Coordination Services for Wireless Sensor Networks

Sameer Sundresh, Gul Agha, Kirill Mechitov, WooYoung Kim, and YoungMin Kwon
Department of Computer Science,
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA.
{sundresh|agha|mechitov|wooyoung|ykwon4}@cs.uiuc.edu
http://osl.cs.uiuc.edu

Wireless sensor-actuator networks can provide the ability to continuously monitor the integrity of structures in real-time, detect damage at an early stage, and provide robustness in the case of catastrophic failures with a fraction of cost associated with today's wired networks. However, sensor-actuator networks require a new paradigm of computing—one which explicitly addresses less capable hardware, unreliable communication with limited bandwidth, and severe energy constraints. We are developing algorithms and software tools which will facilitate monitoring and protection of civil structures using such networks. Specifically, our research addresses problems of communication and coordination in this environment.

A key idea in our approach is to push the computation to the embedded nodes—thus providing greater robustness through decentralization and probabilistic correctness guarantees through statistical reasoning. For example, a high frequency measurement in a dense sensor array results in an overwhelmingly large amount of data. By aggregating data in the network we eliminate redundant information at the sensor nodes in situ, thus reducing transmitted and stored data and conserving power.

Sensor readings need to be correlated in time and space and actuation needs to be synchronized to provide effective monitoring and protection of civil structures. To facilitate such coordination through self-organizing networked sensors, we are developing a range of middleware services such as time synchronization and node localization. Finally, we describe a scalable simulator that we are developing to assist application development on sensor networks. The simulator models three key aspects of a sensor network: the sensor nodes, the data network, and the environment with which the sensors interact. In the future we plan to interface the simulator with actual sensor nodes so that a variety of scenarios can be easily replayed for testing purposes.

1 INTRODUCTION

Sensor-actuator networks can provide the ability to continuously monitor the integrity of structures in real-time, detect damage at an early stage, and provide robustness in case of catastrophic failures. Sensor-actuator networks require a new paradigm of computing—to address less capable hardware, unreliable communication with limited bandwidth and severe energy constraints. We are developing algorithms and software tools that facilitate the monitoring and protection of civil infrastructure using sensor-actuator networks. Specifically, our research addresses problems of communication and coordination in such networks.

A key idea in our approach is to push more processing onto the embedded nodes—thus providing greater robustness through decentralization: correct behavior is guaranteed using statistical reasoning. Another benefit of this approach is that the cost and complexity of wiring large, dense sensor arrays may be reduced through the use of wireless communication if it is used as a primary means of communication. Alternately, wireless communication can be used as a back-up to provide fault-tolerance.

Sensing and actuation need to be correlated in time and space. In order to facilitate such coordination, we develop the following facilities:

- *Time synchronization service:* We use these services for temporal correlation of measurements between custom accelerometer sensors mounted on an experimental model structure.
- *Node localization service:* We employ several techniques—including triangulation using distance estimates based on acoustic ranging, and proximity calculations which are based only on neighbor knowledge—to determine the physical location of sensors. Besides enabling spatial correlation of sensor measurements and actuation control, embedded node locations are used to direct wireless message traffic.
- *Model-based data aggregation service:* Embedded processors may eliminate redundant information directly at the sensor node, thus reducing the amount of transmitted and stored data. We have applied this service to effectively cancel vibrations caused by an impulse disturbance to a beam in simulations.
- *Tracking service:* we have leveraged the location information to develop a tracking service based on binary detection sensors and (piece-wise) linear trajectory prediction.

In addition to the above services, we are implementing a *scalable sensor network simulator* to assist in the development of services and applications. This tool simulates three key aspects of a sensor network: the sensor nodes, the data network and the environment with which the sensors interact. We plan to interface the simulator with real sensor nodes, thus enabling real and simulated scenarios to be replayed for testing purposes.

Our services have been prototyped in simulation and deployed on Crossbow MICA-2 Motes (Crossbow 2003).

