Toward a Flexible Data Management Middleware for Wireless Sensor Networks*

Razia Haider[1], Federica Mandreoli[1,2], Riccardo Martoglia[1], Simona Sassatelli[1] and Paolo Tiberio[1]

Abstract. In this paper we present the research activity we are carrying out in the "Mobile Semantic Self-Organizing Wireless Sensor Networks" Project at the Department of Information Engineering of the University of Modena and Reggio Emilia. In this context, the main aim of our research is to study solutions for the flexible querying of distributed data collected by heterogeneous devices providing measurement readings. To this end, we propose a middleware for wireless sensor networks which is able to autonomously configure the communication and the operations required to each device in order to reduce energy and temporal costs.

Introduction

In recent years, the advances made in the miniaturization, processing, storage and communication technologies have allowed the creation of new families of small and cheap devices capable of wireless communication and significant computation. Thanks to their peculiarities, these devices may be distributed in large quantities in the environment in order to perform data collection, goods movement control or, broadly speaking, to implement new and advanced forms of interaction with the world around them. Sample fields of application include domotics, logistics, biomedicine, remote control systems, distributed sensing of environmental phenomena and more general applications of measurement, detection and monitoring in "ambient intelligence" scenarios. Wireless Sensor Networks (WSN) represent one of the best known and widespread technologies of this kind.

This is the stimulating scenario of the "Mobile Semantic Self-Organizing Wireless Sensor Networks" three-years project at the Department of Information Engineering of the University of Modena and Reggio Emilia. This research activity adopts an interdisciplinary approach by integrating aspects in the areas of computer science, electronics and telecommunications such as design of microelec-

---

* This work is partially supported by the "Mobile Semantic Self-Organizing Wireless Sensor Networks" Project.

[1] DII - Univ. of Modena and Reggio Emilia, Italy, {firstname.lastname}@unimo.it
[2] IEIIT - BO/CNR, Bologna, Italy

tronic systems, ad-hoc radio communications, digital information management, applied electromagnetism and electronic measures. The high profile scientific and technological challenge is to deploy a mobile network of intelligent radio sensors, requiring enormous inputs from all the traditional disciplines of ICT.

In this context, our research activity is focused on data management and, in particular, on studying solutions for the flexible querying of distributed data collected by heterogeneous devices providing measurement readings.

In order to allow a flexible querying of sensor data, in this paper we present a specifically devised Data Management Middleware which is able to autonomously configure the communication and the operations required to each device in order to satisfy the temporal, energetic and accuracy requirements of the specific usage scenario. Users can express their informative needs by composing queries in declarative form, such as "Return the maximum value of the refrigerator temperature each hour for a day, with an accuracy of 0.1%". The middleware receives the queries from user devices, abstracting from all technical issues related to the communication with the specific sensors and devices in use. Then, it produces execution plans maximizing power consumption and temporal efficiency w.r.t. the desired measure accuracy, and finally executes the query, gathering its results and presenting them to the users. Further, the middleware may enact network topology reconfiguration and selectively reprogram some of the nodes capabilities, in order to better distribute communication and computation load.

This paper is organized as follows: we first analyze the state of the art on sensor data management, focusing on query processing issues arising in our project; then, we present our data management middleware for wireless sensor networks, describing the different functionalities and how the specific modules interact in order to provide them; finally, we conclude by introducing possible application scenarios.

Related Work

Wireless sensor networks have been a very active area of research in latest years (see [1] for a survey) and this trend has led the database community to begin a number of research activities focusing on the different aspects of managing sensor data [2-13].

Some recent works [2, 3] have proposed the powerful vision of a sensor network as a distributed database which is programmed and queried by means of a declarative language. The vision of declarative querying is attractive since it allows programmers to "task" an entire network of sensors nodes, rather than requiring them to worry about programming individual nodes. As an example, the Cougar Project [2] treats the entire sensor network as a single streaming database where data collection is performed using declarative queries, allowing the user to focus on the data itself, rather than on the collection operation. Another notable example of such kind of systems is the TinyDB Project [3]. It uses an acquisitional query processing approach which requests data to sensors on the basis of the current query corpus. The adopted SQL-style language allows both data collection and aggregation. Also the work in [4] proposes a deductive framework where an even more expressive language for programming high-level applications is used. None of these projects, however, copes with issues related to the heterogeneity of the specific sensors and devices in use.

One of the most important constraints when dealing with a network of sensors is power consumption. Indeed, because communication is orders of magnitude more expensive than local computations, several research groups have focused on optimized in-network query processing (that is, the pushing of operations into the network) as a means of reducing energy consumption.

