An Overview of the VigilNet Architecture

Tian He, Liqian Luo, Ting Yan, Lin Gu, Qing Cao, Gang Zhou, Radu Stoleru
Pascal Vicaire, Qiuhua Cao, John A. Stankovic, Sang H. Son and Tarek F. Abdelzaher
Department of Computer Science
University of Virginia, Charlottesville, VA 22903

Abstract

Battlefield surveillance often involves a high element of risk for military operators. Hence, it is very important for the military to execute unmanned surveillance by using large-scale wireless sensor systems. This invited paper summarizes the architecture of the VigilNet system – a long-term real-time networked sensor system for military surveillance. Specifically, we review the design of several major subsystems within VigilNet including sensing and classification, localization, tracking, networking, power management, reconfiguration, graphic user interface, and the debugging subsystem. High-level programming abstractions are also presented. This is a balanced design to achieve real-time response, high confidence detection, accurate tracking and energy efficiency simultaneously.

1. INTRODUCTION

Recently, many efforts have been undertaken to support new military applications by using large-scale wireless sensor networks. Unmanned real-time surveillance is one of the most promising applications. By combining the computation, sensing, actuation and wireless networking together, large-scale sensor networks have several advantages over many other distributed systems. First, sensor networks can be quickly deployed in an infrastructure-free environment, which is highly desired in military operations. Second, the redundancy introduced by a large-scale dense deployment makes a sensor system robust to node failures, which is critical in a hostile environment. Third, massively distributed in-network data processing allows exploration of new techniques for detection and classification, not posible with only a small number of devices. Along with these advantages, however, several challenges arise. The constrained resources in wireless sensor nodes, such as limited memory, power, processing, and communication bandwidth, impose problems previous research did not need to address. To realize the vision of wireless surveillance systems, many research efforts have been published that address challenging problems concerning networking [14], self-organization [4], energy-conservation [27, 30] and tracking [16] in this type of systems. However, most existing efforts address individual protocols or subproblems in the design space. Few systems actually provide a complete architecture and a running implementation tested in outdoor environments. The VigilNet project is, therefore, distinguished in this aspect. It is a large-scale sensor network system which has been successfully designed, built, demonstrated and delivered to the military for realistic deployment. To accomplish different mission objectives, the VigilNet system consists of 40,000 lines of code, supporting multiple existing mote platforms including MICA2DOT, MICA2, and XSM. To accommodate various mission requirements, VigilNet is dynamically reconfigurable and reprogrammable. For example, we can flexibly explore trade-offs between surveillance quality parameters and network lifetime by adjusting various system parameters online.

The remainder of this paper is organized as follows: Section 2 discusses the related work. Section 3 presents the overarching architecture of VigilNet and the details of individual subsystems. We list several lessons learned from our experience in Section 4 and conclude the paper in Section 5.

2. RELATED WORK

Middleware services form the basis of sensor network systems. Localization is a key service to identify the locations from where sensor readings are obtained. Two categories of localization have been proposed: range-based schemes [22, 24] and range-free schemes [2, 11]. The former category uses absolute point-to-point distance estimates (range) or angle estimates to localize nodes; The latter makes no assumptions about the availability of such information. Time synchronization is another critical middleware service. The reference broadcast scheme (RBS) proposed in [7] maintains information about the phase and frequency of each pair of clocks in the neighborhood of a node. While RBS achieves a precision of about 1 μs, the message overhead in maintaining the neighborhood information is high and may not be energy-efficient in large-scale systems. Maroti [21] synchronizes the network through limited flooding with timestamp values reassigned at intermediate nodes immediately prior to transmission. This scheme reduces the synchronization error introduced by uncertainty due to MAC contention. Our VigilNet system uses a variation of this approach. In addition, power management is employed to ensure network longevity. Other important protocols developed for modern sensor network hardware [5, 6] include medium access control [23, 31], sensing coverage [27, 30], energy aware routing [29], data aggregation [20, 10], topology management [4] and energy-aware applications [12, 26, 28].

