ITR/SY: A Distributed Programming Infrastructure for Integrating Smart Sensors

NSF Program CCR-0121638

PIs: Umakishore Ramachandran
Kenneth Mackenzie
Steve DeWeerth
Irfan Essa
Thad Starner

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
Phone: (404) 894-5136
FAX: (404) 385-2295
e-mail: rama@cc.gatech.edu
WWW URL: http://www.cc.gatech.edu/~rama

Annual Report via Fastlane May 31, 2007

1 Activities and Findings

1.1 Research and Education

The proposed research is integrating sensing hardware, embedded processing and distributed system support to build a seamless programming infrastructure for ubiquitous presence applications. Fundamental invention and integration of techniques spanning programming idioms and runtime systems for distributed sensors, and building blocks for embedded processing are expected as the primary intellectual contributions of the proposed research. Interfacing these technologies to emerging applications on the one end and novel off-the-shelf sensors at the other end are secondary goals of the proposed research.

In our exploration of novel techniques to support integration of smart sensors into a distributed programming infrastructure, we are conducting focused studies into 1) camera-based sensor networks with applications to airport surveillance and homeland security; 2) efficient resource management for streaming applications in wireless sensor networks, cluster computing, and grid computing environments; and 3) leveraging web service technologies to support stream based distributed computing applications. We are also developing specific driving pervasive computing applications to assist in design and evaluation of our distributed infrastructure components. Further, we are developing an *embedded pervasive lab* as a testbed for integrating sensor technologies into pervasive computing applications. This subsection details the research accomplishments this past year (June 2006-May 2007).

1.1.1 Wireless sensor networks

SensorStack. Current technology trends suggest that futuristic wireless sensor networks (FWSN) are well equipped to support applications such as video surveillance and emergency response that have, in addition to high computation and communication requirements, a need for dynamic data fusion. Because of the dynamic and heterogeneous nature of FWSN environment, adaptability is emerging as a basic need in designing network protocols. Adaptability can be broadly supported at two levels: network level and node level. At the network level, different nodes are the entities that adapt their roles in a cooperative manner to improve network level cost metrics. Similarly, at the node level, different protocol modules are the entities that adapt their behavior in a holistic manner to best perform the roles assigned to the node.

The goal of Rajnish's thesis [8] is to provide an adaptable protocol stack architecture for data fusion applications. Towards the goal, the thesis presents the design of SensorStack, that addresses three key issues. First, towards network-level adaptability, how to dynamically adapt the placement of a fusion application task graph on the network? How to support such an adaptation in an application-specific manner? We have designed a distributed role assignment algorithm and implemented in the context of DFuse, a framework for distributed data fusion. Simulation results show that the role assignment algorithm significantly increases the network lifetime over static placement.

Second, towards node-level adaptability, how to facilitate cross-layering on a node to foster agile adaptation of a node's behavior commensurate with the network-level changes? SensorStack provides information exchange service (IES) as a framework for cross-module information exchange. IES can be thought of as a centrally controlled bulletin-board where different modules can post available data, or request information, and get notification when information becomes available. IES preserves the benefits of layering while facilitating adaptability. IES has been implemented in TinyOS and Linux, to show both the feasibility of the design as well as demonstrate the utility of cross-layering to increase application longevity.

Towards tying the network and node level adaptability together, control data published in IES needs to be shared across the network. SensorStack uses a probabilistic broadcast-based dissemination service (IDS) for control data. Simulation experiments show that IDS allows nodes to share IES data more efficiently than using multiple diffusion trees.

Finally, an efficient and reliable dissemination of information over a large area is a critical ability of a sensor network for various reasons such as software updates and transferring large data objects (e.g., surveillance images). Thus, for bulk data broadcast, we design FBcast, an extension of the probabilistic broadcast based on the principles of modern erasure codes. Simulation results on TOSSIM show that FBcast offers higher reliability with lower number of retransmissions than traditional broadcasts.

