
On Composability of Localization Protocols for Wireless Sensor Networks

Radu Stoleru, Texas A&M University

John A. Stankovic and Sang H. Son, University of Virginia

Abstract

Realistic, complex outdoor environments pose significant challenges for node localization in wireless sensor networks. In spite of the fact that many elegant and clever solutions have been proposed, no robust localization system has emerged. This status quo is because existing solutions work well for single sets of assumptions that, however, do not always hold in complex environments. In this article we review the state of the art for node localization in wireless sensor networks and show how localization protocol composability has the potential to provide the robust solution that is needed. By composing localization protocols in a hierarchy and allowing the execution of multiple localization schemes, robust solutions against any single protocol failure can be built.

Wireless sensor network (WSN) systems have recently been developed for several domains: military surveillance, environmental monitoring, habitat monitoring, and structural monitoring. These application domains emphasize the requirements for WSN systems: they are expected to work in very diverse environments, and need to be reliable and operate untethered. When considering the extremely scarce resources available to each sensor node (processing, communication, storage), the aforementioned requirements pose significant challenges. One such challenge is how to accurately find the location of each sensor node at low cost.

The node localization problem has received tremendous attention from the research community, thus emphasizing that it is an important and difficult problem. Despite the attention the localization problem in WSNs has received, no universally acceptable solution has been adopted for realistic outdoor environments. There are several reasons for this. One reason is that in order to obtain higher location accuracy, localization protocols either make simplifying assumptions (e.g., line of sight with sensor nodes, high density of anchor nodes, deployment knowledge) or require sophisticated hardware. In large-scale realistic outdoor deployments, these assumptions do not always hold, and equipping all sensor nodes with expensive hardware is not feasible. Another reason is that localization protocols that do not have strong simplifying assumptions are generally inaccurate. The research challenge we face is how to obtain highly accurate node locations in large-scale sensor networks deployed in complex environments at the lowest cost possible.

Instead of aiming for the perfect, universally applicable node localization scheme, a solution that composes multiple localization schemes was proposed [1]. The idea is to create a framework that allows the execution of several existing localization schemes. A system designer decides before deployment what localization protocols to include in the system, the order in which they execute, and the conditions that trigger subsequent localization scheme executions. This set of local-

ization schemes is organized in a hierarchical structure. The hierarchy and the localization schemes that are members of the hierarchy are stored on each sensor node. A runtime system is responsible for the coordination among neighboring nodes of which localization scheme to execute and its execution.

For the design of the proposed localization framework (localization manager and localization schemes), several research questions need to be considered: what is the taxonomy that will allow us to compare the entire body of localization schemes proposed so far in order to be able to choose the most appropriate set (for a particular context)? Can we identify a set of equivalence classes in which we can partition the entire suite of localization schemes? Are there combinations of localization schemes that are incompatible, and hence cannot be executed simultaneously or sequentially? How does the framework impact the implementation of a localization scheme? Should the system execute all localization schemes, for all nodes, and then combine the results? Should the system execute localization schemes only on nodes that have not been localized? We address these questions in the remaining part of this article.

State of the Art

Several localization systems and algorithms have been proposed in the past [2–4]. In this section we describe a taxonomy that partitions the existing body of localization schemes into two equivalence classes: range/event-based and range-free localization schemes. These classes provide guiding principles for choosing existing localization schemes to be executed in a WSN system. Range-based schemes estimate the distances between nodes (ranges) in order to obtain a node's location. Event-based schemes generate well controlled events in the network (e.g., light spots or sound waves). Range-free techniques do not estimate absolute distances; they use radio connectivity to infer proximity, relative proximity, and location. The two localization categories have different trade-offs when

considering the hardware requirements, localization accuracy, communication overhead, and infrastructure.

Range/Event-Based Localization Schemes

The Global Positioning System (GPS) is well known today, widely used in both military and civil applications. Ranges to several satellites (by measuring the difference in the time of arrival [TDoA] of signals from different satellites) are used in a multilateration procedure to infer the position of the receiver. This localization scheme has high accuracy, but assumes that a line of sight exists between the devices to be localized and satellites. The scheme also requires hardware that both is expensive and consumes significant power.

