A SOFTWARE ARCHITECTURE AND SUPPORTING KERNEL FOR LARGELY SYNCHRONOUSLY OPERATING SENSOR NETWORKS

K. H. (Kane) Kim, C.S. Im, M.C. Kim, Y.Q. Li, S.M. Yoo, and L.C. Zheng
DREAM Lab, Dept. of EECS, University of California, Irvine, CA, 92697 U.S.A. http ://dream.eng.uci.edu

Abstract: The *largely synchronous sensor network software architecture* (LaSSeNSA) was formulated with the goal of enabling efficient construction of sensor networks that are easily analyzable and can rarely run into a chaotic situation. There are largely three parts in LaSSeNSA; (1) Global time based coordination of uses of shared communication channels; (2) Global time based coordination of group configuration updates; and (3) Global time based coordination of sensing, communication, and relay activities. The essence of LaSSeNSA as well as various issues related to optimal implementation is presented. For efficient implementation of LaSSeNSA, a sensor node operating system supporting *time-triggered functions* (TTFs) is highly desirable. Our first prototype of TTF support facility was built as an RT subsystem on a small-footprint time-sliced multi-threading kernel. The prototype including both the kernel and the TTF support subsystem is called the *TTF Support OS* (TSOS). Major features of TSOS and a sensor network application development experiment are also presented.

Key words: Sensor, network, real time, time-triggered, TTF, service function, TDMA, TCoDA, wireless, kernel, synchronous, LaSSeNSA, analyzable.

1. INTRODUCTION

Tiny communicating sensors which consume low power and contain sensing, computation, and wireless communication components, are appearing with steadily growing capabilities and varieties [Agr00, Aky02, Hil02, Levo2, She0l]. Such sensors are called *C-sensors* in this paper. New potential applications of networks of such C-sensors are recognized continuously in military and commercial areas such as command-control, surveillance, environment monitoring, targeting system, etc., although their practical deployments may be still several years away.

Due to their low costs, C-sensors may be densely deployed inside or close to the object of interest, dynamically forming a cooperative sensor network. We judge that one of the key technological issues in developing applications of C-sensor networks is to enable harmonious orderly cooperation among the C-sensors to meet real-time (RT) application goals. Cooperation among the C-sensors is needed mainly due to the reliability and functionality characteristics of C-sensors as well as due to the nature of the typical applications. To be more specific, the following causes exist.

(1) The radio communication mechanisms of C-sensors typically have short communication ranges. Multi-hop store-and-forward modes of communications are inevitable.

(2) Applications often require fusion of data from multiple geographically dispersed C-sensors.

(3) C-sensors are often deployed in rough environments and thus temporary or permanent failures of C-sensors are significantly more frequent than the failures of nodes in networks of office computers are. Healthy neighbors of failed C-sensors must thus cooperate to bypass the failed C-sensors or take over their duties and work toward the application goals.

(4) Since C-sensors are often deployed in high density, not all of them need to be activated at any given time to meet the application requirements. This then offers opportunities for cooperative power saving, i.e., permitting some non-essential C-sensors in hibernating modes.

Simplistic designs of sensor nodes to exhibit poorly coordinated highly probabilistic self-centered behavior are bound to result in C-sensor networks which operate in chaotic manners and cannot accomplish RT applications successfully. For example, consider a spy-tracking application involving C-sensors spread out in a field to detect and track any intruding spy in the field. Sensor nodes might use magnetic, acoustic, or other types of sensing mechanisms for the detection and tracking purpose. As spies are tracked, their position reports are sent to a command station equipped with storage and long-distance communication capabilities which are not available in C-sensors. In this spy-tracking C-sensor network, poorly coordinated C-sensor nodes will frequently collide in accessing communication channels. Consequences will then be untimely arrivals of sensor reports at command stations and untimely arrivals of sensor adjustment commands at sensor nodes, which directly lead to application failures. Message forwarding functions will be highly unreliable, too. Poorly coordinated C-sensor nodes are also bound to produce sensor reports which collectively contain little more than what can be provided by a small subset. As multiple spies move around, the sensor-network will be plunged into a chaos.

