for example, the location queries considered in this paper). The base station and the sensor nodes around it can easily become hotspots, which would shorten the network lifetime.

Recently, in-network storage has been advocated in many research projects. In the TinyDB project, Madden et al. [20] presented the pull-based acquisitional query processing (ACQP), where the sensor nodes control where, when, and how often data are acquired and delivered to query operators. The Cougar project [7] employed a hybrid pull-push model, in which sensed data are pushed to some selected view nodes, from where the data are pulled to answer queries. Ratnasamy et al. [26] proposed an in-network data-centric storage (DCS) model: sensed data are pushed to the sensor node nearest to some geographical location hashed from a predefined key. Zhang et al. [36] suggested storing sensed data locally. A centric ring-based index was proposed to facilitate query processing. More recently, Lu et al. [19] proposed a spatiotemporal query service called MobiQuery to allow

mobile users to query their surrounding areas through a sensor network. Jiang and Jin [14] developed robust aggregation techniques for extracting statistical information from sensor networks. Unfortunately, none of these prior studies have examined the ability of approximate data storage to improve energy efficiency.

2.3 Approximate Query Processing

Early work on approximate query processing focused on a wired network [23]. Han et al. [11] conducted a pioneering study on answering approximate queries in sensor networks. They developed an efficient data collection protocol to fulfill the application-specified data quality while minimizing the energy consumption of sensor nodes. However, the solutions developed in [11] are not applicable to object-tracking applications. This is because they considered only a simple single-hop system where each sensor node communicates with the server directly, and each target phenomenon is always captured by a fixed sensor. In contrast, we consider a dynamic sensing scenario where the location of a moving object is acquired by different nodes at different times; thus, a cooperative location updating protocol is needed. Moreover, we consider a multihop sensor network. As a result, the cost of location updates/queries also varies according to the locations of moving objects. These differences make our system modeling and performance analysis completely different from those in [11].

Precision-constrained queries [6], [27], [30] have also been studied for in-network data aggregation, which has a different focus from object-tracking sensor networks. Compressing historical sensor readings for transmission also saves energy [5]. However, it is applicable to querying historical data only. In contrast, we consider applications that are interested in querying the current locations of moving objects.

2.4 Data Routing

A sensor network is typically connected by wireless links in an ad hoc manner. To relay data in a sensor network, many routing algorithms have been proposed to address energy efficiency, scalability, and reliability issues. They can be classified into three categories: *data-centric*, *hierarchical*, and *location-based routings* [3]. In data-centric routing (for example, directed diffusion [13]), the sink floods the query to a certain region of interest, and the sources report data to the sink through the route established based on the named query. Due to the high cost of flooding, such a protocol is suitable for long-lived queries only. Hierarchical protocols (for example, LEACH [12]) group the sensor nodes into clusters such that a cluster head performs data aggregation/fusion and communicates with other heads on behalf of the nodes within its cluster. Location-based protocols (for example, greedy perimeter stateless routing (GPSR) [15]) make use of geographical position information to transport data.

3 SYSTEM MODEL

We consider a sensor network consisting of a large number of stationary sensor nodes deployed in some operational area. Each sensor node is aware of its own location, through GPS for example. We assume that the nodes organize themselves into clusters and that every cluster has a cluster head. A

cluster head is more powerful than an ordinary sensor node. It is equipped with some local storage to store data and is also capable of communicating with other cluster heads to exchange data. The sensor nodes in a cluster can work together to recognize and track the objects in their vicinity; for example, a cluster head can triangulate object locations based on signal readings from its slave nodes [31], [29], [33]. The object locations are sampled at a fixed sampling rate. We also assume that each moving object being tracked has a unique identifier. Since this paper aims at energy-efficiently storing and disseminating object location data in support of approximate location queries, we shall focus on reducing communication among cluster heads. Unless explicitly specified, a sensor node refers to a cluster head in the rest of this paper.

Approximate location queries. The sensor network under consideration supports a large number of users making one-shot queries for the locations of moving objects. The queries can be made via a sensor node (known as the *querying node*) from anywhere in the network. Each approximate location query is specified by a tuple $\langle object_id, p \rangle$, where *object_id* is the identifier of the target object, and *p* is the error in object location that the query can tolerate.²

Local storage (LS) and centralized storage (CS). Intuitively, the object location data can be stored at 1) the sensor node that detected the object or 2) a centric storage node, which could be either a centralized base station or a sensor node determined by the DCS scheme [26] (see Fig. 1a).³ If the location data are stored locally at the detecting node (known as LS), a query that wants to find the location of some object has to be *flooded* over the whole network. Thus, the query cost is high. In contrast, if the CS scheme is adopted in the system, any location update of a moving object should be sent to the centric storage node. This results in high update traffic. In the next section, we propose a hybrid in-network storage scheme that achieves a good balance between the query cost and the update cost for approximate location queries. Without loss of generality, we shall assume for the rest of this paper that the DCS scheme [26] is employed for CS.

4 EASE: ENERGY-CONSERVING APPROXIMATE STORAGE

This section proposes the new EASE scheme that takes advantage of the error tolerances of queries. We first give an overview of EASE in Section 4.1. Section 4.2 describes the location updating protocol for working with EASE. Finally, we discuss how EASE handles node failures and message losses in Section 4.3.

4.1 Overview

The EASE scheme attempts to cut down the update traffic by maintaining two versions of the location data for each object: an accurate version at a local storage node and an

2. For simplicity, we assume in this paper that *p* is the error tolerance in addition to the unavoidable system error such as the inaccuracy of the positioning technique. In practice, *p* can be determined by the user's error tolerance minus the maximum system error.

3. In DCS, the centric storage node of an object is determined by applying a predefined hash function to the object identifier.