The Cricket localization system [5] uses ultrasound ranging and predeployed anchor nodes to identify the location of a mobile node. With a large number of anchor nodes, Cricket can identify the location of a node with inch accuracy. This is achieved through a process called trilateration, in which a node with range estimates to three anchor nodes can estimate its own location by solving a system of two equations that are linear in the Cartesian coordinates of the sensor node. A similar linear system can be created when there are more than three anchor nodes in a process called multilateration. The difficulties of this approach are twofold. First, under constraints of form factor and power supply, the effective ranges of such devices are very limited. For example, the effective range of the ultrasonic transducers used in the Cricket system is less than 2 m when the sender and receiver are not facing each other. Second, since most sensor nodes are static (i.e., the location is not expected to change), it is not cost effective to equip these sensors with special circuitry just for a one-time localization.

Another recent localization technique, RIPS [6], is based on radio interferometry, which utilizes two transmitters to create an interfering signal. The frequencies of the emitters are very close to each other; thus, the interfering signal will have a low frequency envelope that can be measured easily by a receiver node. The ranging technique performs very well. The long time required for localization and multipath environments, however, pose significant challenges.

To address the high costs required by range-based localization schemes, event-based localization schemes have been proposed. In the Spotlight localization system [4] the sensor nodes do not need any additional hardware other than what they currently have. All the sophisticated hardware and computation reside on a single Spotlight device. The Spotlight device uses a steerable laser light source, illuminating the sensor nodes placed within a known terrain. The illumination (i.e., an event) is controlled, in that its position and timing are known to the Spotlight device. From the spatiotemporal characteristics of the events known to the Spotlight device, and the temporal properties of the events detected by the sensor nodes, the Spotlight device can infer locations of sensor nodes. The Spotlight localization system assumes that there is a line of sight between the Spotlight device and all sensor nodes in the networks. This assumption does not always hold in deployments of sensor nodes in complex rugged environments. The localization system, however, is much more accurate (i.e., tens of inches) than range-based localization schemes and has a much longer effective range (i.e., thousands of feet) than solutions based on ultrasound/acoustic ranging.

Range-free Localization Schemes

Range-free localization is based on the observation that nodes with radio connectivity are typically in close proximity, while those without radio connectivity are not. Because it only relies

on the use of a radio, this approach can be applied to any wireless device. In this section we describe algorithms that use the radio to estimate node locations in two scenarios: when many nodes in the network have known locations and when very few have known locations.

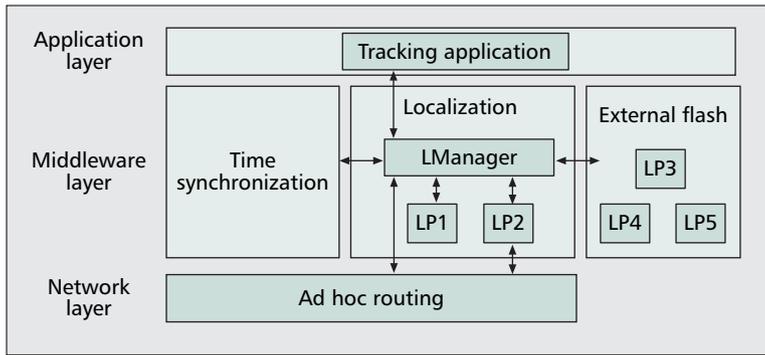
Some sensor networks have anchor nodes, which are nodes with known locations, deployed uniformly throughout the network such that non-anchor nodes will always have radio connectivity to several anchors at once. The locations of the non-anchor nodes can then be computed by the Centroid algorithm [2], which localizes a node to the average of the coordinates of all anchors with which it has connectivity. Hence, the Centroid localization scheme requires a dense set of anchor nodes that all periodically broadcast their locations. The Centroid localization scheme assumes an idealized radio model: spherical radio propagation and identical transmission range for all radios of anchor nodes. When the radio hardware approximates these requirements well, and the anchor density is low, accuracies on the order of half the communication range can be obtained. For higher localization accuracy, higher anchor density is required.

