Middleware for On-Demand Access to Sensor Networks

Sanem Kabadayi
Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712
s.kabadayi@mail.utexas.edu

Abstract

As sensor networks become increasingly ubiquitous, we envision an instrumented environment that can provide varying amounts of information to mobile applications immersed within the network. Such a scenario deviates from existing deployments which are often highly application-specific and funnel information to a central collection service for a single purpose. Instead, we target future scenarios in which multiple mobile applications will leverage sensor network nodes opportunistically and unpredictably. To address the challenges introduced by this style of interaction, we introduce a middleware platform that enables the development of these adaptive mobile applications. This middleware attempts to overcome the challenges through the use of *scenes* (for locality of interactions) and *virtual sensors* (to ease programming complexity and efficiently use network resources).

1. Introduction

Sensor networks have emerged as an integral component of pervasive computing. Such networks consist of numerous miniature, battery-powered devices that communicate wirelessly to share information about the instrumented environment. While many new concerns arise in comparison to existing distributed or mobile computing scenarios, potential applications of this technology abound and range from intrusion detection to habitat monitoring.

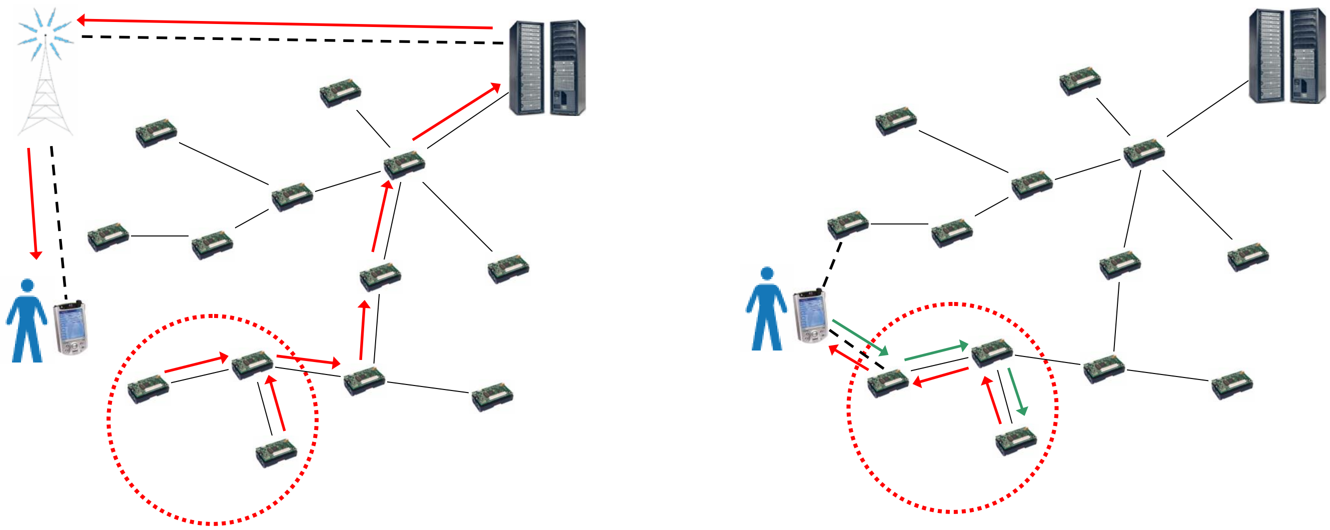
Instead of using sensor networks for their remote data collection, our work highlights future scenarios in which multiple mobile applications will leverage sensor network nodes opportunistically and unpredictably. In current interactions with sensor networks (as shown in Figure 1(a)), a user's query is physically detached from the sensor field and the resolution of a query relies on a centralized collection point, generally the root of a routing tree constructed over the network. However, in *immersive sensor networks*, an application running on the user's device needs to seamlessly connect to the sensors in the local region, removing the requirement that physically distant sensors participate in the query's resolution (as shown in Figure 1(b)). This style of interaction differs from common uses of sensor networks, introducing several unique challenges and heightening existing ones:

- *Locality of interactions:* In immersive sensor networks, an application interacts directly with local sensor nodes. While this can minimize the communication overhead and latency, it can also be cumbersome with respect to enabling the application to precisely specify the area from which it collects information.
- *Mobility-induced dynamics:* While the sensor nodes are likely stationary, the application interacting with them runs on a device carried by a mobile user. Therefore, the device's connections to particular sensors and the area from which the application desires to draw information are subject to constant change.
- *Unpredictability of coordination:* Immersive sensor networks demand that the network be general-purpose. As such, few *a priori* assumptions can be made about the needs or intentions of applications, requiring the network to adapt to unexpected and changing situations.
- *Complexity of programming:* In immersive sensor networks, programming burdens are magnified due to the challenges described here. In addition, the desire to provide end-user applications (as opposed to more database-oriented data collection) increases the demand for applications and the number of programmers that will need to construct them.

The confluence of these challenges necessitates a flexible yet expressive programming environment that enables application development for immersive sensor networks while paying careful attention to resource constraints. To meet the needs of the intended applications, a middleware is required that allows an application running on the user's device to seamlessly connect to the sensors in the local region, removing the requirement that physically distant sensors participate in the query's resolution.

To address these issues, we are creating a new middleware platform, DAIS [1] (Declarative Applications in Immersive Sensor networks), that provides programming abstractions tailored to ubiquitous applications. This is not a middleware for sensor networks in the sense that it runs strictly on the sensors. Instead, it allows developers to create applications

[1] DAIS (dā′ĭs): from the middle English word meaning "raised platform"



(a) (b)

Figure 1. Comparison of (a) existing operational environments with (b) the DAIS approach.

that run on *client devices* (e.g., laptops or PDAs) that interact directly with embedded sensor networks.

DAIS overcomes the challenges described above through the use of *scenes* (that enable local interactions) and *virtual sensors* (that ease programming complexity and promote network efficiency). The specific novel contributions that will result from this work can be categorized into model-level and implementation-level contributions. We are creating a new coordination model defined by the scene abstraction and the virtual sensors abstraction. The model will be realized in a middleware implementation whose performance and usefulness will be practically evaluated.

Section 2 of this paper gives an overview of related projects. In Section 5, we discuss how the two abstractions described in Sections 3 and 4 support the resulting middleware model.

2. Related Work

Existing work has highlighted several design tenets that a middleware for wireless sensor networks must adhere to [11], and the DAIS platform attempts to follow these guidelines. Other projects have also undertaken similar efforts, and we highlight a few of these systems.

As one example of a middleware for pervasive computing, GAIA [9] introduces *Active Spaces* that can be programmed into pervasive computing applications. However, the model assumes a centralized and heavyweight system structure which is in direct opposition to the goal of middleware for coordinating with wireless sensor networks.

Projects targeted directly for sensor networks have often explored representing the sensor network as a database. Two demonstrative examples are TinyDB [7] and Cougar [10]. Generally these approaches enable applications with data requests that flow out from a central point (i.e., a base station) and create routing trees to funnel replies back to this root. Much of the work within these approaches focuses on performing intelligent in network aggregation and routing to reduce the overall energy cost while still keeping the semantic value of data high.

VM⋆ [6] provides a virtual machine approach directed at handling device heterogeneity. While this is also an important concern in DAIS, VM⋆ assumes situations where the application and its needs can be known in advance so that the VM deployed can be optimized with respect to the application and device. TinyGALS [1] allows programmers to represent applications in terms of relatively high-level components which are subsequently synthesized into the low-level, lightweight, efficient programs that are deployed on the nodes. While such approaches are highly beneficial when the application is known and the networks are relatively application-specific, they do not map well to immersive sensor networks where the nodes must be able to service a variety of unpredictable applications.