2 TIME SYNCHRONIZATION

In structural health monitoring applications, it is essential to correlate the times at which measurements are made by different, physically distributed sensors. For example, given measurements from two different accelerometer devices, we need to know the phase relationships between these data sets. Although time synchronization is required in many types of networks, wireless sensor networks require new solutions due to the constraints of low power, high precision, and a flat, rather than hierarchical, network structure. Elson & Estrin (2001) introduce an approach called *post-facto synchronization,* in which a designated sensor broadcasts a clock pulse message shortly after an event is detected; other sensors use their local clocks to measure the time difference between when they sense the event and when the pulse is received.

Unfortunately, the post-facto synchronization approach is not directly applicable to structural health monitoring applications, where the sample rate measures typically on the order of hundreds of Hz. Typical wireless sensor nodes, however, can only send or receive tens of messages per second.

We have constructed a simple time synchronization service, which calibrates the clocks of adjacent sensor nodes to be within at most 7ms (milliseconds) of each other. With additional refinement, time synchronization better than 1ms can be achieved. The general strategy employed is to exchange clock synchronization messages between sensor nodes. As we have measurements of the delay between the sending and the receipt of the message, we add this as an offset onto the time provided by the sender of the message. Receiving sensors then adjust their clocks accordingly.

In addition to synchronizing adjacent sensor nodes, we build a spanning tree of all the nodes, rooted at the gateway to the sensor network. Nodes throughout the network are synchronized through this tree. This allows us to ensure that even sensors which cannot directly communicate with one another are synchronized. After the synchronization, sensors collect data for a pre-determined duration. In order to save power and improve performance during data collection, the radio is turned off at this time. Upon completion of the data collection phase, the radio is re-enabled, and data is routed to the gateway along the spanning tree. If desired, the system can repeat the *form-tree/synchronize/collect-data/route-data* cycle indefinitely. Since the time synchronization and spanning tree are re-initialized, our service is able to recover from network and sensor node failures.

Figure 1 depicts accelerometer readings made by M. Ruiz-Sandoval and T. Nagayama in B.F. Spencer's group at the University of Illinois at Urbana-Champaign using this time synchronization and routing framework. Note that the synchronization is not perfect between the two sensor nodes. It is close enough for a human to see that they are from the same situation, but it does help to improve the synchronization for the benefit of SHM algorithms.

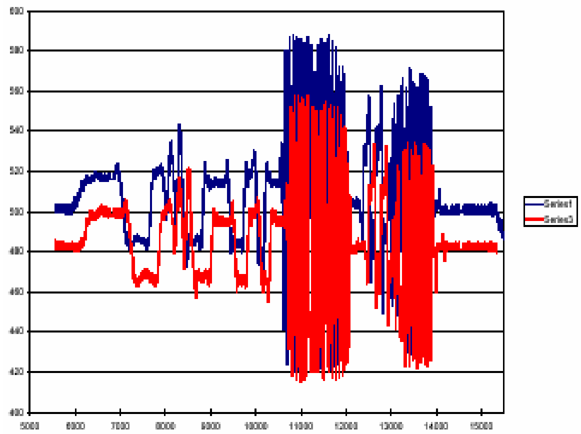

Figure 1. *Accelerometer readings from two imprecisely synchronized sensor nodes (256Hz sample rate).*

In certain cases, we are able to attain better time synchronization, but only for a short period of time. While a node is sending a message on the radio, its neighbors' CPUs receive periodic events for each bit of data transmitted. On MICA-2 nodes, this allows us to create a 2400Hz clock, however nodes only remain synchronized for a small amount of time after the last bit of the message is sent over the radio. This approach is used to measure the travel time of sounds for one of our localization services.

3 SENSOR LOCALIZATION

Sensor network applications such as structural health monitoring require information about the locations of sensors in order to make meaningful use of the collected data. In order to fully realize the benefits of *ad hoc* deployment and provide scalable installation of sensors, it is useful to have the sensor nodes which automatically determine their locations. We have developed two different localization algorithms, and are currently investigating a third approach.