Coordinated behavior of C-sensor nodes must be achieved by means involving little or modest amount of wireless communications. Coordination mechanisms requiring high-frequency message communications cannot be effective. They may actually add to the forces inducing a chaos. The recognition of this nature led us to explore the principle of *global-time-based coordination of distributed actions* (TCoDA) [Kop97]. Here coordination is realized by use of globally referenced time information, or for short, *global time,* rather than "last-minute" exchange of messages.

We started with applying the TCoDA principle to the wireless communication area, which quickly led us to adopting the TDMA (time-division multiplexed access) approach. Later we saw the needs for well coordinated behavior in other areas as well, e.g., coordination of sensing times of distributed sensors, dynamic formation and reconfiguration of closely interacting sensor-groups, etc. Subsequent efforts for exploring the TCoDA principle in a top-down manner in the context of C-sensor networks, led us to the formation of a new software architecture named the *largely synchronous sensor network software architecture* (LaSSeNSA). The main purpose of this paper is to discuss the essence of LaSSeNSA as well as various issues related to optimal implementation.

Our subsequent studies on implementation approaches for LaSSeNSA produced an idea that if the sensor node operating system (OS) supports *time-triggered functions* (TTFs), not only efforts for implementation of C-sensor network application systems but also their analysis and validation efforts are greatly simplified. The most general form of a TTF can be expressed as,

"*From GlobalTime* = T1 *to* T2, *Do an execution of the TTF body Every* P *time-units (= iteration-interval) By* GCT"

where GCT denotes the *guaranteed completion time* [Kim00]. TTFs make it very simple to implement periodic tasks which are often needed in sensor reading, communication, etc.

After some analysis and preliminary experimental investigations, we judged that in spite of the small memory foot-print requirements inherent in C-sensors, it was well worth equipping C-sensors with OS mechanisms supporting TTFs. In principle, TTFs can be facilitated on both *multi-threading OSs* and *single-threading non-preemptive task* (STNT) *OSs* (which are equipped with device drivers and use a single thread to execute both commanded and interrupt-triggered tasks in first-in-first-out (FIFO) manners). Although we first devised an approach for supporting TTFs on STNT OSs [Kim03], our first prototype of TTF support facility was built as an RT subsystem on a small-footprint time-sliced multi-threading kernel. The prototype including both the kernel and the TTF support subsystem is called the *TTF Support OS* (TSOS). TSOS adopted some device drivers included in TinyOS, an STNT OS developed at UC Berkeley [Hil02], with some modifications.

We have performed a spy-tracking experiment by using the Mica sensor nodes equipped with TSOS and magnetic sensing mechanisms and acoustic sensing mechanisms. The results clearly indicated the promising nature of LaSSeNSA and TSOS. Both TSOS and the experiment performed are also discussed in this paper.

This paper starts in Section 2 with a discussion on the essence of LaSSeNSA. Several issues related to efficient implementation are discussed. The main features of TSOS are discussed in Section 3. The spy-tracking experiment is discussed in Section 4 and the conclusion is provided in Section 5.

2. THE ESSENCE OF LASSENSA

The C-sensors such as Mica sensor platforms have the following common characteristics.

(1) The amount of memory available, including both RAM and flash ROM, is fairly small although its continuous increase can be safely predicted. 32 KB RAM and 512 KB ROM may become available in 2005.

(2) The wireless communication mechanisms are limited in their communication ranges (up to 200 feet) and bandwidth. The number of channels is limited to one or a few. Controlling the communication ranges to stay below the maximum possible range is usually needed to enable reasonable chances for multiple simultaneous inter-node message communications.

(3) The batteries cannot be recharged easily in most application environments.

(4) An RS232 or similar serial communication port is an optional item. This can be used for loading a program onto the flash ROM and message communication between a command station and its slave C-sensor.

Given these characteristics and the possibility of deploying dense networks of C-sensors in application fields, effecting well-coordinated operations of C-sensors is a research issue of critical importance. LaSSeNSA was devised to enable avoiding chaotic situations arising from poorly coordinated behavior of C-sensors. Therefore, it should be relatively easy for developers of sensor network applications based on LaSSeNSA to avoid facing an explosion in the number of special cases to be dealt with.

There are largely three parts in LaSSeNSA and they are discussed in the following subsections.