With the assistance of various middleware services, several outdoor sensor network systems have recently been built. The Great Duck Island Project [26] explores long-term habitat monitoring in remote environments. ZebraNet [15] focuses on wildlife tracking in Africa. The Extreme Scaling project [6] investigates scalability issues in surveillance systems. The shooter localization system [8] combines precise time synchronization with accurate acoustic sensing to localize positions of snipers. The Wisden system [28] monitors the health of building structures by continuously retrieving structural response data. To complement the aforementioned efforts, VigilNet aims at building a practical military surveillance system, which can survive in harsh and hostile environments for a long period of time, and exhibit a high detection, classification and tracking performance. These requirements necessitate the design of unique solutions to various sensing, communication, and tracking problems [9, 17, 18, 19, 25, 32] and call for seamless integration of the resulting middleware components [12].

3. VIGILNET ARCHITECTURE

The VigilNet system has a layered architecture as shown in Figure 1. This architecture provides an end-to-end solution for supporting military surveillance applications. As an overview, this paper focuses on the design of individual components of this system, but omits the details of system implementation and performance evaluation. Such details can be found in other publications by the authors [9, 12, 17, 18, 19, 25, 32]

3.1 Sensing Subsystem

Sensing is the basis for any surveillance system. The VigilNet sensing subsystem implements detection and classification of targets using continuous online sensor calibration (to a changing environment) and frequency filters to determine critical target features. These filters extract the target signatures from a specific spectrum band, eliminating the burden of applying a computation-intensive Fast Fourier Transform. The sensing subsystem contains three detection algorithms for the magnetic sensor, acoustic sensor and passive infrared sensor (PIR), respectively.

- The magnetic sensor detection algorithm computes two moving averages of most recent magnetic readings. The slower moving average, with more weight on previous readings, establishes a baseline to follow the thermal drift noise caused by the changing temperature during the day. The faster moving average, with more weights on the current reading, detects the swift change in magnetic filed caused by ferrous targets. To make a detection decision, the difference between the two moving average values is compared to a dynamic threshold, which is established during the calibration phase.
- The acoustic sensor detection algorithm uses a lightweight power-based approach. It first computes a moving average of multiple recent acoustic readings, then establishes an auto-adapting acoustic threshold by calculating a moving standard deviation of readings over a certain time window. If an acoustic reading is larger than the sum of the moving average and its corresponding moving standard deviation, we consider it is a crossover. If the number of crossovers exceeds a certain threshold during a unit of time, this algorithm signals a detection to the upper layer components.
- The passive infrared sensor is designed to sense changes in thermal radiation that are indicative of motion. When there is no movement, the thermal reading is stable and does not trigger detections. If an object is moving in front of a PIR sensor, this object causes a thermal disturbance, triggering the PIR. Most moving objects, such as shaking leaves, rain drops, and vehicles, can trigger the PIR sensor. However, different thermal signatures generate trigger events with different frequencies. Low frequency detections ($< 2Hz$) are normally triggered by wind-induced motion and other slow moving objects. On the other hand, fast-moving targets such as vehicles generate signals with a much higher frequency. Therefore, it is sufficient to design a high pass ARMA filter to filter out the frequency components lower than 2Hz. Similar to other detection algorithms, this threshold is dynamically adaptated to accommodate the changing environment. For example, thermal noise is much higher in a hot and humid environment than that in a dry and cool one. In fact, thermal and humidity variations over the course of a day can significantly change threshold values.

Due to the space constraints, we do not detail the specific implementation of each sensing algorithm. More information on this subsystem can be found in [9].

3.2 Context-Awareness Subsystem

Sensed data is meaningful only when it is interpreted along with the context in which it is obtained. For example, a temperature reading is useless, if we don't know where and when such value is measured. The Context-Awareness subsystem comprises lower-level context detection components such as time synchronization and localization. These components form the basis for implementing other subsystems, such as the tracking subsystem. Localization ensures that each node is aware of its location, so that we can determine the location of detected targets. Time synchronization is responsible for synchronizing the local clocks of nodes with the clock of the base station, so that every node in the network has a consistent global view of time. Combining time synchronization and localization, we are able to estimate the velocity of targets.