More generalized approaches such as EmStar [4], SNACK [5], and Agilla [3] attempt to provide integrated suites of tools that enable simplified programming of sensor networks. DAIS differs from these approaches in that it perceives the sensor network not as a data repository but as an instrumented environment that has the capacity to augment a user's pervasive computing experience. Similarly, TinyLIME [2] is a tuple space based middleware that enables mobile computing devices to interact with sensor data in a manner decoupled in both space and time. TinyLIME provides only single-hop connections to sensors and assumes that the sensors do not communicate among themselves.

This effectively places all of the burden of aggregation on the shoulders of the application developer.

3. Localized Interactions Through Dynamic Group Formation

In an instrumented sensor network, a user's operational context is highly dynamic. That is, the set of available data sources changes based on the user's location and movement. Furthermore, if the sensor network is well-connected, the user's device will be able to reach vast amounts of raw information that must be filtered to be usable. For an efficient approach, the application must be able to limit the scope of its interactions to include only the data that matches its needs. In the proposed middleware model, an application's operating environment (i.e., the sensors with which it interacts) will be encapsulated in an abstraction called a scene.

By its definition, a scene will constrain which particular sensors may influence an application. The constraints may be on properties of hosts (e.g., battery life), of network links (e.g., bandwidth), and of data (e.g., type). These types of constraints offer generality and flexibility and provide a higher level of abstraction to the application developer. The declarative specification defining a scene will allow a programmer to describe the type of scene he wants to create, without requiring him to specify the underlying details of how it should be constructed. The programmer only needs to specify the following three parameters for the scene: 1) the *metric*, a quantifiable property of the network or physical environment that defines the cost of a connection (i.e., a property of hosts, links, or data); 2) the *aggregator*, an aggregation function (e.g., SUM, AVG, MIN, MAX) that operates on link weights in a network path to calculate the cost of the path; and 3) the *threshold*, the maximum allowed value for the cost calculated to a sensor for the sensor to be considered a member of the scene. The algorithm for constructing a scene is represented in Figure 2.

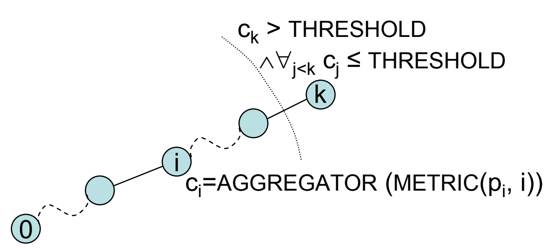


Figure 2. Scene construction algorithm

The scene concept takes these dynamics into consideration and does not require constant proactive behavior of any sensor that may belong to a region. The goal is to minimize the overall communication cost incurred by having a lightweight collection algorithm that reacts directly to a user's actual movement. Within this area, research will focus on topics ranging from programming abstractions for providing the scene abstraction to developers to the creation and evaluation of novel algorithms to dynamically construct scenes over unpredictable, resource-constrained networks.

4. Virtual Sensors: Expressive In-Network Processing

In existing deployments of sensor networks, data collection schemes commonly require sensors to relay raw data to sink nodes to perform further processing. This is not very efficient considering the resource constraints of sensor networks. Sensor network aggregation mechanisms offer in-network data processing algorithms that are successful in limiting resource usage. However, these approaches support only standard mathematical operators (e.g., MIN, COUNT, and AVG) over homogeneous data types. Emerging sensor networks will need to support localized cooperation of sensor nodes to perform complicated tasks and in-network data processing to transform raw data into high-level domain-dependent information.

Thus, a new virtual sensor model designed to abstract data from physical sensors will be introduced. This abstraction will allow a developer to precisely specify the operations that are to be performed in the network over a set of data from different (possibly heterogeneous) data sources. Virtual sensors will enable adaptive and efficient in-network processing that dynamically responds to an application's needs. Within this area, research will focus on topics ranging from algorithms defining abstractions to algorithms for allowing the virtual sensors to intelligently move through the environment, for example following the user as he moves.