As an example, in the Cougar Project, given a user query, a central query optimizer generates an efficient query plan aiming at minimizing resources usage within the network by means of in-network processing performed at specialized sensor nodes, called *leaders*, which are statically chosen when sensors are deployed [2]. A more dynamic roles differentiation and network topology adaptation has instead never been studied.

Also the Acquisitional Query Processor (ACQP) of TinyDB adopts energy-efficient techniques which minimize resource usage by tuning frequency and timing of data sampling [5]. The sensor network project at USC/ISI Group has then proposed Direct Diffusion [6], an energy-efficient data dissemination paradigm which is data-centric and offers reinforcement-based adaptation and in-network data aggregation and caching. There has been some work on operator placement [7] too, but a number of challenges still remain.

Particularly interesting are issues concerning heterogeneity [8], which involve, for example, the choice of where to place operators given that the nodes in the network may have different processing power or battery life and may be experiencing different computational or communication loads. Dealing with dynamic heterogeneity, such as variations in load or energy charge, suggests some form of adaptive query optimization which has not been investigated till now. Further, the possibility to selectively send some specific execution code to few properly chosen sensors during network operation has not yet been explored.

Another key issue concerning sensor data is that they usually contain incomplete and noisy measurements of environmental phenomena, such as temperature and light, which are continuous in both time and space. This problem is coupled with different sources of noises induced by the transmission process. Then, statistical analysis and probabilistic modeling are perhaps the most suitable solutions for appropriately managing such kind of uncertain data.

Regarding modeling, the work in [9, 10] presents the BBQ system which improves TinyDB by building statistical data models which capture correlations among attributes and attribute value changes; the aim is to enable reduced sensing rates while meeting a query-specified confidence. Ken [11] equally deals with the issue of reducing power consumption, this time by using replicated dynamic probabilistic models to minimize communications in the sensor network. A framework for representing uncertainty of sensor data is presented in [12] which also proposes techniques for qualitatively and quantitatively representing answers imprecision. The work in [13] goes further and quantifies uncertainty of query results as noisy data pass through various processing stages. In these approaches, the data uncertainty is due to the sampling process [9, 10], to the low communication rate among sensors [11], or to the outdated data used for answering queries [12]. Nevertheless, none of them seems to model accuracy by exploiting information on the sensors current status and on the inherent uncertainty due to the physical measurement process.

A Flexible Data Management Middleware

Querying distributed data related to measurement readings collected by sensor devices is currently not a simple operation for users. Even for very simple informative needs, such as "The mean temperature of room A over the last hour", to formulate and execute the appropriate query can be a very difficult task. Indeed, sensor devices only react to low level commands and can not understand and process declarative queries. Thus, in order to query sensor data, users would be required to know how data is acquired, what sensors are available, what is their power status and what are the specific protocols, languages and communication modes each of the sensors supports; this would clearly be unfeasible. As a further complication, we should note that heterogeneity strongly comes into play not only from sensors but also from the devices requesting the information, which could be computers, PDAs, mobile phones, and so on.

In order to allow a flexible querying of sensor data, we envision a specifically devised Data Management Middleware (Figure 1a) which is interposed between user devices (application layer) and sensors (data source layer) and is able to autonomously configure the communication and the operations required to each device in order to satisfy the user needs, while always reducing energy and temporal costs. In particular, the middleware should:

- communicate with user devices, providing a simple unified interface to compose declarative queries while abstracting from all technical issues related to the specific sensors and devices in use;
- parse the query and find the best (i.e. temporally and energetically efficient) execution plan, possibly rewriting the query for a balanced distributed execution;
- configure the network nodes and topology w.r.t. the specific computation and communication needs (*auto-adaptivity*);
- execute the query on the network and present the final results to the user.

Since the middleware has to support different user needs, our design provides different query types: *instant queries* (involving specific measures to be acquired at the time the query is executed, and possible calculations to perform on them, such as “return the current average noise level in the factory premises”), *event-based queries* (involving specific trigger events, such as “return the temperature of all the rooms when one of the temperatures exceeds 40C”) or *lifetime-based queries* (involving lifetime clauses, such as “return the light level of the office each minute for one day”). Further, in all situations, the user should be able to specify the level of measure *accuracy* which is most suited to the application requirements. The middleware will thus produce a plan that best satisfies all user needs: for instance, the execution of a lifetime query should primarily focus on optimizing power consumption w.r.t. the given accuracy goal, while an instant query should also achieve a satisfying temporal efficiency.