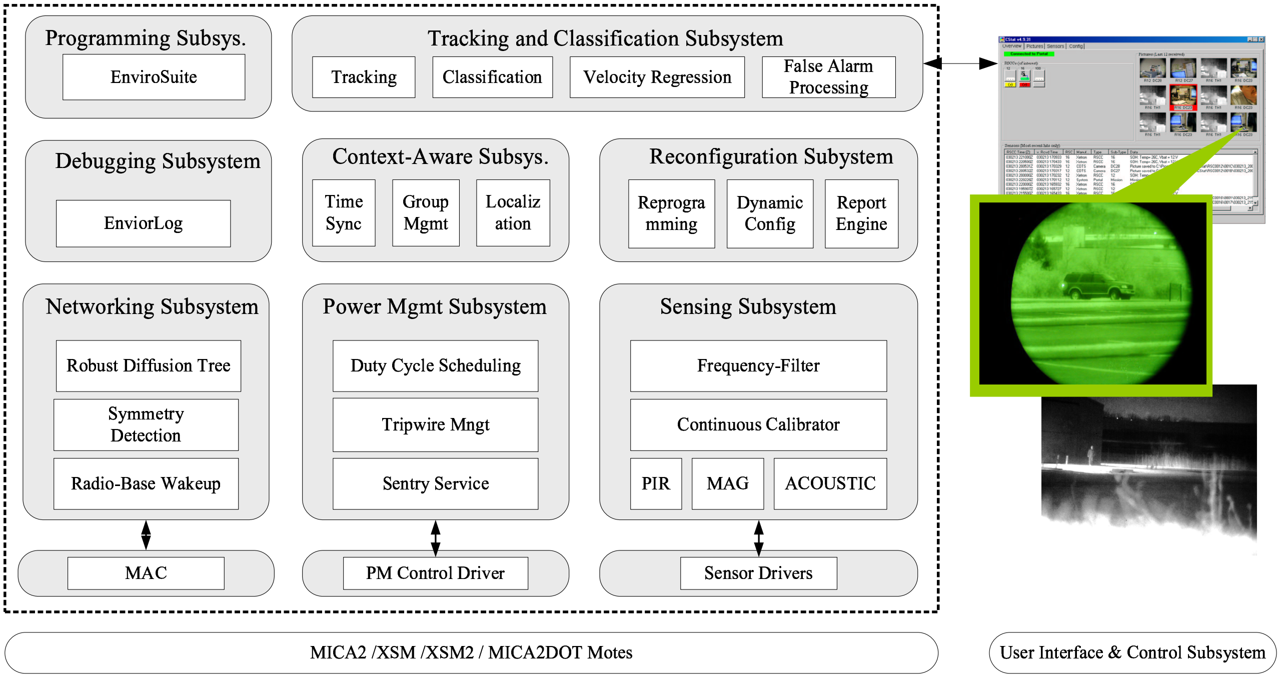

Figure 1: The VigilNet System Architecture

In the VigilNet system, we adopt a variation of the time synchronzation protocol developed by Maroti [21] and we design and implement a walking GPS solution [25], which assigns nodes their location at the time they are deployed. We are currently investigating realistic implementations of more dynamic localization schemes such as those described in [11].

3.3 Tracking and Classification Subsystem

When a target is detected by a set of nearby nodes, the tracking component creates a group. All nodes that detect the same event join the same group. The main contribution of the tracking component is to ensure the uniqueness (one-to-one mapping of external events to logical groups) and consistent identification (immutability of the mapping function) of targets, as long as targets are far enough apart from each other or have different signatures. When targets are very near each other and process an identical signature, we provide a disambiguation mechanism based on their path-histories. More information on this subsystem can be found in [1] and [18].