Most localization schemes are based on the concept of measuring distances from a small number of surveyed "anchor" locations and performing triangulation. Perhaps the most straightforward distance measurement approach is to measure the time it takes for a sound (or ultra-sound) wave to propagate from an anchor to a sensor node. We have developed a localization service based on this concept using MICA-2 Motes in conjunction with a custom loud speaker device for anchors. Given sensor nodes placed 10cm above the ground in an outdoor parking lot environment free of obstructions, sensors are able to measure their distance from anchors to within 20cm at a distance of up to 28m. Using this acoustic ranging technique in conjunction with gradient descent triangulation computed on the sensor nodes results in a typical node localization error of 30-40cm.

The primary drawback to direct acoustic distance measurement from anchors is that all sensors must be within range of at least three anchor nodes. In order to mitigate this problem, researchers have developed *multi-hop localization* systems. For example, the Ad Hoc Positioning System (APS: Niculescu & Nath 2001) counts the minimum number of radio hops required for a sensor to communicate with an anchor node as an estimate of the true distance to that anchor. Various techniques may be used to normalize these distance estimates, such as considering the ratio of hops to true distance between different anchor nodes.

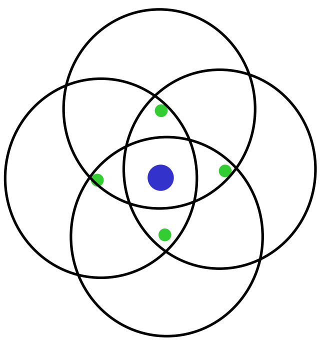

Figure 2. *Center node uses proximity information from neighbors to determine their relative positions.*

The above localization schemes require surveyed anchor locations in order to function. We have developed a *proximity matrix* technique which is able to construct a polar coordinate system and assign locations to nodes without any explicit anchor information (Figure 2); when anchors are available, they can be used to improve results. Each sensor examines which of its neighbors can communicate with each other in order to determine their relative positions. Starting from some arbitrary origin sensor node, this relative position information is used to assign the *θ-coordinate* angle to each sensor node. As in *APS*, the number of radio hops from a sensor to the origin is used as an estimate of the *r-coordinate*.

We are now examining methods to take advantage of known structure to improve localization results. For example, if a number of sensors are laid out along a beam, we expect that the localized positions should form a long, rectangular shape. Thus we can pre-program different sets of sensors with different group ids, and then deploy sensors from the same group along a given beam, and from different groups along adjacent beams. When performing localization, initially sensors localize within their group, taking advantage of the expected shape of the region, and then the groups themselves are localized with respect to each other and anchor nodes.

4 DATA AGGREGATION

Given the broad, flat structure of wireless sensor networks, collecting information from the entire network at a single sink can result in immense network traffic near the sink. This characteristic is at odds with the fact that most current structural health monitoring algorithms are centralized, and even given a distributed algorithm, eventually data needs to be supplied to a human operator. Data aggregation is an approach to help alleviate this pressure; as information flows towards the sink, similar data from different sources is aggregated into a single message, rather than forwarded as separate messages.