2.1 Global time based coordination of uses of shared communication channels: TDMA

A well-established cost-effective way for letting distributed nodes share wireless communication channels is to enforce TDMA rules [Kop97]. It is our judgment that TDMA schemes are particularly attractive in C-sensor network environments. TDMA is effective in preventing collisions among C-sensors which can listen to each other. It is well known that in light-traffic conditions, TDMA has the drawback of relatively high overhead. However, this small drawback is more than compensated by the enormous advantages which become evident in typical C-sensor network application environments where wireless communication bandwidths are severely limited precious resources and collision costs are severe.

In the case of the prototype C-sensor developed at UC Berkeley and called the Mica platform, there is only one frequency radio channel of 40Kb/S bandwidth [Hil02]. In applications involving dense networks of this type of C-sensors, the time costs for collision detections and repeated competitions for accessing channels easily become prohibitively high unless sensor nodes are designed to access channels in well coordinated manners.

On the other hand, in large-scale networks, nodes separated beyond certain distance cannot talk and listen to each other. Therefore, grouping the nodes into closely cooperating groups and making groups provide *message-forwarding services* to other groups are essential requirements.

For the sake of simplicity in discussing the basic requirements and the promising approaches, we will consider C-sensors all of which share single fixed-frequency radio communication channel. In LaSSeNSA, a node-group has the following characteristics:

(1) Every node-group is a TDMA group.

(2) Each node-group has at least one member which can communicate with a non-member. Such a member is called a *gateway node.*

(3) Each node-group has one *group-master node* which should have the capability of talking to and listening from every member node.

(4) A member of a group, which is not the group-master, is called a *worker node.* A worker can send a message to another normally via the master of the group. Therefore, worker nodes do not communicate directly among themselves. Naturally, workers in a group *must always listen during the slots belonging to their master.*

An ideal situation is where each node-group has one and only one gateway node. In such a case, the only constraint on multiple simultaneous message broadcasts is:

(C1) No more than one message broadcast which can be heard by a gateway node during the time-slots paid attention by the gateway node should be generated within the sensor network.

Enforcing C1 requires global knowledge on the network status, especially knowledge on which node can listen to which other nodes. Moreover, the function of assigning group membership and TDMA slots to individual members can be a computationally intensive one. Therefore, in LaSSeNSA, that job is held by a relatively powerful node called the *region-command station.* A region-command station is usually a general-purpose server computer with relatively powerful hardware components (e.g., CPU, memory, and disk), abundant energy sources, and strong connections (e.g., wired connections) to the application customers. It collects data from all sensor groups via wireless communication channels and sends useful information to the application customers. It may use a C-sensor as a slave connected through a wire, e.g., serial communication wire, to itself. In a large-scale application, there may be multiple region-command stations which can interact through high-bandwidth high-reliability connections.

Figure 1 illustrates a sensor network instance of LaSSeNSA consisting of 7 active C-sensors and one region-command station. The 7 active C-sensors form 3 C-sensor groups. Each C-sensor group has a group-master node. The link between two C-sensors represents the reachability of radio transmission from one of the two to the other. We initially treated each link as a symmetric two-way link [Kim03] but recently we refined LaSSeNSA to recognize possibilities of asymmetric wireless reachabilities between a pair of C-sensors. For example, Node 4 can hear message broadcasts by the region-command station and Node 5 but the latter nodes cannot hear message broadcasts by the former node.

The TDMA slot assignment for all sensor groups is made by the region-command station. More details regarding the TDMA slot assignment for all

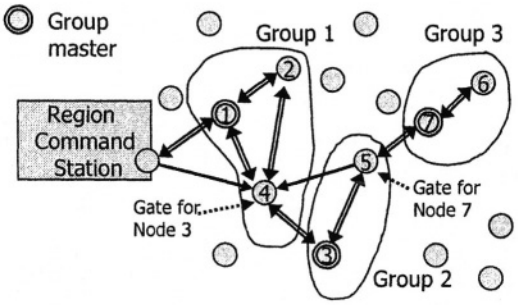

Figure 1. Sensor Groups in a C-sensor network

sensor groups can be found in [Kim03].

The global time base which is an essential ingredient in implementing the TDMA approach can be established in various ways nowadays. If GPS receivers are parts of the sensor nodes, then a global time base of microsecond-level precision can be established easily. Otherwise, a master-slave scheme which involves time announcements by the master as well as exploitation of the knowledge on the message delay between two nodes, can be used to establish a global time base of about 10 millisecond level precision with the C-sensors such as Mica platforms.