Information Exchange Service. Wireless Sensor Networks are deployed in demanding environments, where application requirements as well as network conditions may change dynamically. Thus the protocol stack in each node of the sensor network has to be able to adapt to these changing conditions. Historically, protocol stacks have been designed with strict layering and strong interface between the layers leading to a robust design. However, cross-layer information sharing could help the protocol modules to make informed decisions and adapt to changing environmental conditions. There have been ad hoc approaches to facilitating cross-layer cooperation for adaptability. However, there has been no concerted effort at providing a uniform framework for cross-layer adaptability that preserves the modularity of a conventional protocol stack. In [9], we present a novel service, information exchange service (IES), as a framework for cross-module information exchange. IES is a centrally controlled bulletin-board where different modules can post available data, or request for useful information, and get notified when the information becomes available. IES is integrated into the proposed *SensorStack* architecture that preserves the benefits of layering while facilitating adaptability. IES has been implemented in TinyOS and Linux, to show both the feasibility of the design as well as demonstrate the utility of cross-layering to increase application longevity.

Fountain Broadcast. Efficient and reliable dissemination of information over a large area is a critical ability of a sensor network for various reasons such as software updates and transferring large data objects (e.g., surveillance images). Thus efficiency of wireless broadcast is an important aspect of sensor network deployment. In [10], we study FBcast, a new broadcast protocol based on the principles of modern erasure codes. We show that our approach provides high reliability, often considered critical for disseminating codes. In addition FBcast offers limited data confidentiality. For a large network, where every node may not be reachable by the source, we extend FBcast with the idea of repeaters to improve reliable coverage. Simulation results on TOSSIM show that FBcast offers higher reliability with lower number of retransmissions than traditional broadcasts.

1.1.2 Service Composition and Scheduling

Streaming applications are good candidates to use service oriented architecture due to their high computation and communication needs. Such an application is often represented as a coarse-grain dataflow graph, where each node corresponds to a transformation service that may be applied to the data as it continuously streams through. For example, a video-based surveillance application may analyze multiple camera feeds from a region to extract higher level information such as motion, presence or absence of a human face, or presence or absence of any kind of suspicious activity. Each of the compute intensive part of the application is provided as a service and the services need to executed in a particular order as specified by the dataflow graph. Multiple instances of each service may provide the same functionalities but differ in the implementation and QoS parameters such as latency and throughput. Moreover, the QoS parameters of each service may change due to dynamism in the application and resource availability.

The service composition problem addresses the issue of composing independent services, as specified by the streaming application dataflow graph, in order to deliver the required QoS to the application. Service composition for streaming applications has received attention in the literature only recently. Streamline[1, 2] addresses the closely related problem of scheduling each node of the application dataflow graph in a distributed environment. In contrast, our work provides service composition framework in a grid environment and presents distributed algorithm for composition.

We formulate the service composition problem for streaming applications as a shortest path problem in a layered graph. Each layer of the graph represent multiple instances of each service with different QoS. The weight associated with each link represent the bandwidth availability between the services. We use Dijkstra's shortest path algorithm as the centralized service composition algorithm. In addition, we design a distributed composition algorithm that is run locally by each service node in order to select the next level service in the composition graph.

For comparison, we adapt Streamline, which was originally designed for scheduling streaming application, for service composition. We only consider the eligible services for each node of the dataflow graph, and pick the best service based on Streamline heuristic. Initial experimental results on a 4 stage dataflow graph reveal that the distributed composition algorithm performs close to centralized Dijkstra and streamline algorithm on throughput metric. However, the distributed algorithm requires minimum changes to the dataflow graph when QoS changes and therefore, has the least overhead. We are conducting more experiment comparing the algorithms for large dataflow graphs to validate the benefits of our approach.