In many sparsely deployed sensor networks, it is not practical to have anchor nodes uniformly distributed throughout the network; only a few nodes out of hundreds or thousands can be manually localized, so most non-anchor nodes will not have connectivity with any anchor nodes at all. DV-Hop [3] uses the hop count between sensor nodes and anchors to infer the distances among them. The protocol contains two phases. In the first phase anchors flood the network with beacons, and each node records the shortest hop count to each of the anchors. In the second phase, which occurs when one anchor receives beacons from other anchors, anchor nodes estimate the physical distance (Euclidian) of one radio hop, also called the correction factor (which is, simply, the ratio of the distance between two anchors and the number of hops between them). This correction factor is flooded back to the network, allowing sensor nodes to infer physical distances to anchor nodes (as the product between the number of hops to an anchor and the correction factor). After obtaining the distance estimates to three or more anchors, a sensor node employs a multilateration (similar to that of GPS) for iteratively improving its location estimation. DV-Hop does not work well in irregular deployments, where the true distance between two points can be artificially inflated due to voids present between the endpoints. While the high overhead (i.e., communication overhead) and low accuracy of the scheme may not make DV-Hop an ideal solution for deployments where high accuracy in node positions is needed, the fact that only connectivity information is used makes it very attractive (especially for scenarios where more precise node localization schemes do not work, i.e., complex environments).

Protocol Composition

Protocol composition has been frequently obtained by layering protocols vertically, in stacks. Horus [7], for example, is based on a vertical stack of protocols, where the events are strictly passed from one layer to an adjacent one. A nonhierarchical protocol composition framework is JGroup/ARM [8]. It introduces the concept of a dependency graph among protocols/layers (graph obtained by each layer registering its interest in other layers). The work proposed in [1] has similarities with HLS [9] (which proposes a hierarchical framework for composing soft real-time schedulers) and [10] (which uses sensing multimodality for robust localization).

Cryptographic systems frequently run multiple concurrent instances of a single protocol and share part of their state.



■ Figure 1. Software architecture for robust node localization.

Analyzing the strength of the entire system is rather difficult. Consequently, [11] proposes a solution that analyzes protocol instances and uses composition theorems to prove the strength of the security for the entire system. The notion of *universal composability* is being put forward.

In this section we surveyed a representative set of localization scheme for wireless sensor networks and research areas where protocol composition was successfully applied. We pointed out some limitations of existing localization schemes in meeting the requirements posed by the node localization problem in outdoor complex environments: the need for higher density of anchor nodes, hardware requirements (e.g., ultrasound or GPS), the assumption of circular radio range, the need for high node density or uniformly distributed sensor nodes (i.e., irregular topologies cause problems), and the assumption of line of sight between a central device and sensor nodes. To address these limitations, localization protocol composition was proposed in [1]. This allows us to overcome the situation when an individual protocol fails (due to its assumptions) through successful execution of a different localization protocol.

Robustness through Composable Localization Schemes

The main idea in ensuring robustness for node localization (i.e., meeting the multidimensional requirements in complex environments) is to develop a framework that allows the execution of multiple node localization schemes in a WSN deployment. The goal is to reduce the impact of any single localization protocol to the average location estimation error in a particular WSN deployment and to address the trade-offs posed by the multidimensional requirements for node localization. In the remaining part of this section we present the architecture for node localization, and the framework for robust node localization through composable protocols.

Node Localization Architecture

The software architecture for node localization is shown in Fig. 1. We assume that the sensor network is connected (i.e., all sensor nodes can communicate, single-hop or multihop, with a base station), that a time synchronization protocol has already executed, and that the sensor network is static.