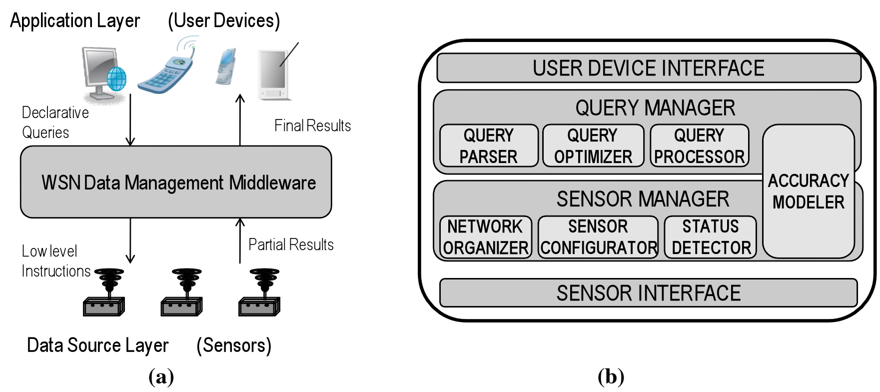

Fig. 1. WSN Data Management Middleware overview (a) and inner organization detail (b)

In order to achieve such a vision, our middleware has been designed in a modular way. Let us analyze the composing modules, depicted in Figure 1b, and their specific functionalities:

User device interface. It exploits a web and application server which user devices connect to. In this way, just by using a simple browser, users can easily express their requirements and compose their query with an easy to use interface; finally, the module outputs the query in declarative language form.

Query manager. This module offers all the functionalities required for understanding the query (*query parser*), producing execution plans maximizing power consumption and temporal efficiency w.r.t. the given accuracy goal (*query optimizer*), and finally executing the query and gathering its results (*query processor*). The *accuracy modeler*, whose services are shared with the sensor manager module, contributes with the probabilistic modeling of the measurement accuracy, which includes the inherent uncertainty due to the physical measurement process and the communication noise. It thus helps the query optimizer to understand the current sensors’ capabilities and the query processor to answer queries based on accuracy figures associated to the data.

Sensor manager. It is in charge of managing the sensor network and configure it for specific requirements. In particular, the *network organizer* module takes care of the network topology: it interacts with the query optimizer to automatically derive the best topology for executing a query. If a network reconfiguration is advisable, the *sensor configurator* is able to enact it. Further, it can re-configure each of the sensors' communication modes (for instance streaming or on-demand mode) and even send them specific software allowing them to make ad-hoc computations, thus maximizing the exploitation of the limited capabilities of each device. Finally, the *status detector* monitors the status of the different sensors, desumes their availability, i.e. power level, signal strength, etc., and makes the query optimizer aware of it.
Sensor interface. This module takes care of handling all the actual low-level communications with the sensors, managing and understanding all the relevant protocols in use (e.g. ZigBee, Bluetooth, etc.)

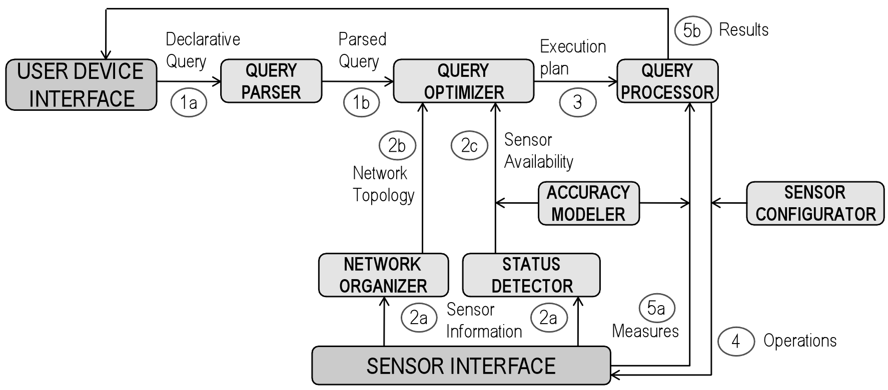

Fig. 2. How the middleware modules handle a query

Let us conclude our analysis by following in detail the interactions and the steps performed by the different middleware modules after a query is issued (Figure 2):
Step 1: The query is issued through the user device interface and is sent to the query parser (1a), which then sends it in parsed form to the query optimizer (1b);
Step 2: The query optimizer gathers all the information needed for its computations. First, it analyzes the query and identifies the requirements, such as the kinds of computations and functions employed and the desired accuracy level. Then, it analyzes the current status of the sensor network: this includes the network topology (2b) and the availability of each sensor (2c) (i.e. power level, signal stregth and an estimate of measure accuracy), which are derived from the sensor information acquired at the sensor interface (2a);
Step 3: The query optimizer analyzes the gathered information and produces different execution plans, evaluating which sensors to involve, what kinds of communication modes to employ, and, for complex queries, how to re-organize net-

work topology and distribute the required computations among the available devices. This could also involve the reformulation of the query following algebraic properties (for instance, a mean could be split into sum and cardinality operations) and reprogramming the software of specific sensors. The execution plans are then sorted according to temporal and/or energetic efficiency, and the best one is sent to the query processor;

Step 4: The query processor enacts the selected execution plan, by communicating with the sensor configurator and sensor interface to configure the network and execute the query;

Step 5: The measurements and partial results are then sent back to the query processor together with their accuracy figures (5a); then, the query processor performs possible final computations and sends the results to the user device interface (5b).