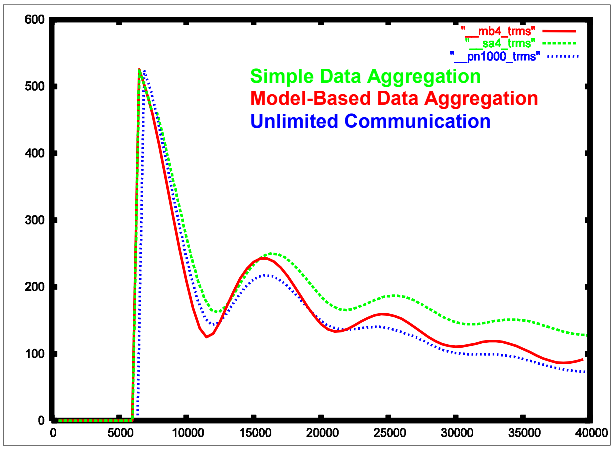

Figure 3. *Beam excited by impulse: RMS Velocity over time.*

Data aggregation has primarily been studied in relation to power savings (Intanagonwiwat *et al.* 2001, Krishnamachari *et al.* 2002) or computing simple functions over the entire network's data set (Madden *et al.* 2002). We have found that aggregation can also be used to improve the *quality* of data available to a real-time application. In our approach to data aggregation, the network is structured as a hierarchy, with the centralized client to the aggregation service at the highest level. In addition to sampling its own data, each sensor node maintains a *predictive model* of some function of the readings from sensors at lower levels. Rather than continually transmitting the newest samples, messages are only sent to update the predictive models at higher levels. These messages are assigned priorities based on the significance of the deviation between the model and reality (Figure 3). Such a priority based communication scheme is explored in Karl *et al.* (2003).

5 TRACKING

Tracking is a common problem in sensor networks when we wish to monitor discrete entities rather than continuous fields. For example, one may want to track the locations of personnel in a failing structure to facilitate evacuation. Leveraging location information, we have developed a tracking service based on *binary-detection proximity sensors* and *linear trajectory prediction* (Mechitov *et al.* 2003). Since our service does not require fine-grained distance measurements between sensors and entities being tracked, nor a specific sensing modality, it can be applied to a wide range of problem scenarios.

Our approach makes use of the computational capabilities of each sensor node to predict the possible locations of the target over time. The measurements and predictions from each of the sensors are combined at a single designated node. For example, given a known maximum detection radius and the time when a sensor first encounters the tracked object, it is safe to assume that the tracked object is at the maximum detection distance from the sensor. If the tracked object moves according to a linear trajectory, the longer the duration that the object spends in a sensor's detection region, the closer it would have had to pass by that sensor. We build on this idea to estimate the distance from a sensor to the object. Thus we calculate its approximate location with a weighted average of the detecting sensors' coordinates, where the inverses of these distance estimates are used as weights. This weighted average can be computed efficiently inside the network during data aggregation. In order to predict the movement of the subject over time, we fit a linear trajectory estimate to the set of most recent location estimates.

6 SIMULATION

Due to time and cost factors, simulation is an attractive alternative to experimental deployment of wireless sensor network applications in development. While several sensor network simulators are currently available, they tend to emphasize one or the other specific aspect while neglecting others. *SensorSim* (Park *et al.* 2000) and *SensorSimII* (Ulmer 2000) are predominantly network simulators; *TOSSIM* (Levis *et al.* 2003) and *TOSSF* (Perrone & Nicol 2002) simulate MICA sensor nodes; *Siesta* (Ledeczi *et al.* 2001) and *EmStar* (Elson *et al.* 2003) focus on applications.

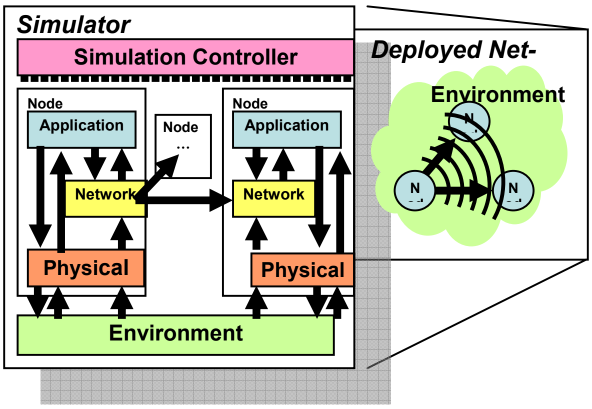

Figure 4. *Simulator block diagram.*

To overcome limitations of the above simulators and aid in prototyping and assessing services such as those discussed above, we are developing a flexible sensor network simulator. Our simulator contains a suite of *interchangeable* and *customizable* components which model applications, network communication, and the physical environment (Figure 4). Simulated sensor nodes run the same source code as is deployed on actual sensor nodes, and include diagnostic facilities such as power utilization analysis. Different network components offer varying degrees of realism, ranging from immediate guaranteed delivery to lossy wireless channels with bit-level collision calculation. Users can assemble application-specific environments from interchangeable tiles representing regions of concrete, grass and walls, which have different signal propagation characteristics. Note that simulation is very fast: e.g., we have simulated an 8192-node acoustic localization scenario at a rate of only 12 real seconds for 1000 simulated seconds on an ordinary PC.