1.1.3 Distributed Garbage Collection Algorithms for Timestamped data.

Memory Optimizations. In [7], we explore the potential of using application data dependency information to reduce the average memory consumption in distributed streaming applications. By analyzing data dependencies during the application runtime, we can infer which data items are not going to influence the application's output. This information is then incorporated into the garbage collector, extending the garbage identification problem to include not only data items that are not reachable, but also those data items that are not fully processed and dropped. We present three garbage collection algorithms. Each of the algorithms uses different data dependency information. We implement the algorithms and compare their performance for a color tracker application.

Streaming applications are often distributed, manage large quantities of data and, as a result, have large memory requirements. Therefore, efficient *garbage collection (GC)* is crucial for their performance. On the other hand, not all data items *affect* the application output due to differences in the processing rates of various application threads. In [6], we propose extending the definition of the garbage identification problem for streaming applications and include not only data items that are not "reachable" but also data items that have no effect on the final outcome of the application. We present four optimizations to an existing garbage collection algorithm in *Stampede*, a parallel programming system to support interactive multimedia applications. We ask the question how far off these algorithms are from an *ideal garbage collector*, one in which the memory usage is exactly equal to that which is required for buffering only the relevant data items. This oracle, while unimplementable, serves as an empirical lower-bound for memory usage. We then propose optimizations that will help us get closer to this lower-bound. Using an elaborate measurement and post-mortem analysis infrastructure in Stampede, we simulate the performance potential for these optimizations and implement the most promising ones. A color-based people tracking application is used for the performance evaluation. Our results show that these optimizations reduce the memory usage by up to 60%.

Distributed Streaming Applications. Distributed stream-based applications manage large quantities of data and exhibit unique production and consumption patterns that set them apart from general-purpose applications. Nissim Harel's dissertation [5] examines possible ways to harness the unique characteristics of such applications to assist in creating efficient memory management schemes. Two complementary approaches are suggested:

- Garbage Identification
- Adaptive Resource Utilization

Garbage Identification is concerned with an analysis of dynamic data dependencies to infer those items that the application is no longer going to access. Several garbage identification algorithms are examined. Each one of the algorithms uses a set of application properties (possibly distinct from one another) to reduce the memory consumption of the application. The performance of these garbage identification algorithms is compared to the performance of an ideal garbage collector, using a novel logging/post-mortem analyzer. The results indicate that the algorithms that achieve a low memory footprint (close to that of an ideal garbage collector), perform their garbage identification decisions locally; however, they base these decisions on best-effort global information obtained from other components of the distributed application. The Adaptive Resource Utilization (ARU) algorithm analyzes the dynamic relationships between the production and consumption of data items. It uses this information to infer the capacity of the system to process data items and adjusts data generation by the application accordingly. The ARU algorithm makes local capacity decisions based on global information. This algorithm is found to be as effective as the most successful garbage identification algorithm in reducing the memory footprint of stream-based applications, thus confirming the previous observation that using global information to perform local decisions is fundamental in reducing memory consumption for stream-based applications.

1.1.4 Packet Delivery in Sensor Networks.

The PI, working with Professor Mostafa Ammar and a student has been exploring issues with respect to the reliability of packet delivery in dense wireless sensor networks [13].