Application Scenarios and Concluding Remarks

In this paper, we presented the results of the study we conducted in the "Mobile Semantic Self-Organizing Wireless Sensor Networks" Project which includes the preliminary proposal of a flexible data management middleware we are currently working on.

Thanks to a variety of upcoming collaborations and supported by the newly established Wireless Sensor-Network Laboratory (WiSe-NetLab) of our Department and its recently ad-hoc engineered Alpha Node sensor, in the near future we plan to thoroughly test our sensor network middleware in a wide range of application scenarios. Among the planned applications, our proposed data management middleware could be used for medical care purposes with some specific hardware equipment devised for medical care. In hospitals, patients could have attached wearable wireless sensors to their bodies that would allow the doctors and nurses to continuously monitor their status. In a disaster scenario or an emergency, this technology would enable medics to more effectively care for large numbers of patients.

The proposed system can also be employed in the military field, where the same technology would allow the captain to know about his soldiers' conditions. The soldiers would have attached specific wireless sensors to their bodies, helmets and weapons, monitoring their body temperature and body position (e.g. standing, laying down, etc.).

Further possible applications of the proposed system could be in industrial working environments. In manufacturing, wireless sensors could be attached to specific points of processing pipelines in order to constantly measure the temperature, pressure etc. during the manufacturing process. Depending on the speed of the production line, very high frequency data rates should be monitored in real time and, from the performed analysis, possibly critical situations could be identified or even predicted as soon as possible. Finally, sensors could also enhance the security of the workers in dangerous environments, such as construction sites, for instance by constantly monitoring if they are wearing all their protecting gear, such as belts or helmets.

All these application scenarios will benefit from the flexibility and optimization capabilities of our middleware, each one primarily focusing on specific requirements, such as, high measurement accuracy for medical care or fast response time for military.

References

1. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4): 393-422.

2. Yao, Y. and Gehrke, J. (2002). The Cougar Approach to In-Network Query Processing in Sensor Networks. *SIGMOD Record,* 31(3): 9-18.

3. Madden, S., Franklin, M. J., Hellerstein, J. M. and Hong, W. (2005). TinyDB: An Acquisitional Query Processing System for Sensor Networks. *ACM Transaction On Database Systems (ACM TODS)*, 30(1): 122-173.

4. Gupta, H., Zhu, X. and Xu, X. (2009). Deductive Framework for Programming Sensor Networks. In *Proc. of the 25th International Conference on Data Engineering (ICDE)*, 281-292.

5. Madden, S., Franklin, M. J., Hellerstein, J. M. and Hong, W. (2003). The Design of an Acquisitional Query Processor For Sensor Networks. *SIGMOD Conference*, 491-502.

6. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J. S. and Silva, F. (2003). Directed Diffusion for Wireless Sensor Networking. *IEEE/ACM Transaction on Networking*, 11(1): 2-16.

7. Bonfils, B. J. and Bonnet, P. (2004). Adaptive and Decentralized Operator Placement for In-Network Query Processing. *Telecommunication Systems*, 26(2-4): 389-409.

8. Hellerstein, J. M., Hong, W. and Madden, S. (2003). The Sensor Spectrum: Technology, Trends, and Requirements. *SIGMOD Record,* 32(4): 22-27.

9. Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J. M. and Hong, W. (2004). Model-Driven Data Acquisition in Sensor Networks. In *Proc. of the 30th International Conference on Very Large Data Bases (VLDB)*, 588-599.

10. Deshpande, A., Guestrin, C. and Madden, S. (2005). Using Probabilistic Models for Data Management in Acquisitional Environments. In *Proc. of the 2nd Biennial Conference on Innovative Data Systems Research (CIDR)*, 317-328.

11. Chu D., Deshpande, A., Hellerstein, J. M. and Hong, W. (2006). Approximate Data Collection in Sensor Networks Using Probabilistic Models. In *Proc. of the 22nd International Conference on Data Engineering (ICDE)*, 48-59.

12. Cheng, R. and Prabhakar, S. (2003). Managing Uncertainty in Sensor Database. *SIGMOD Record*, 32(4): 41-46.

13. Diao, Y., Li, B., Liu, A., Peng, L., Sutton, C., Tran, T. and Zink, M. (2009). Capturing Data Uncertainty in High-Volume Stream Processing. In *Proc. of the 4th Biennial Conference on Innovative Data Systems Research (CIDR).*