Application-Aware Monitoring in Large-Scale Sensor Networks

Uğur Çetintemel
Department of Computer Science
Brown University
ugur@cs.brown.edu

Abstract

Large-scale pervasive platforms of inexpensive micro-sensor units will enable a variety of real-time monitoring applications, and become fundamental to high-level decision-making in many civil and military domains.

We first discuss some important data and resource management issues that need to be addressed in order to efficiently extract and deliver useful information from sensor nets in a timely manner. We then briefly describe an ongoing project, called InfoFabric, which aims to address some of these challenges. InfoFabric is developing algorithms and architectures for proactive, application-aware information management in large-scale distributed sensor networks.

1 Introduction

Recent advances in low-power analog and digital electronics and RF systems have led to the emergence of compact, inexpensive, battery-operated sensor units equipped with computational and wireless communication capabilities [5, 8]. Due to their increasingly favorable form and cost factors, it is feasible to link together a large number of such sensors in order to support fault-tolerant, fine-grained monitoring and tracking applications [3, 7].

Cheap and ubiquitous platforms of networked sensors will be the key to deliver, in real-time, large volumes of useful information to support a variety of military applications (e.g., battlefield monitoring, intrusion detection, etc.) and civil applications (e.g., traffic monitoring, disaster recovery, industrial tracking, plant/equipment maintenance, etc.). Unlike present computing infrastructures that access and manipulate data generated primarily by computers or humans, next-generation information systems will deal with extraordinary amounts of live real-world data provided by large-scale, dense sensor networks that link the physical world to the digital world.

In order to fully realize the benefits of this newly-emerging technology, it is crucial to build robust and scalable data management infrastructures that will facilitate efficient querying, extraction, and delivery of useful information from a deployed sensor network in a timely manner. Below we describe some of the important data and resource management challenges that arise in this context.

2 Challenges in Wireless Sensor Nets

On the surface, a wireless sensor network is basically a distributed system of autonomous nodes (possibly) connected in an ad-hoc fashion. Data management solutions (e.g., synchronization, caching, query processing) designed for conventional distributed systems and databases, however, need to be adapted to the unique requirements and characteristics of sensor network environments and target application domains. In the following, we shortly describe some of the main challenges that arise in this context.

2.1 Energy Constraints

Current technology trends indicate that sensor units will continue to become increasingly smaller, cheaper, and computationally more powerful. Even though improvements in computational power is likely to be determined by Moore's Law, battery power is expected to increase only by 3% yearly [6].

It is clear that energy-efficient mechanisms and policies will be the key to optimize the average lifetime of sensor-based networks. Communication is significantly more expensive than performing local computations (e.g., transmitting a bit is more than an order of magnitude as power expensive as executing an instruction) [4]. Energy limitations, thus, dictate performing as much local processing at the source to reduce the volume of necessary communication.

2.2 Proactive, Real-Time Processing

Unlike traditional distributed information systems, which typically react to human inputs (i.e., *reactive* computing), sensor networks continually generate large amounts of data that need to be proactively processed and delivered to the applications in real-time. This non-traditional mode of operation is referred to as *proactive* computing. On-line monitoring and tracking applications, such as intrusion detection and battlefield monitoring, inherently have strict performance requirements, requiring continuous proactive information processing.

2.3 Collaborative Information Processing

Sensor units may need to collaborate in order to achieve the desired degree of precision and reliability while maximizing operational lifetime. Collaboration is

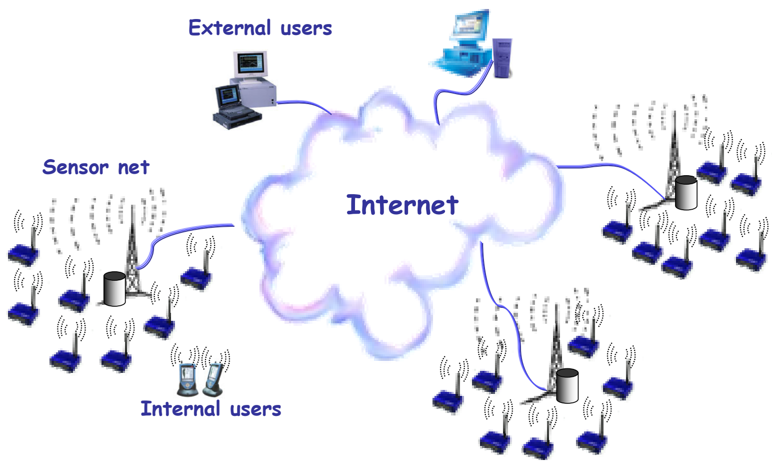

Figure 1: Sensor net structure and usage model

further motivated by the limitations of low-power, short-range wireless communication, which favors localized interactions to incrementally achieve global tasks. For instance, in a decentralized mode of execution, queries injected to arbitrary nodes in the network should be properly distributed, sub-queries be routed to resource-efficient points in the network, and the results be aggregated and relayed back to where they are needed [2]. Such a task clearly requires a high-degree of collaboration among network units.

2.4 Volatile, Highly-Dynamic Environments

Sensor networks are expected to work under hostile, dynamic, and unexpected conditions and scenarios. Ad-hoc deployment implies that there is usually no well-formed communication topology. This situation is exasperated by mobile sensor units (e.g., those attached to the bodies of soldiers). Moreover, communication links and sensor units have high error and fault rates. It is therefore crucial to design flexible mechanisms and policies that *autonomously* adapt to changing environmental (e.g., topology, bandwidth, energy) and application-specific factors (e.g., new ad-hoc tasks specified from multiple network locations).