3.4 Networking Subsystem

After VigilNet collects detection information about incoming targets through the tracking and classification subsystem, it needs to deliver detection reports back to the control center through a multi-hop network. The networking subsystem consists of three major components: a link symmetry detection service, a robust diffusion service and a radio-based wakeup service. Low power radio components, such as Chipcon CC1000 used by MICA2 [5], exhibit very irregular communication patterns. To address this problem, we design a Link Symmetry Detection (LSD) module to reduce the impact of radio irregularity on upper layer protocols. The main idea of the LSD module is to build a symmetric overlay on top of the anisotropic radio layer, so that those protocols whose correctness depends on the link symmetry can be used without modification. More details on this solution can be found in [32]. The robust diffusion service utilizes a well-known path-reversal technique [14]. Basically, a base node disseminates tree construction requests to the rest of the network with a running hop-count initialized to zero. Requests are flooded outwards with hop-count incremented at every intermediate hop. After receiving tree construction requests, nodes establish multiple reverse paths towards the sending node. As a result, a multi-parent diffusion tree is constructed with the base node residing at the root. The Radio-based Wakeup service is designed to ensure end-to-end data delivery even intermediate nodes are in the dormant state (due to power management). To support the illusion of on-demand wakeup, a dormant node wakes up and checks radio activity periodically (e.g., for five milliseconds every several hundred milliseconds). If no radio activity is detected, this node goes back to sleep. Otherwise, it remains active to receive and relay messages. If an active node wants to wake up all neighboring nodes, it only needs to send out a message with a long enough preamble to last longer than the checking period of the dormant nodes.

3.5 Graphic User Interface and Control Subsystem

The networking subsystem delivers the reports to one or more command and control centers, where the Graphic User Interface and Control subsystem is located. This subsystem provides three major functionalities. First, it accepts the reports from the sensor field and displays such information graphically to the mission operators. Second, it allows the mission operators to disseminate the system configurations through the reconfiguration subsystem. Third, based on the initial detections from the sensor field, it makes final decisions on whether to wake up more advanced sensors. These advanced sensors not only classify the type of targets but also differentiate the model of the targets. Since they are extremely power consuming, they are normally turned off and only used when awakened by initial detections coming from the sensor field.

3.6 The Power Management Subsystem

One of the key design objectives of the VigilNet system is to increase the system lifetime to $3 \sim 6$ months in realistic deployment. Due to the small form factor and low-cost requirements, sensor devices such as XSM motes [6] are equipped with limited power sources (e.g., two AA batteries). The normal lifetime for such a sensor node is about 4 days if it remains active all the time. To bridge such a gap, we add a power management subsystem. Among all the middleware services, the tripwire service, sentry selection, duty cycle scheduling and wakeup service form the basis for the power management subsystem. We organize them into a multi-dimensional architecture. At the top level, we use the tripwire service to divide the sensor field into multiple sections, called tripwire sections. A tripwire section can be scheduled either into an active or a dormant state at a given point of time. When a tripwire section is dormant, all nodes within this section are in a deep-sleep state to conserve energy. When a tripwire section is active, we apply a second-level sentry service within this section. The basic idea of the sentry service is to select only a subset of nodes, which we define as *sentries*, to be in charge of surveillance. Other nodes, defined as *non-sentries*, can be put into a deep-sleep state to conserve energy. Rotation is periodically done among all nodes, selecting the nodes with more remaining energy as sentries. At a third-level, since a target can normally be sensed for a non-negligible period of time, it is not necessary to turn sentry nodes on all the time. We can schedule a sentry node in and out of sleep state to conserve energy. The sleep/awake schedule of a sentry node can be either independent of other nodes or coordinated with that of others in order to further reduce the detection delay and increase the detection probability. More information on nodes' duty cycle scheduling can be found at [3].

3.7 The Reconfiguration Subsystem

The VigilNet system is designed to accommodate different node densities, network topologies, sensing and communication capabilities and different mission objectives. Therefore, it is important to design an architecture that is flexible enough to accommodate various system scenarios. The reconfiguration subsystem addresses this issue through two major components: a multi-hop reconfiguration module and a multi-hop reprogramming module. The reconfiguration module allows fast parameter tuning through a data dissemination service, which supports limited flooding. Data fragmentation and defragmentation are supported in the reconfiguration subsystem to allow various sizes of the system parameters. The reprogramming module provides a high level of flexibility by reprogramming the nodes. More information on reprogramming can be found at [13].

3.8 The Debugging Subsystem

Debugging and tuning event-driven sensor network applications such as VigilNet are of great difficulty for the following reasons: (i) big discrepancies exist between simulations and empirical results due to various practical issues (e.g., radio and sensing irregularity) not captured in simulators, which makes them less accurate; (ii) In-field tests of the system require walking or driving through the field to generate events of interest actively, which makes in-field tests extremely costly. To address this issue, we add a debugging subsystem called EnviroLog [19] into VigilNet. EnviroLog logs environmental events into non-volatile storage on the motes (e.g., the 512 KB external flash memory) with timestamps. These events can then be replayed in their original time sequence on demand. EnviroLog reduces experimental overhead by eliminating the need to physically re-generate events of interest hundreds of times for debugging or parameter tuning purposes. It also facilitates comparisons between different evaluated protocols.