For example, on an intelligent construction site, users may desire the cranes to have safe load indicators that determine if a crane is exceeding its capacity. Such a virtual sensor would take measurements from physical sensors that monitor boom angle, load, telescoping length, two-block conditions, wind speed, etc. [8] Signals from these individual sensors are used in calculations within the virtual sensor to determine if the crane has exceeded its safe working load. With the virtual sensors model, an application interacts with a combination of physical and virtual devices embedded in the environment.

5. Resulting Middleware

By abstracting the process of selecting sensors to interact with, the middleware will enable applications to make intelligent tradeoffs regarding the properties of the selected sensors and their communication links without having to directly deal with low-level programming concerns. Applications will be able to use a high-level, asynchronous query and response mechanism for retrieving data from the environment. As necessitated by the unpredictable sensor environment, these interactions will be data-centric, thereby directly leveraging application knowledge within the communication process. The use of virtual sensors will support reusability in the future multipurpose nets deployed to support numerous applications. The cost of physically visiting each sensor to reprogram it is prohibitive, and therefore the ability to remotely reprogram sensor networks to tailor them to particular applications will be essential.

The above contributions will be combined into an implemented middleware system that will allow evaluation of both the model and the particular implementation of the model and evaluation of the communication approaches used to support the model.

Conclusions

The proposed middleware model will present a unique view of programming pervasive computing environments that in the future will include large numbers of heterogeneous wireless sensors. By creating high-level programming abstractions that encapsulate the locality of pervasive computing interactions, this middleware model will be a first exploration in enabling novice programmers to create sophisticated pervasive computing applications, thus bringing the future of ubiquitous environments one step closer.

Acknowledgments

The author would like to thank her Ph.D. adviser, Dr. Christine Julien, for many valuable discussions, suggestions, and her encouragement. Thanks are also due to the Center for Excellence in Distributed Global Environments for providing research facilities and the collaborative environment. This research is funded, in part, by the National Science Foundation (NSF), Grant #CNS-0620245. The conclusions herein are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

[1] E. Cheong, J. Liebman, J. Liu, and F. Zhao. TinyGALS: A programming model for event-driven embedded systems. In *Proc. of the 2003 ACM Symp. on Applied Computing*, pages 698–704, 2003.

[2] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. Murphy, and G.P. Picco. TinyLIME: Bridging mobile and sensor networks through middleware. In *Proc. of the 3rd Int'l. Conf. on Pervasive Comp. and Comm.*, pages 61–72, 2005.

[3] C.-L. Fok, G.-C. Roman, and C. Lu. Rapid development and flexible deployment of adaptive wireless sensor network applications. In *Proc. of ICDCS*, pages 653–662, 2005.

[4] L. Girod, J. Elson, A. Cerpa, T. Stathopoulous, N. Ramanathan, and D. Estrin. EmStar: A software environment for developing and deploying wireless sensor networks. In *Proc. of the 2004 USENIX Technical Conference*, pages 283–296, 2004.

[5] B. Greenstein, E. Kohler, and D. Estrin. A sensor network application construction kit (SNACK). In *Proc. of SenSys*, pages 69–80, 2004.

[6] J. Kosh and R. Pandey. VM*: Synthesizing scalable runtime environments for sensor networks. In *Proc. of SenSys*, 2005.

[7] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TinyDB: An acquisitional query processing system for sensor networks. *ACM Trans. on Database Systems*, 30(1):122–173, 2005.

[8] L. Neitzel, S. Seixas, and K. Ren. A review of crane safety in the construction industry. *Applied Occupational and Environmental Hygiene*, 16(12):1106–1117, 2001.

[9] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. Campbell, and K. Narstedt. A middleware infrastructure for active spaces. *IEEE Pervasive Comp.*, 1(4):74–83, 2002.

[10] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record*, 31(3):9–18, 2002.

[11] Y. Yu, B. Krishnamachari, and V.K. Pasanna. Issues in designing middleware for wireless sensor networks. *IEEE Network*, 18(1):15–21, 2004.