The components of interest for node localization are the localization manager (LManager), the localization protocols (LP $_i$, $i = 1, 2, \dots$) that are executing, and the interface used for their integration. The LManager provides the application layer with location services, such as node position. The LPs implement a generic interface for starting and stopping their execution. The LManager is responsible for dynamically loading localization schemes from the external flash memory and for their execution. As shown, the LPs may or may not need to use the ad hoc routing layer for communication with neigh-

bor nodes. The three main parts, the LManager, LPs, and interface, are further described below.

Localization interface: This is the contract between the LManager and LPs. All LPs need to implement the commands that are part of the interface, and the LManager needs to implement the events specified in the interface. The manager can configure an LP with a location through a *set-Location* command. This is required since some protocols make use of nodes that have knowledge about their location (i.e., anchor nodes). Through use of the *setLocation* command, the manager can set a particular node to act as an anchor. The pair *startLocalization* and *localizationDone* implement a standard split-phase operation in TinyOS. The LP notifies the manager about the result of the localization operation through the *localizationDone* event. The LManager has the ability to stop the execution of a localization protocol or inquire about the status of execution (initializing, executing, completed) through use of the *stop* and *getStatus* commands, respectively.

Localization protocol: This is an implementation of a protocol for node localization in WSNs. The implementation does not make use of the fact that its execution is part of a framework in which other protocols may execute simultaneously or subsequently. One slight “adaptation” of an LP to the proposed framework is that after a predetermined period of time, the protocol needs to indicate whether or not the localization scheme was successful. This is typically accomplished by setting a timer, at the expiration of which the protocol informs the LManager about the status of the localization phase and the node location if the scheme was successful.

Localization manager: The manager contains a hierarchy of localization schemes to be executed in the network. In addition to loading and executing localization protocols, the manager is responsible for ensuring that neighboring nodes execute the same localization protocol. The manager also aggregates results (i.e., node location information) from multiple localization protocols that execute simultaneously, or passes location information between protocols that execute in sequence. A subsequent protocol may use location information about by a previous scheme. The execution of the LManager is described in more detail next.

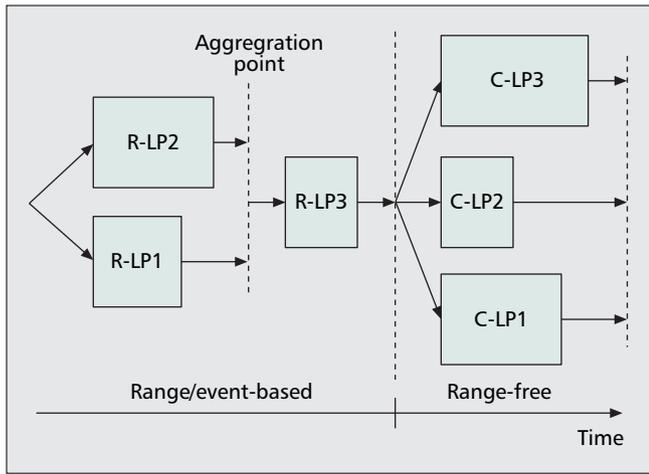
Hierarchical Framework for Node Localization

The main idea in designing how localization protocols are executed is (based on the taxonomy presented earlier) to use a hierarchical, multiphase, multiprotocol operation.

The localization protocols that are part of the proposed hierarchy are the protocols available to the LManager (stored in flash memory). A generic hierarchical structure is shown in Fig. 2 where rectangular boxes represent LPs, and vertical lines represent aggregation points (i.e., location results obtained by different protocols are aggregated).

As shown, the framework operates in two phases. The goal of the first phase (R-LP* boxes in Fig. 2) is to achieve the highest location accuracy possible (given the existing capabilities of sensor nodes). Localization schemes that are candidates for this phase are typical range/event-based schemes. The goal of the second phase (C-LP* boxes in Fig. 2) is to estimate the locations of nodes for which the first phase was not successful.