3.9 The Programming Interface

The programming interface is an extension of our prior work on EnviroSuite [18]. It adopts an object-based programming model that combines logical objects and physical elements in the external environment into the same object space. EnviroSuite differs from traditional object-oriented languages in that its objects may be representatives of physical environmental elements. EnviroSuite makes such objects the basic computation, communication and actuation unit, as opposed to individual nodes. Thus, it hides implementation details such as individual node activities and interactions among nodes from developers. Using language primitives provided by EnviroSuite, developers of tracking or monitoring applications simply can specify object creation conditions (sensory signatures of targets), object attributes (monitored aggregate properties of targets), and object methods (desired computation, communication or actuation in the vicinity of targets). Such specifications can be translated by an EnviroSuite compiler into real applications that are directly executable on motes. When defined object conditions are met, dynamic object instances are automatically created by the run-time system of EnviroSuite to collect object attributes and execute object methods. Such instances float across the network following the targets they represent, and are destroyed when the targets disappear or move out of the network.

3.10 System Work Flow

To avoid interference among different operations, VigilNet employs a multiple-phase work flow. The transition between phases is time-driven, as shown in Figure 2. Phases I through VII comprise the initialization process which normally takes about several minutes. In Phase I, the reconfiguration subsystem initializes the whole network with a set of parameters. In Phase II, the context-awareness subsystem synchronizes all nodes in the field with the master clock at the base, followed by the localization process in Phase III. In Phase IV and V, the networking subsystem establishes a robust diffusion tree for end-to-end data delivery. Phase VI invokes the power management subsystem to activate tripwire sections and select a subset of the nodes as sentries. The system layout, sentry distribution and network topology are reported to the graphic user interface and control subsystem in Phase VII. After that, the nodes enter into the main phase VIII– the surveillance phase. In this phase, nodes enable the power management subsystem in absence of significant events, and activate the tracking subsystem once a target enters into the area of interest.

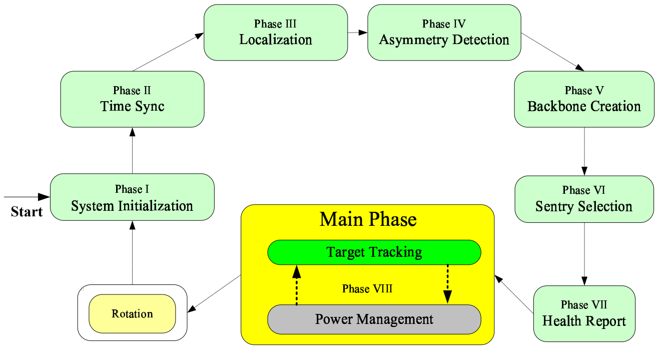

Figure 2: Phase Transition and Rotation

4. LESSONS LEARNED

We obtained valuable lessons during the process of building the VigilNet system. We share them here to assist similar efforts in other application domains.

1. **False Alarm Reduction:** False alarms introduce unnecessary energy consumption and inappropriate reactions. To deal with transient false alarms caused by distortion of sensing readings, one can use a simple exponential weighted moving average (EWMA). To address false alarms caused by slow-changing environments, one needs to use adaptive detection thresholds. To address persistent false alarms due to errors in a single sensor device, one can utilize in-network aggregation of inputs from a group of nodes to detect such an anomaly. In the worst case, when multiple persistent false alarms are generated simultaneously, one can filter out such false alarms by analyzing spatial and temporal correlations among the consecutive reports at the base station. More information on false alarm reduction can be found at [9].

2. **Communication Reliability:** Communication reliability is affected by link quality. Poor link quality can be addressed by retransmissions, however with a very high overhead. With high link redundancy in a dense sensor network, it is beneficial to carefully select high quality links for data delivery than to use FEC/ARQ techniques to improve transmission reliability over poor radio links. To select high quality links, one can use the link symmetry detection service we developed [32] for the VigilNet system. Moreover, it is beneficial to provide reliability selectively according to the semantics of the payload. Application-level mechanisms are appropriate to achieve such differentiation.