Wireless sensor networks (WSN) built using current Berkeley Mica motes exhibit low reliability for packet delivery. There is anecdotal evidence of poor packet delivery rates from several field trials of WSN deployment. All-to-one communication pattern is a dominant one in many such deployments. As we scale up the size of the network and the traffic density in this communication pattern, improving the reliability of packet delivery performance becomes very important. This study is aimed at two things. Firstly, it aims to understand the factors limiting reliable packet delivery for all-to-one communication pattern in dense wireless sensor networks. Secondly, it aims to suggest enhancements to well-known protocols that may help boost the performance to acceptable levels. We pick three protocols, namely, Flooding, AODV, and Geographic routing as candidates for this study. We first postulate the potential reasons hampering packet delivery rates with current CSMA-based MAC layer used by the radios deployed in WSN. We then propose a set of enhancements that are aimed to mitigate the ill-effects of these factors. Using TOSSIM, we perform a detailed study of these protocols and the proposed enhancements. This study serves several purposes. First, it helps us to quantify the detrimental effects of these factors. Second, it helps us to quantify the extent to which our proposed enhancements improves packet delivery performance. Concretely, we show that using Geographic routing in a WSN with 225 nodes spread over 150 feet x 150 feet, the proposed enhancements yield a 23-fold improvement in packet delivery performance over the baseline. Further, the enhancements result in fairness (measured by the number of messages received from each node at the destination). Lastly, we show that the overhead (in terms of retransmissions, acknowledgement messages, and control messages) is reasonable. A paper based on this work is to appear in ICCCN 2007 [15].

1.1.5 Sensor Technologies: Camera Sensor network testbed

Situation awareness is an important application category in cyber-physical systems, and *distributed video-based surveillance* is a good canonical example of this application class. Such applications are interactive, dynamic, stream-based, computationally demanding, and needing real-time or near real-time guarantees. A *sense-process-actuate* control loop characterizes the behavior of this application class. ASAP [14] is a scalable distributed architecture for a multi-modal sensor network that caters to the needs of this application class. Features of this architecture include (a) generation of *prioritization* cues that allow the infrastructure to pay *selective attention* to data streams of interest; (b) *fidelity adjustments* to effectively use the available distributed resources; (c) *virtual sensor* abstraction that allows easy integration of multi-modal sensing capabilities; and (d) dynamic redirection of sensor sources to distributed resources to deal with sudden burstiness in the application. In addition to providing effective resource utilization, ASAP provides real-time guarantees in the presence of data deluge. In both empirical and emulated experiments, ASAP shows that it scales up to a thousand of sensor nodes (comprised of high bandwidth cameras and low bandwidth RFID readers), significantly mitigates infrastructure and cognitive overload, and reduces *false negatives* and *false positives* due to its ability to integrate multi-modal sensing.

We have built an ASAP testbed with network cameras and RFID readers for object tracking based on RFID tags and motion detection. In implementing ASAP, we had three important goals: (1) platform neutrality for the "box" that hosts each components of ASAP system, (2) ability to support a variety of sensors seamlessly (for e.g., network cameras as well as USB cameras), and (3) extensibility to support a wide range of handheld devices including iPAQs and cellphones. Consequent to these implementation goals, we chose Java as the programming language for realizing the ASAP architecture. The building blocks of our testbed include (1) a network camera, Axis 207MW from Axis Communication, (2) RFID antenna from Alien Technology, and (3) iPAQ. Currently, we have 8 network cameras, 8 USB cameras, and 2 RFID readers with 8 RFID antennas in our testbed. An iPAQ client can send queries indicating which object needs to be tracked or which violation the system needs to detect. For example, a client query can be to find a specific person with a given RFID tag or to find anomalies with a given policy. For the time being, RFID tag and motion detection are used as cues, and sound (phonetic analysis), object flow detection, and vision processing will be incorporated with the next version.

1.1.6 Sensor Technologies: RFID Testbed

RF^2ID (Reliable Framework for Radio Frequency IDentification) is a middleware specifically designed to address the challenges faced by RFID readers. RFID readers are inherently error prone in nature exhibiting large number of false negative readings (not being able to read items within its reading range) and false positive readings (reading items beyond its physical range). The main focus of the system is to provide reliability at middleware level at the same time providing distributed infrastructure for load management. The system uses two major abstractions: virtual readers and virtual paths among the system components for improved system performance. Virtual readers are computation elements that are in charge of RFID readers in a particular geographic location and virtual paths are the communication channels among these virtual readers. The system improves its performance locally among virtual readers by using redundant RFID readers in the system at the same time the overall performance is achieved using the communication mechanism among virtual paths. The path based communication mechanism is also used to develop efficient load management strategies that are appropriate for these RFID devices.