The node localization process starts, for each sensor node, from the top of the hierarchy. The LManager is responsible for loading all localization schemes that execute simultaneously between two aggregation points. When the actively running localization schemes finish execution and an aggregation point is reached, the LManager “aggregates” through a weighted



■ Figure 2. Hierarchical framework.

mean the node locations produced by the LPs. For example, R-LP1 and R-LP2 in Fig. 2 could be GPS and Spotlight localization schemes. Because equipping all sensor nodes with GPS is not a feasible solution, only a small set of sensor nodes will successfully obtain a location estimate from GPS. Nodes that are in line of sight of a Spotlight device are able to obtain their location from executing the Spotlight localization scheme.

If a node obtains a location (and reaches an aggregation point), it stops following the hierarchical localization graph. The reason for this is that the order of execution of localization schemes is such that later localization schemes are unlikely to produce more accurate location estimation.

As shown in Fig. 2 by the R-LP3 box, it is possible to have sequential execution of a localization scheme. If a scheme is considered “heavier” because it utilizes resources (e.g., GPS), it can be scheduled to execute only on nodes for which previous schemes have failed. When designing the hierarchical structure for node localization, the designer needs to consider that not all nodes may be available for execution of this heavier localization protocol (some nodes successfully localized using earlier localization schemes).

When the LManager finishes the execution of the first phase, it is ready to execute the second phase if no location information was obtained. The second phase is based on localization schemes that use proximity to anchor nodes. Since neighboring nodes may have obtained their location, the LManager needs to coordinate with its neighbors. This is accomplished by broadcast of a HELP message by nodes that do not know their locations. The LManager is responsible for broadcasting the HELP message. Only nodes that do not have a location forward the HELP message, and they forward only the first HELP message. This has an effect of controlled flooding of HELP messages. If a node was localized and receives a HELP message, it immediately transitions to the second phase of operation and acts as an anchor. It thus broadcasts its location in a single hop or multihop, depending on the type of LPs included in the second phase of the hierarchy.

Implementation and Evaluation

For the performance evaluation of the framework, TinyOS implementations of a set of representative localization schemes were used [1]: Spotlight [4], GPS, Centroid [2], and DV-Hop [3]. Due to the fact that TinyOS does not have capabilities for dynamically loading modules, all localization schemes were statically linked with the LManager.

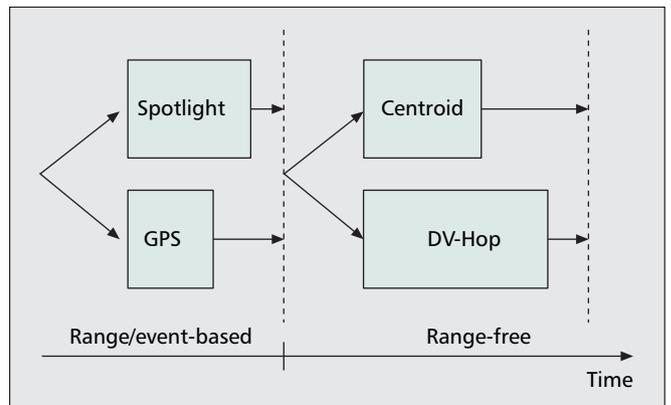
The hierarchical framework we evaluated is depicted in Fig.

3. In this framework GPS and Spotlight execute during the first phase of the framework, and Centroid and DV-Hop are chosen as LPs to be executed during the second phase. The manner in which Centroid and DV-Hop work is as follows. Let us consider two scenarios. In the first an isolated sensor node fails to obtain its location from a range/event-based scheme. In the second one a large group of nodes (multihop radius) fail to obtain their locations. In the first case the HELP message sent by the LManager on the isolated node is received by the neighboring nodes (successfully localized). They immediately transition to the second phase and act as anchors, broadcasting their locations (simultaneously executing Centroid and DV-Hop). In order to limit the area where beacon nodes are forwarded, we enforce that anchor nodes do not forward beacons from other anchors. In our scenario the isolated node would then successfully compute its location as the average of locations obtained from the Centroid and DV-Hop schemes. In the second scenario the nodes that have not been localized and are positioned at the perimeter of the “void” area are able to localize themselves using both Centroid and DV-Hop (and aggregate the results), while the non-localized nodes that are multiple hops away from the perimeter will be successfully localized using DV-Hop.