We are now looking into integrating *structural simulation* as an environment component into our sensor network simulator. This feature will allow structural health monitoring applications to be simulated on a realistic model of a large-scale sensor network. Since it is much easier to change the capabilities of simulated sensor nodes than real ones, researchers will be able to determine what capabilities are necessary for distributed structural health monitoring rather than being restricted to the platforms available today.

7 CONCLUSION

We have discussed several coordination services for wireless sensor networks. The general applicability of these services is demonstrated both through their use with applications and through the way they build upon each other. We expect that the availability of this toolkit of wireless sensor network services will aid in the future development of structural health monitoring systems.

ACKNOWLEDGEMENTS

This research has been supported in part by DARPA IXO NEST Program (F33615-01-C-1097) and in part by Motorola Center for Communications (1-5-83423). Work related to time synchronization and routing and integration of sensor network simulation with structural simulation is in collaboration with B.F. Spencer's group at Illinois.

REFERENCES

Crossbow Technology, Inc. 2003. "MICA: Wireless Measurement System." *<http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA.pdf>*

Elson, J., Bien, S., Busek, N., Bychkovskiy V., Cerpa, A., Ganesan, D., Girod, L., Greenstein, B., Schoellhammer, T., Stathopoulos, T., Estrin, D. 2003. "EmStar: An Environment for Developing Wireless Embedded Systems Software." *Center for Embedded Networked Sensing Technical Report 009*, University of California, Los Angeles.

Elson, J. & Estrin, D. "Time Synchronization for Wireless Sensor Networks." In 2001 *International Parallel and Distributed Processing Symposium (IPDPS)*, Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, USA, April 2001.

Intanagonwiwat, C., Estrin, D., Govindan, R., & Heidemann, J. 2001 "Impact of network density on data aggregation in wireless sensor networks." *Technical Report 01-750, University of Southern California*.

Karl, H., Löbbers, M. & Nieberg, T. 2003. "A Data Aggregation Framework for Wireless Sensor Networks." *Technical University Berlin Telecommunication Networks Group Technical Reports Series.*

Krishnamachari, B., Estrin D. & Wicker, S. 2002. "Impact of Data Aggregation in Wireless Sensor Networks." In *Proc. of the IEEE International Workshop on Distributed Event-Based Systems (DEBS),* pp 575–578, Vienna, Austria.

Ledeczi, A., Maroti, M. & Bartok, I. 2001. "Simple NEST Application Simulator (Siesta 02-05-02)." (part of the latest Siesta distribution). *<http://www.isis.vanderbilt.edu/projects/nest/downloads.asp>*

Levis, P., Lee, N., Welsh, M. & Culler, D. 2003. "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications." *In Proc. of the First ACM Conference on Embedded Networked Sensor Systems (SenSys)* (to appear).

Madden, S., Szewczyk, R., Franklin, M., & Culler, D. 2002. "Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks." In *Proc. of 4th IEEE Workshop on Mobile Computing and Systems Applications*.

Mechitov, K., Sundresh, S., Kwon, Y., & Agha, G. 2003. "Cooperative Tracking with Binary-Detection Sensor Networks." Technical Report UIUCDCS-R-2003-2379, Department of Computer Science, University of Illinois at Urbana-Champaign.

Niculescu, D. & Nath, B. 2001. "Ad Hoc Positioning System (APS)." In *Proc. of the 2001 IEEE Global Communications Conference*.

Park, S., Savvides, A. & Srivastava, M.B. 2000. "SensorSim: A Simulation Framework for Sensor Networks." In *Proc. of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2000)*, pp 104–111.

Perrone, L.F. & Nicol, D.M., 2002. "A Scalable Simulator For TinyOS Applications." In *Proc. of the 2002 Winter Simulation Conference*.

Ulmer, C. 2000. "Wireless Sensor Probe Networks - SensorSimII." *<http://users.ece.gatech.edu/~grimace/research/sensorsimii/>*