We have developed an experimental testbed using RFID readers and tags that mimic a supply chain environment to evaluate our system performance. It uses railroad tracks and cars carrying RFID tags in an environment where the readers are placed at different static points. It shows the system behavior when the items are in motion. Due to the limited number of RFID readers, we have used emulated readers for scalability study of the system. Our system shows improved system reliability and dynamic load management mechanisms in the real testbed as well as the emulation based study

We have two publications in this space:

1. The first paper [3] shows the improvement reliability improvement of the system when our middleware infrastructure is used. The main goal of the work is to improve the system reliability at middleware level. We have used component level redundancy (larger number of RFID readers) and efficient

communication mechanism among components to improve the system reliability. It uses a small experimental setup of an RFID reader with two antennas and at most 6 tags and emulation based experiments to validate the scalability of our system.

2. The second work [4] focuses on the evaluation of the middleware in a supply chain environment when large numbers of items are moving across the system and the RFID readers are statically placed at different points. Two load management system that matches the behavior of RFID devices are studied and implemented in the system named space based load shedding and time based load shedding strategy. The load shedding mechanism is used where overloaded system components can shed some amount of its computational load in a controlled way so that the overall system performance is minimally affected. We have used an experimental testbed using a reader with four antennas and a toy train system with cars carry the RFID tags along the railroad tracks. To validate the system scalability an emulation based study has also been done.

1.1.7 Textbook Development

In previous reports, we have discussed a course the PI (Ramachandran) has developed and teaches, CS 2200: An introduction to systems and networks. This is a novel approach to teaching a first course in systems in an integrated fashion combining the hardware and software issues in one introductory course. Due to its novelty, there is no textbook that serves the purpose for this course. In Spring 2005, the PI (Ramachandran) wrote a comprehensive set of notes for this course. In Spring 2006, he has signed a contract with Addison-Wesley to have the book published as a textbook [11]. The book review is underway and it is expected that it will be in print by Spring 2008. A paper describing the pedagogical approach underlying this textbook is to appear in WCAE 2007 [12].

1.1.8 Curriculum Development and Korea Initiative

Working with colleagues in the College of Computing, Professor Ramachandran has established a dual degree MS program in embedded software with *Korea University* with funding from the South Korean Government. The students enrolled in the program are employees from leading industries such as Samsung. We have started offering this program from Spring 2007. Professor Ramachandran has been appointed as the Director of Korean Programs to oversee the development and delivery of this dual MS program.

1.2 Training and Development

Needless to say, graduate students pursuing their doctorate are the primary focus of training and development. Recent graduates from our group partially funded by this ITR grant include Nissim Harel (independent entrepreneur doing a start-up in Israel) and Rajnish Kumar (Research Scientist II, Georgia Tech). Two students (Namgeun Jeong and Bikash Agarwalla) are in the final stages of their dissertation research and are expected to graduate in August/September timeframe.

Graduate students associated with the project have been benefiting from interships in leading research labs. This summer David Hilley is at IBM Poughkeepsie working on GPFS file system; Dushmanta Mohapatra is at IBM Almaden working on gray box technologies for storage servers.

We continue to attract bright and interested graduates and undergraduates to research projects in our group. Undergraduate participation in research within the College is facilitated by the excellent UROC program (`www.cc.gatech.edu/program/uroc`), coordinated by Professor Amy Bruckman. A variety of institute-wide programs are also available (`www.undergraduateresearch.gatech.edu`) including a special fund sponsored by the president of Georgia Tech (PURA) and several NSF-sponsored projects. We were pleased to support several undergraduates on our ITR-related projects during Summer/Fall 2006 and Spring/Summer 2007. They were: Robert Steven French (RFID), Steven Dalton (Wireless sensor networks), Sam Young (MobiGO - stateless computing).