2.2 Global time based coordination of group configuration updates: Dynamic TDMA grouping

The scheme in LaSSeNSA for admitting a newly awakened node into an existing cooperating sensor group was formulated with the following challenging goals:

(1) As a newly awakened node joins an existing cooperating group, the members of the group should be able to continue their application tasks *without significant interruptions* due to the joining of the new node.

(2) Upon becoming awakened, a node should be able to join an existing group, receive an assignment of an application-specific task, and start performing its application task *within an acceptable time bound.*

(3) When a member node crashes and thus loses its membership of a group, remaining members of the group should not experience any unacceptable interruptions in their application tasks.

Therefore, the group configuration update is a dynamic process and the scheme adopted is called the *dynamic TDMA grouping scheme.*

We initially formulated the dynamic TDMA grouping scheme with the simplistic assumption about the symmetry in wireless reachability between every pair of C-sensors [Kim03] but recently we refined the scheme to handle the cases of asymmetry in wireless reachability between pairs of C-sensors. As a result, the complexity of the scheme grew considerably.

The asymmetry in wireless reachability between a member node and a newly joining node can be detected in various manners. One such case is depicted in Figure 2. A newly joining node N can hear from the group-master M and three workers, W1, W2, and W3. It thus announces its liveliness. M will not forward the liveliness report of N to the command station since it does not hear the liveliness report. After a while, W1, W2, and W3 detect the liveliness report of N and realize that M has not heard it. The three workers then alert M of N's liveliness report. M forwards it to the command station. Various possible cases have been considered and an admission protocol which accounts for all those cases has been established.

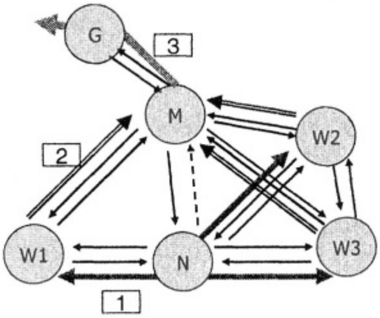

Figure 2. Admission of a node in a case of asymmetry in reachability

Due to the space limit, all those cases are not discussed here.

An additional aspect of the dynamic TDMA grouping scheme is related to the cases where C-sensors are too densely populated in certain parts of the application field. How the sensor nodes will be clustered or spread out geographically is not always predictable at design time. In the case of an excessively dense population, it is not desirable to keep all the C-sensors running all the time. An excessively dense sensor network will unnecessarily burden the communication channels while producing the amount of valuable information which is nearly the same as that which can be obtained with the operation of a subset of the sensors. Therefore, if a node in an area where active worker nodes are already densely populated, announces its liveliness, then it is assigned the role of an *auxiliary worker* under the dynamic TDMA grouping scheme. An auxiliary worker is allowed to occasionally share TDMA slots of a nearby worker to provide its sensor reports. Therefore, workers can cover the application field without the help of auxiliary workers but the overall sensing accuracy can be improved by receiving occasional helps from auxiliary workers.

Optimal assignment of sensor-groups and TDMA slots is the duty of the region-command station and a very complicated subject. So far, we have adopted a heuristic approach. Much further study is needed in this area.

The orderly joining scheme discussed in this section can be extended easily to fit the case where multiple frequency channels are used. The orderly joining enabled is another significant advantage of LaSSeNSA.

2.3 Global time based coordination of sensing, communication, and relay activities

Each member of a sensor group must operate one or more sensing mechanisms during the time-slots in which it does not have to pay attention to communication channels [Kim03].

LaSSeNSA allows fine-precision coordination of the timings of sensing activities in multiple C-sensors. Coordination of the timings of communication and relay activities in different C-sensors is also not difficult. TDMA slot assignment can also be done to reflect such coordination goals.