The framework depicted in Fig. 3 was simulated in TOSSIM. We simulate a sensor network of 400 nodes, deployed in a 300×300 ft² area. The radio range was fixed at 50 ft. We assume that 10 percent of sensor nodes are equipped with GPS devices. To simulate harsh environments, we designated a 200×200 ft² area in the center of the deployment to be heavily wooded and hence lack a line of sight. In addition, any node that is not positioned in the wooded area has a 15 percent probability of not having a line of sight with satellites or aerial vehicles (and hence cannot be localized through GPS or Spotlight).

The experimental results of 10 simulations with random seeds are shown in Fig. 4. As shown, Spotlight and GPS have the highest accuracies (2–5 ft), but fail to localize all the nodes in the sensor network. On the other hand, DV-Hop localizes the entire network, but the accuracy is low (14 ft average localization error). The Centroid scheme also has low accuracy and fails to localize the entire network due to the low number of anchor nodes. The Robust Localization scheme, however, successfully localizes the entire WSN and has an average localization error of about 7 ft. Even though individual node localization schemes fail (GPS fails for 90 percent of nodes and Spotlight for 50 percent of nodes), the Robust solution localizes all sensor nodes with a good average localization accuracy.

The price paid for achieving higher accuracy than individu-



■ Figure 3. Localization system.

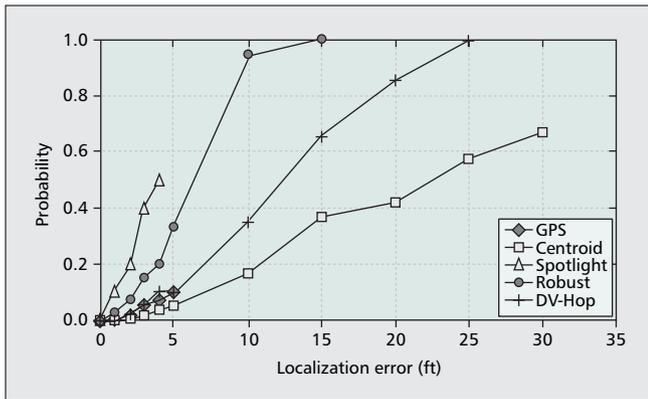


Figure 4. Localization error CDF.

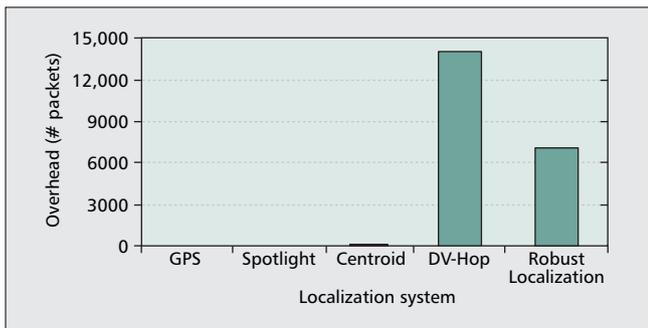


Figure 5. Localization protocol overhead.

al components is shown in Fig. 5: the overhead (which considers only communication overhead). While GPS and Spotlight have no communication overhead, the Robust Localization scheme has a higher overhead than Centroid, but lower than DV-Hop, which requires flooding of the entire network. The overhead is relatively modest when considering the obtained accuracy in node location.

The results obtained show that in addition to ensuring a 100 percent success rate for localizing a WSN, the framework reduces the average localization error with respect to schemes that use connectivity information. More complete evaluation of the costs (e.g., energy consumption, localization time) incurred by the framework are left for future research.

Conclusions

Despite the efforts made, no system has emerged as a robust, practical solution for the node localization problem in realistic complex outdoor environments. In this article we argue that the existing localization algorithms individually work well for single sets of assumptions. These assumptions do not always hold, as in the case of outdoor complex environments. This article shows that complex, more robust localization systems can be built by composing localization schemes that have limitations.