1.3 Outreach

Professor Ramachandran has concluded negotiations with Samsung Electronics Corporation to establish a joint-center at Georgia Tech for carrying out collaborative research of mutual interest to Samsung and Georgia Tech. Under this agreement, Samsung will fund a number of research projects that had their roots in the ITR award. One such project is *MobiGo* that supports seamless mobility in a ubiquitous setting. A second one (to be carried out by PI Professor Irfan Essa) involves Non Photo-realistic Rendering.

The joint-center will have engineers from Samsung visiting on a year-round basis and working closely with the faculty and students in the College of Computing.

We are in communication with the City of Baton Rouge and their department of Homeland Security. They are interested in serving as a beta-site for the camera sensor network technology we have developed for video-based surveillance.

We have similar ongoing dialogue with the IT department of the Hartsfield-Jackson International Airport in Atlanta to deploy our camera sensor network.

1.3.1 Research Commercialization

Georgia Tech has active interest in research commercialization. *Venture lab* is an entity that identifies research work in Georgia Tech that are worthy of consideration for commercialization. The *SensorStack* dissertation work performed by one of the students (Rajnish Kumar) funded by this ITR award has been chosen for research commercialization. We have developed an architecture for *priority aware situation awareness* with the SensorStack at its core. Potential application of this technology includes airport surveillance and critical infrastructure protection. Rajnish Kumar, research scientist, has been hired expressly to oversee the development of these technologies for application in such scenarios.

2 Publications and Products

2.1 Publications

See the references at the end of this document for publications that appeared in the period that covers this annual report.

,

2.2 Web Site

Please visit the project web site at www.cc.gatech.edu/~rama/ubiq-presence/

3 Contributions

The activities we are currently undertaking have resulted in a significant number of publications and software artifacts. These are listed in the references at the end of this report.

3.1 Human Resources

Roughly 10 graduate students, 6 undergraduate students, and 1 research scientist have been associated with this project during the period covered by this annual report.

3.2 Student Placement

We continue to place student members of our research group in interesting project-related internships, graduate programs and industry jobs.

Undergraduate alumni of this project include:

- Seth Horrigan has been accepted to several graduate schools including UC-Berkeley and UIUC.
- Robert Steven French has accepted a position with Microsoft Research. He also received the *Outstanding Undergraduate Research Assistant* award in March 2007.
- Vladimir Urazov has started graduate work at Georgia Tech from Spring 2007.
- Sam Young has accepted a position with Amazon corporation.

Several students trained by this projects are sought after for pursuing internships in industries:

- David Hilley won an IBM Fellowship in 2007 and is doing an internship at IBM Poughkeepsie for Summer 2007.
- Dushmanta Mohapatra is doing an internship at IBM Almaden for Summer 2007.
- Professor Ramachandran has an ongoing relationship with the Federal Reserve Bank of Atlanta under which two graduate students are employed as interns year-round.
  Currently, Vladimir Urazov and Steven Dalton are working as interns at FRB.
  Some of the technologies developed through partial support from the ITR grant such as the Stam-

pede system is being used as a dynamic cluster scheduling framework for running compute intensive applications developed by economists.

Graduate alumni of this project include:

- Dr. Rajnish Kumar [8] graduated in August 2006, and has taken up the position of Research Scientist II, College of Computing, Georgia Tech.
- Dr. Nissim Harel [5] graduated in December 2006; he is currently involved with doing a start-up company in Israel.

3.3 Research and Education

The research artifacts from the project are finding their way into graduate courses and we have significant undergraduate participation in project-related research. We have funded a number of undergraduates through the REU supplement attached to this ITR grant, sponsored a number of independent undergraduate research projects for course credit (CS 4903), and have sponsored capstone senior design projects (CS 3901) that each result in a poster presentation at the annual Undergraduate Research Symposium.

