Information Fusion 15 (2014) 44–55

Contents lists available at SciVerse ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

On query processing in wireless sensor networks using classes of quality of queries

Angelo Brayner *, André L.V. Coelho, Karina Marinho, Raimir Holanda, Wagner Castro

Graduate Program in Applied Informatics, University of Fortaleza (UNIFOR), Brazil

ARTICLE INFO

Article history:
Received 23 April 2010
Received in revised form 1 July 2011
Accepted 31 January 2012
Available online 9 February 2012

Keywords:
Wireless sensor networks
In-network query processing
Quality of queries
Novelty detection
Energy consumption

ABSTRACT

This paper introduces the concept of quality of queries (QoQs) towards a more adaptive query processing in wireless sensor networks (WSNs). This approach aims at the intelligent consumption of the limited resources (energy and memory) available in these networks while still delivering a reasonable level of data quality as expected by client applications. In a nutshell, the concept of QoQ stipulates that the results of different queries injected into the same WSN can be tailored according to different criteria, in particular the levels of query result accuracy and energy consumption. For this purpose, four classes of QoQ (CoQoQ) are specified having in mind distinct requirements in terms of these criteria. To allow the implementation of these classes in a real WSN setting, a new novelty-detection based algorithm, referred to as AdaQuali (which stands for "ADAptive QUALIty control for query processing in WSN"), is also proposed in a manner as to control the sensor node activities through the dynamic adjustment of their rates of data collection and transmission. In order to validate the novel approach, simulations with a prototype implemented in Sinalgo have been conducted over real temperature data. The results achieved evidence the suitability of the proposal and point to gains of up to 66.76%, for different CoQoQ, in terms of reduction in energy consumption.

1. Introduction

Sensors are resource-constrained devices used to collect data from the environment aiming at detecting or measuring physical phenomena. Advances in wireless communication have enabled the development of massive-scale wireless sensor networks (WSNs). In a WSN, sensors are usually scattered in the network and use low-power communication channels; they disseminate the collected data to a base station, from where the information (query) was originally requested. Potential WSN applications include environmental monitoring (e.g., traffic and habitat), industrial sensing and diagnostics (e.g., factory and supply chains), infrastructure protection (e.g., water distribution), battlefield awareness (e.g., multi-target tracking) and context-aware computing (e.g., intelligent home) [1]. The greatest challenge for WSN applications relates to hardware constraints in sensor nodes: limited power, computational capacity, and memory size. The limited energy supply in sensor batteries, in particular, directly influences the sensor node's lifetime.

Three activities usually have a major effect on energy consumption: (i) sensing (data collection from the environment), which is the primary goal of a sensor; (ii) communication (sending and receiving packets), which is essential to set up a WSN; and (iii) data processing, which involves all computational operations necessary to manipulate data in a WSN. However, it is well known that communication is the activity responsible for the bulk of the power consumption, thus being the main point of attention when designing algorithms for sensors. By reducing communication costs, energy may be drastically saved, consequently increasing the WSN's lifetime.

An effective strategy to reduce energy consumption is thus to reduce the number of messages sent across the network. This can be achieved by introducing database query processing (QP) and in-network aggregation (fusion) techniques [2–4] into WSN. While the distributed query engines proposed by Brayner et al. [5] and Madden et al. [6] are representatives of the former, ADAGA [7] and TinyDB [6] relate to two effective mechanisms based on in-network aggregation.

An alternative way that also shows to be worth in reducing the number of messages exchanged across WSN is oriented to the identification of *outliers* [8–10] and follows the idea of inducing a model directly from collected data in order to characterize the normal profile of values associated with sensed phenomena. As a result of this *novelty detection* (ND) perspective [11–13], only streamed data considered to be (ab)normal in some sense need to be effectively forwarded by the sensors.

Although interesting, the aforementioned query processing and in-network aggregation models fall short in not considering the specificities associated with the different client applications that

* Corresponding author. Fax: +55 85 34773061.
E-mail addresses: brayner@unifor.br (A. Brayner), acoelho@unifor.br (A.L.V. Coelho), karina@unifor.br (K. Marinho), raimir@unifor.br (R. Holanda), wagner@unifor.br (W. Castro).

1566-2535/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2012.01.010

may have access to the data flowing in a WSN. This is because the query engines proposed for WSN do not consider the different requirements that may be imposed by distinct applications with respect to the quality of the result of a given query. For example, in some cases, it is reasonable to adjust the time interval of data collection (e.g., instead of collecting data every 5 s, as initially defined in a query, the data may be collected every 6 s). By doing this, the volume of sensed data injected into the WSN can be significantly reduced and, consequently, the energy consumed by the whole network. The negative side effect is that reducing the data volume may impact the accuracy of the query result.

To properly take advantage of this sort of strategy for adaptive query processing in WSN, this paper introduces a novel approach that is rooted in ND and *quality of queries* (QoQs). The gist of the concept of QoQ is to allow that different client applications have their queries processed in different ways in the same WSN according to different criteria. By this means, peculiarities and requisites imposed by applications can be taken into account when queries are simultaneously processed by a given WSN.