3. TTF SUPPORT OS

As mentioned in Section 1, we believe that TTF is a sound fundamental building-block for constructing LaSSeNSA systems. We first devised an approach for supporting TTFs on STNT OSs [Kim03]. However, a prototype implementation of such a TTF support facility has not been completed yet. Instead, our first prototype of a *TTF* support facility was built as an RT subsystem on a small-footprint time-sliced multi-threading kernel running on the Mica platform. The prototype including both the kernel and the TTF support subsystem is called the *TTF Support OS* (TSOS). A substantial part of the structuring approaches and resource management techniques adopted in TSOS originated from our previous research on middleware support for high-level RT distributed computing objects [Kim97, Kim99, Kim00, Kim02]. Some major features of TSOS are briefly discussed in this section.

3.1 Threads and dynamic binding

In LaSSeNSA, applications are programmed as a set of TTFs. Each TTF is then assigned a thread within TSOS. This assignment or *TTF-thread binding* is a dynamic binding.

To be more specific, TTFs are dynamically registered to TSOS by applications. The registration parameters of TTFs, including the timing requirement specifications, a pointer to its code body, and the interrupt mask specification, are stored in a list called the TTF List within TSOS. The TTF List is periodically examined by the TTF scheduler within TSOS and when the next execution round of a TTF is found to be approaching within a pre-determined time-window, the TTF is bound to an available thread.

3.2 Two-level scheduling

TSOS enables applications to specify special grouping of TTFs. TTFs without such specification belong to the main group. Typically applications involve one or two groups of TTFs. TSOS then attempts to provide one virtual machine to each TTF group.

TSOS manages multiple *ready queues* and each ready queue corresponds

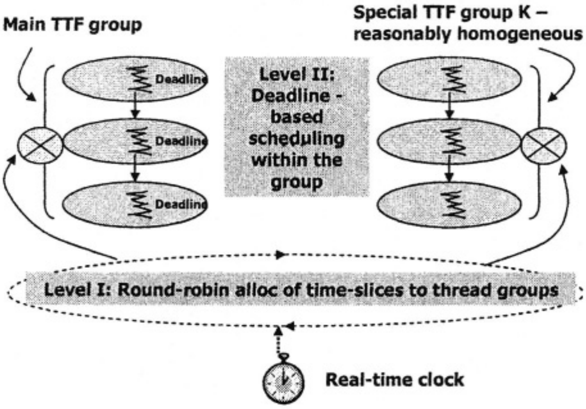
Main TTF group
Special TTF group K – reasonably homogeneous
Deadline
Deadline
Deadline
Level II: Deadline - based scheduling within the group
Level I: Round-robin alloc of time-slices to thread groups
Real-time clock

Figure 3. Two-level scheduling of TTF-threads in TSOS

to one TTF group. The two-level scheduler of TSOS selects one of multiple groups as the running group in the round-robin fashion. Therefore, if there are two TTF groups, time-slices are strictly alternatively given to the TTF groups. Within the running group, the time-slice is given to a TTF-thread following the policy of earliest-deadline-first or its variation. Figure 3 depicts this two-level scheduling structure.

The criteria for forming groups are a subject for much further study. We initially adopted this grouping approach for the cases where separating a group of compute-bound TTFs from the group of I/O-bound TTFs gave certain advantages in terms of performance prediction, search for effective in-group scheduling approaches, etc.

3.3 Critical sections

TSOS does not provide semaphores or mutexes for critical sections because these mechanisms may cause non-deterministic timing behavior of user applications in forms that make the timing analysis of user applications quite complicated. Instead, the protection of critical sections can be achieved via the *thread-to-thread atomic section* (TTAS) [Kim99].

4. AN EXPERIMENTAL SPY-TRACKING SENSOR NETWORK

We have performed a spy-tracking experiment by using the Mica sensor nodes equipped with TSOS. The microcontroller in the Mica C-sensor platform used in our prototype implementation is Atmega 128 which has a 8-bit RISC CPU core running at the frequency of 7.3728MHz, 64K of program memory, 4K of data memory, and several peripherals including SPI, USART, ADC, and digital I/O ports. The sensing mechanisms used in this experiment include a magnetic sensor and an acoustic sensor. Two core requirements of this application are timely sensing of moving objects and dynamically configuring an ad hoc network for delivering sensor data from all C-sensors to designated region-command stations. To fulfill these requirements, three TTFs were designed and implemented on TSOS:

(1) The TTF for periodic sensing of moving objects with the scheduling frequency of 20.83Hz;

(2) The TTF for radio transmission with the scheduling frequency of 1.736Hz; and

(3) The TTF for handling incoming radio messages with the scheduling frequency of 20.83Hz.