The ITR project has reinforced the connectedness of hardware and software and the need to train students in system architecture quite early in their undergraduate preparation. With this in mind, Professor Ramachandran embarked on writing a set of course notes [11] for use in the sophomore level course on systems and networks. The course notes were well received by the students. In Spring 2006, he has signed a contract with Addison-Wesley to have the book published as a textbook. It is expected that it will be in print by Spring 2008. A paper describing the pedagogical approach underlying this textbook is to appear in WCAE 2007 [12].

4 Special Requirements

We have received permission from the Program Director (Dr. Helen Gill) for expending the remaining funds in the project by August 2008.

References

[1] Bikash Agarwalla, Nova Ahmed, David Hilley, and Umakishore Ramachandran. Streamline: A Scheduling Heuristic for Streaming Application on the Grid. In *Thirteenth Annual Multimedia Computing and Networking Conference (MMCN'06)*, San Jose, CA, Jan 2006.

[2] Bikash Agarwalla, Nova Ahmed, David Hilley, and Umakishore Ramachandran. Streamline: scheduling streaming applications in a wide area environment. *Multimedia Systems*, May 2007. http://dx.doi.org/10.1007/s00530-007-0082-0.

[3] Nova Ahmed, Rajnish Kumar, Robert Steven French, and Umakishore Ramachandran. Rf2id: A reliable middleware framework for rfid deployment. In *Proceedings of the 21st International Parallel and Distributed Processing Symposium*, 2007.

[4] Nova Ahmed, Vladimir Urazov, and Umakishore Ramachandran. Evaluating rf2id in a supply chain environment. Submitted for publication.

[5] Nissim Harel. *Memory Optimizations for Distributed Stream-based Applications*. PhD dissertation, Georgia Institute of Technology, College of Computing, December 2006.

[6] Nissim Harel, Hasnain A. Mandviwala, Kath Knobe, and Umakishore Ramachandran. Methods of memory optimizations in streaming applications. In *The 2007 International Conference on Parallel Processing*, September 2007.

[7] Nissim Harel, Hasnain A. Mandviwala, Umakishore Ramachandran, and Kath Knobe. Use of dependency information for memory optimizations in distribute streaming applications. In *16th IEEE International Conference on Computer Communications and Networks (ICCCN)*, August 2007.

[8] Rajnish Kumar. *Adaptable Protocol Stack Architecture for Future Sensor Networks*. PhD thesis, College of Computing, Georgia Institute of Technology, August 2007.

[9] Rajnish Kumar, Santashil PalChaudhuri, Charles Reiss, and Umakishore Ramachandran. System support for cross-layering in sensor network stack. In *The 2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2006)*, 2006.

[10] Rajnish Kumar, Arnab Paul, Umakishore Ramachandran, and David Kotz. On improving wireless broadcast reliability of sensor networks using erasure codes. In *The 2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2006)*, 2006.

[11] Umakishore Ramachandran and Bill Leahy. Course notes for CS 2200 - introduction to systems and networks, 2005. Draft of textbook for CS 2200, URL: http://swiki.cc.gatech.edu:8080/cs2200/1.

[12] Umakishore Ramachandran and Jr. William D. Leahy. An integrated approach to teaching computer systems architecture. In *Workshop on Computer Architecture Education*, June 2007. held in conjunction with 34th International Symposium on Computer Architecture.

[13] Junsuk Shin, Mostafa Ammar, and Umakishore Ramachandran. On improving the reliability of packet delivery in dense wireless sensor networks, 2006. Unpublished manuscript, submitted for publication to a conference.

[14] Junsuk Shin, Rajnish Kumar, Dushmanta Mohapatra, Umakishore Ramachandran, and Mostafa Ammar. ASAP: A camera sensor network for situation awareness. Submitted for publication.

[15] Junsuk Shin, Umakishore Ramachandran, and Mostafa Ammar. On improving the reliability of packet delivery in dense wireless sensor networks. In *16th IEEE International Conference on Computer Communications and Networks (ICCCN)*, August 2007.