In this paper, in particular, two QoQ criteria have been investigated, namely, the accuracy of the query result and the efficiency in energy consumption. These criteria were selected mainly for two reasons. First, because both can be controlled by adjusting a given parameter, either the temporal pace of data collection (sensing) or the time interval between data transmissions, respectively. Second, because they are complementary and conflicting in nature: An increase in the query result accuracy typically implies an increase in the levels of energy consumed. Thus, for any given query injected into the WSN, a proper tradeoff between these criteria needs to be set beforehand, which is made possible by resorting to the notion of *classes of QoQ* (CoQoQ).

The key goal of using the notion of CoQoQ is to capture distinct levels of tradeoff between any set of QoQ criteria. In this paper, four classes have been specified, in order to strike different balances between query result accuracy and energy consumption. By this means, it would be possible, for example, to specify that a given query should be serviced with less accuracy w.r.t. its result for the sake of energy conservation of the whole WSN. By customizing the query processing through different CoQoQ, we claim that significant gains in terms of energy consumption may be achieved, mainly in those cases where the application requirements are less stringent.

In order to adapt the behavior of the sensor nodes according to the specification of different CoQoQ, the AdaQuali algorithm (after "*ADAptive QUALIty* control for query processing in WSN") was conceived to operate within the query engine component running on sensor nodes belonging to a given WSN. The source of inspiration of this algorithm lies in other works that already exploit the idea of online ND (that is, the ability of recognizing significant variations, anomalies, or even novel concepts in data streams) in the realm of WSN [8–13]. AdaQuali's main functionality is to adjust sensing and transmission activities of the sensor nodes in a dynamic fashion by adapting the balance between resource (energy and memory) consumption and the levels of query result quality as expected by the client applications. As far as we know, a query processing approach based both on QoQ and ND has not been deeply undertaken yet in the context of WSN, and the main goal of this paper is thus to provide a detailed investigation in such regard.

For empirically validating the novel approach proposed in this paper, simulations with a prototype implemented in Sinalgo [14] have been conducted over real temperature data made available by the Intel Berkeley Research Laboratory [15]. Overall, the results achieved point to significant gains in energy efficiency that vary for different CoQoQ, evidencing the suitability of the approach based on the concepts of QoQ and ND for in-network query processing in WSN.

The remainder of this paper is structured as follows. Section 2 outlines important concepts and strategies linked to query processing in WSN, whereas Section 3 focuses specifically on related work investigating quality of service (QoS) issues in this context. Section 4 presents in detail the proposed QoQ-based approach for in-network query processing. Thus, four CoQoQ are specified in Section 4 in terms of data accuracy and energy consumption requirements, relying on the proper characterization of normal and abnormal data. A step-by-step description and a complexity analysis of the AdaQuali component are also provided in this section. Simulations with a Sinalgo prototype operating over real sensor data are reported in Section 5. Section 6 concludes the paper and provides remarks on future work.

2. In-network query processing

There is a growing interest in query processing for WSN, mainly because in-network data filtering (by means of selection and projection operations) and aggregation have shown to be a suitable approach to reduce energy consumption in WSN when propagating data from sensors to base station. In this context, TinyDB [6] is a distributed query engine developed for WSN composed of sensors running the TinyOS operating system. TinyDB processes queries expressed through an acquisitional query language, which has some simple extensions to SQL for controlling data acquisition. In particular, Madden et al. showed how acquisitional issues influence query optimization, dissemination and execution.

Chatterjea et al. [16] also described a distributed query engine for processing queries in WSN. The essence of this approach is to allow sensor nodes to execute queries and aggregation operators on local data. By doing this, the proposed query engine may apply filters on sensors' data, thus reducing the volume of data injected into the network. However, the query engine proposed by Chatterjea et al. does not decompose queries into sub-queries. It is important to observe that decomposing queries into sub-queries may significantly reduce the data volume injected into the network. This is because the use of sub-queries makes possible the application of (i) reduction techniques (for projection, aggregation, selection and join operations of relational algebra) to sub-queries before disseminating the query to the sensors and (ii) in-network data filtering (selection and projection operations of relational algebra) [5,6].

Brayner et al. [5] proposed a distributed in-network query engine, called WSNQE, which distributes the burden of executing a given query over the base station(s) and sensor nodes of a WSN. Additionally, the authors showed that WSNQE is able to handle common conditions present in WSN, such as failures, resource limitations (e.g., energy and memory), the existence of large amounts of data streams, and mobility of the sensors. This is the engine within which the AdaQuali component described in Section 4 has been implemented.

The strategy behind WSNQE is to process a query in four distinct phases: parsing, query decomposition, fragment execution, and post-processing. The parsing, query decomposition, and post-processing phases are executed on the base station. Conversely, the fragment execution phase runs on the sensor nodes of a WSN. Next, we briefly describe how a query is processed by WSNQE. For an in-depth discussion, the reader is referred to the work of Brayner et al.

For specifying declarative queries to be processed by the WSNQE, a query language, called Sensor Network Query Language (SNQL), was devised. SNQL enables the tuning of several parameters, such as the volume of data to be injected into the network and the levels of accuracy of the query results. Due to these functionalities, SNQL was adopted for running the simulations reported in this paper.