All these TTFs have an RT requirement in the sense that a small deviation of execution timing of any TTF may cause a loss of message or message collision during ratio communication and a wrong decision on sensing objects.

We ran this application with 24 C-sensors in a 4x6 grid network with the grid spacing of 4 ft. These 24 C-sensors build up an ad hoc network in an incremental fashion as each one of the C-sensors joins the current network. One of the 24 C-sensors plays a role of a slave / gateway to a command station where human operators can also monitor the operation of the sensor network. This slave is connected to a PC via serial communication.

We achieved the 410 usec jitter in global clock synchronization with the local real-time clocks set to operate at the resolution of 30.5 usec. In addition, our TDMA protocol showed an average 95.6% of successful radio message transmission between any two adjacent motes in the network. This result is an indirect demonstration of the strong RT scheduling ability of TSOS that has not been seen much in the C-sensor network field so far. The smooth spy-tracking performance demonstrated had not been shown much either in the past C-sensor network field.

5. CONCLUSIONS

LaSSeNSA is a C-sensor network architecture which exploits the TCoDA principle extensively. Its main advantages are in making the application systems easily analyzable. Its optimal implementation in various application environments is regarded as a fruitful area for future research.

The current TSOS was developed for platforms not containing GPS receivers. We plan to develop another version of TSOS for platforms equipped with GPS sensors. Further performance evaluations of TSOS are in plan.

ACKNOWLEDGMENT

The research work reported here was supported in part by the NSF under Grant Numbers 02-04050 (NGS) and 03-26606 (ITR), in part by the US DARPA under Contract F33615-01-C-1902 monitored by AFRL, and in part by a gift from Microsoft Corp. No part of this paper represents the views and opinions of any of the sponsors mentioned above.

REFERENCES

[Agr00] Agre, J. and Clare, L., "An integrated architecture for cooperative sensing networks", *IEEE Computer,* Vol. 33, Issue 5, May 2000, pp. 106-108.

[Aky02] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "A survey on sensor networks", *IEEE Communications Magazine,* Vol. 40, Issue 8, Aug. 2002.

[Hil02] Hill, J.L. and Culler, D.E., "Mica: a wireless platform for deeply embedded networks", *IEEE Micro,* Vol. 22, Issue 6, Nov/Dec 2002, pp. 12-24.

[Kim97] Kim, K.H., "Object Structures for Real-Time Systems and Simulators", *IEEE Computer,* Vol. 30, No.8, pp. 62-70, August 1997.

[Kim99] Kim, K.H., Ishida, M., and Liu, J., "An Efficient Middleware Architecture Supporting Time-Triggered Message-Triggered Objects and an NT-based Implementation", *Proc. ISORC '99 (IEEE CS 2nd Int'l Symp. on Object-oriented Real-time distributed Computing),* May 1999, pp.54-63.

[Kim00] Kim, K.H., "APIs for Real-Time Distributed Object Programming", *IEEE Computer,* June 2000, pp.72-80.

[Kim02] Kim, K.H., "Commanding and Reactive Control of Peripherals in the TMO Programming Scheme", *Proc. ISORC '02 (IEEE CS 5th Int'l Symp. on Object-Oriented Real-Time distributed Computing),* Crystal City, VA, April 2002, pp.448-456.

[Kim03] Kim, K.H., and Li, Y.Q., "Toward Easily Analyzable Sensor Networks via Structuring of Time-Triggered Task", *Proc. FTDCS 2003 (9th IEEE Workshop on Future Trends of Distributed Computing Systems),* San Juan, Puerto Rico, May 2003.

[Kop97] Kopetz, H., *'Real-Time Systems: Design Principles for Distributed Embedded Applications',* Kluwer Academic Publishers, ISBN: 0-7923-9894-7, Boston, 1997.

[Lev02] Levis, P., and Culler, D., "Mate: A Tiny Virtual Machine for Sensor Networks", *ASPLOS,* Dec. 2002.

[She01] Shen, C.C., Srisathapornphat, C., and Jaikaeo, C., "Sensor information networking architecture and applications", *IEEE Personal Communications,* Vol. 8, Issue 4, Aug. 2001, pp.52-59.