3 Application-Aware Monitoring

We now briefly describe a novel project, *InfoFabric*, which aims to address some of the challenges outlined above. InfoFabric is developing algorithms and architectures for energy-efficient, adaptive storage and extraction of information in large-scale distributed sensor fields. InfoFabric relies on the performance objectives and semantic requirements of real-time monitoring tasks, target network lifetime, and available resources while making data and resource management decisions.

InfoFabric employs introspection techniques to continually adapt to changes in the application mix (e.g., continuous monitoring vs. ad-hoc tasks), quality of service requirements (i.e., specified in terms of performance, accuracy, reliability objectives), and available resources (i.e., energy, bandwidth, storage, and computational). The issues currently being investigated include application-aware data synchronization, placement, and discovery; decentralized query routing and processing (using localized information propagation and coordination); and capacity planning (e.g., for a set tasks with specific quality of service requirements and desired operational lifetime, what are the resource requirements?).

We now highlight our basic application-aware approach by discussing our basic synchronization model. Figure 1 shows a general sensor network structure and usage model where a large number of sensors are organized into an ad-hoc network and communicate to sensor proxies (i.e., base stations) and possibly among themselves over wireless links. Proxies are typically powered, wired nodes with significant storage and computational capabilities. They are responsible for gathering information from sensors and processing or propagating them, thereby acting as a gateway between sensors and the rest of the world. Internal users may directly interact with the network, whereas external users access the network via the proxy.

In a typical monitoring application, a sensor transmits its readings to its proxy aperiodically whenever there is a change in its sensed data or periodically using preset frequencies. Proxies maintain a view consisting of the sensor readings and execute continuous or ad-hoc queries on these views. The fundamental limitation of this simplistic approach is that synchronization is unaware of the *semantics* and *requirements* of the applications (i.e., queries) executing on view data. It is, however, possible to achieve significant energy savings by using more intelligent synchronization mechanisms that exploit the semantics and requirements of monitoring applications.

Our basic approach is to exploit quality of service information (i.e., a form of application profile [1]) to drive synchronization decisions. In our model, service quality is defined in terms of multiple attributes such as *value*, *staleness*, and *precision*. For each attribute, we define a quality of service graph specifying the aggregate utility of each value in the attribute's domain. A value graph serves the purpose of specifying the importance of a sensor reading. Clearly, different values might have different utility to the system depending on application semantics. For instance, a trigger specified using a simple numerical threshold only cares about sensor values that are above (or below) the trigger threshold. Such a graph allows a synch unit (i.e., a sensor or a cluster sensors) to defer the transmission of or even discard the values that have low utility. Value graphs can also be time dependent, which makes them particularly useful in cases when it is possible to *predict* future readings most of the time (e.g., consider a mobile-object tracking application). A *staleness* value indicates the maximum period of time that the synch unit can defer transmitting a data value after it is generated. This is used to defer, prioritize, or even drop the transmission of individual values. Similarly, *precision* specifies the quality of individual readings and enables a transmission size vs. data quality tradeoff.

Once such utility graphs are obtained for each synch unit (either by the system after an examination of application requirements and semantics, or directly from the application administrator), they are compacted and pushed to corresponding synch units for in-network processing. The graphs are then used at each unit to reduce the amount of data transmission (thereby reducing energy consumption) with minimal negative impact on the overall utility of the system. Clearly, these graphs need to be revised as resource or application parameters change.

We are currently exploring different aspects of this synchronization model. We have been investigating adaptive near-optimal utility specifications, coverage issues (i.e., finding the minimal set of dimensions that can cover the semantics and requirements of the target application domain), choosing proper synch units, and integrating pull and push to optimize the execution of ad-hoc tasks.

4 Conclusions

Large-scale, pervasive sensor network platforms will be the key for supporting a variety of real-time decision making, monitoring, and maintenance applications. Many resource and data management issues remain to be addressed in order to exploit the full potential of these newly-emerging systems and applications. The mechanisms and protocols targeted for these systems need to operate under severe energy constraints (especially battery and bandwidth) and real-time collaborative information processing requirements. The extreme scale, fragility, and unpredictability of the underlying environments introduce further challenges, rendering conventional approaches developed for traditional distributed systems and databases ineffective.

We have started to address some of these issues in the context of *InfoFabric*, which aims to develop a software infrastructure for energy-efficient, *application-aware* extraction and delivery of information from large-scale sensor networks. The problems that we are currently focusing on include adaptive synchronization, data/service placement and discovery. Our decisions are guided by service quality requirements, which are extracted from the workload. These requirements are then pushed inside the network, which continually reconfigures its operation based on this information and available resources.

References

[1] M. Cherniack, M. J. Franklin, and S. Zdonik. Expressing User Profiles for Data Recharging. *IEEE Personal Communications*.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proceedings of the 5th International Conference on Mobile Computing and Networking (MOBICOM)*, 1999.

[3] G.Pottie and W. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*.

[4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proceedings of the 33rd International Conference on System Sciences (HICSS)*, 2000.

[5] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Mobile Networking for Smart Dust. In *International Conference on Mobile Computing and Networking (MOBICOM)*, Seattle, WA, 1999.

[6] J. Rabaey. Silicon Platforms for the Next Generation Wireless Systems – What Role does Reconfigurable Hardware Play? In *Proceedings of the 10th International Conference on Field Programmable Logic and Applications (FPL)*, Villach, Austria, 2000.

[7] P. Saffo. Sensors: The Next Wave of Information. *Communications of the ACM*.

[8] S. Schlosser, J. Griffin, D. Nagle, and G. Ganger. Designing Computer Systems with MEMS-based Storage. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Cambridge, MA, USA, 2000.