The described localization framework is just a small part of the more complete solution that will allow a nonexpert to build a robust and efficient localization system for a particular

WSN deployment. Several areas require further research: development of higher-level abstractions for composing localization protocols (we aim to provide a programming tool, with a script-like language, for building a localization system from individual localization protocols); development of an analysis tool that evaluates the correctness of a hierarchical localization framework and possibly gives soft-guarantees (e.g., largest expected localization error and the overhead required to achieve it); optimization of simultaneous executions of protocols that use, for example, radio communication (instead of having each protocol send/receive its own messages, an aggregation of data contained in these messages may significantly reduce the communication overhead); and analysis of robustness against malicious attacks (due to localization protocol multi-modality, it is more difficult for an attacker to compromise the integrity of the node localization service).

References

- [1] R. Stoleru, J. A. Stankovic, and S. H. Son, "Robust Node Localization for Wireless Sensor Networks," *Proc. IEEE Wksp. Embedded Networked Sensors*, 2007.
- [2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very 16 Small Devices," *IEEE Pers. Commun.*, Oct. 2000.
- [3] D. Niculescu and B. Nath, "DV-Based Positioning in Ad Hoc Networks," *Telecommun. Sys.*, 2003.
- [4] R. Stoleru et al., "A High-Accuracy Low-Cost Localization System for Wireless Sensor Networks," *Proc. ACM Conf. Embedded Networked Sensor Sys.*, 2005.
- [5] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Compass for Context-Aware Mobile Applications," *Proc. Int'l. Conf. Mobile Computing and Networking* 2001.
- [6] M. Maroti et al., "Radio Interferometric Geolocation," *Proc. ACM Conf. Embedded Networked Sensor Sys.*, 2005.
- [7] R. van Renesse et al., "A Framework for Protocol Composition in Horus," *Principles of Distributed Computing*, 1995.
- [8] H. Meling, "Non-Hierarchical Dynamic Protocol Composition in Jgroup/ARM," *Proc. Norsk Informatikkonferanse*, 2006.
- [9] J. Regehr and J. A. Stankovic, "HLS: A Framework for Composing Soft Real-Time Schedulers," *Proc. IEEE Real Time Sys. Symp.*, 2001.
- [10] L. Girod and D. Estrin, "Robust Range Estimation Using Acoustic and Multimodal Sensing," *Proc. IEEE/IRJ Int'l. Conf. Intelligent Robots and Sys.*, 2001.
- [11] R. Canetti, "Universally Composable Security: A New Paradigm for Cryptographic Protocols," *Proc. Symp. Foundations of Comp. Sci.*, 2001.

Biographies

RADU STOLERU (stoleru@cs.tamu.edu) is an assistant professor in the Department of Computer Science at Texas A&M University. His research interests are in deeply embedded wireless sensor systems, distributed systems, embedded computing, and computer networking. He has authored over 27 papers and won the Outstanding Graduate Student Research Award from the Department of Computer Science, University of Virginia in 2007. He received a Ph.D. in computer science from the University of Virginia in 2007.

JOHN A. STANKOVIC [F] is the BP America Professor in the Computer Science Department at the University of Virginia. He is a Fellow of the ACM. He also won the IEEE Real-Time Systems Technical Committee's Award for Outstanding Technical Contributions and Leadership (inaugural winner). He also won the IEEE Technical Committee on Distributed Processing's Distinguished Achievement Award (inaugural winner). His research interests are in distributed computing, real-time systems, operating systems, and wireless sensor networks. He received his Ph.D. from Brown University.

SANG HYUK SON is a professor in the Department of Computer Science at the University of Virginia. He is Chair of the IEEE Technical Committee on Real-Time Systems. He has served as an Associate Editor of *IEEE Transactions on Parallel and Distributed Systems*, and is currently serving as an Associate Editor for *IEEE Transactions on Computers and Real-Time Systems*. His research interests include real-time systems, data services, QoS management, and wireless sensor networks.