3. **Energy Bottleneck:** Although the radio power draw is very high, the amortized power draw from communication is actually very low in a surveillance system, due to its very low duty cycle. In contrast, we need to monitor the environment continuously to ensure detection. Accordingly, the amortized power draw for sensing is much higher than that of communication. Therefore, the most effective way to conserve energy is to reduce sensing redundancy (turn off a subset of nodes) in the absence of significant events and to activate nodes on-demand. In addition, a promising direction is to utilize hardware-driven wakeup functions to significantly reduce energy consumption during surveillance.

4. **Other Lessons:** Debugging and performance tuning in distributed sensor networks are extremely time consuming, especially during field tests. It is critical to have appropriate built-in system support for these functions, such as the reconfiguration subsystem and the debugging subsystem [19]. Second, in addition to the hardware and software, the mechanical design is very important to ensure good system performance. For example, enclosure design can significantly affect the sensitivity of senor nodes. Third, since the sensor nodes fail at a much higher rate in hostile outdoor environments, self-healing should be supported by every protocol integrated into the system.

5. CONCLUSIONS

This paper presents the design, implementation and evaluation of VigilNet – an integrated sensor system for long-term surveillance. We describe the functionalities of different subsystems within VigilNet. From our experience, we believe that surveillance using wireless sensor networks is a very promising direction. It has a lot of advantages such as fast ad hoc deployment, fine-grained robust sensing and tracking, low-power consumption, and low cost. VigilNet presents a proof that viable surveillance systems can be implemented and successfully deployed on current motes hardware.

6 ACKNOWLEDGEMENT

This work was supported in part by the DAPRPA IXO offices under the NEST project (grant number F336615-01-C-1905), the MURI award N00014-01-1-0576 from ONR and NSF grant CCR-0098269.

References

[1] B. M. Blum, P. Nagaraddi, A. Wood, T. F. Abdelzaher, S. Son, and J. A. Stankovic. An Entity Maintenance and Connection Service for Sensor Networks. In *The First Intl. Conference on Mobile Systems, Applications, and Services (MobiSys)*, May 2003.

[2] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less Low Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications Magazine, Special Issue on Networking the Physical World*, 7(4):28–34, August 2000.

[3] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic. Towards Optimal Sleep Scheduling in Sensor Networks for Rare Event Detection . In *The Fourth International Conference on Information Processing in Sensor Networks*, 2005.

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *6th ACM MOBICOM Conference*, 2001.

[5] CrossBow. *Mica2 data sheet*. Available at http://www.xbow.com.

[6] P. Dutta, M. Grimmer, A. Arora, S. Biby, and D. Culler. Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events. In *IPSN'05*, 2005.

[7] J. Elson and K. Romer. Wireless Sensor Networks: A New Regime for Time Synchronization. In *Proc. of the Workshop on Hot Topics in Networks (HotNets)*, October 2002.

[8] G.Simon and et. al. Sensor Network-Based Countersniper System. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[9] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, T. He, A. Tirumala, Q. Cao, J. A. Stankovic, T. Abdelzaher, and B.H.Krogh. Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments. In *In submission*, 2004.

[10] T. He, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. AIDA: Adaptive Application Independent Data Aggregation in Wireless Sensor Networks. *ACM Transactions on Embedded Computing System, Special issue on Dynamically Adaptable Embedded Systems*, 2004.

[11] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-Free Localization Schemes in Large-Scale Sensor Networks. In *Proc. of the Intl. Conference on Mobile Computing and Networking (MOBICOM)*, September 2003.

[12] T. He, S. Krishnamurthy, J. A. Stankovic, and T. Abdelzaher. An Energy-Efficient Surveillance System Using Wireless Sensor Networks. In *The Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2004.

[13] J. Hui and D. Culler. The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *the Sixth Annual International Conference on Mobile Computing and Networks*, 2000.

[15] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Proc. of ASPLOS-X*, October 2002.

[16] J. Liu, J. Reich, and F. Zhao. Collaborative In-Network Processing for Target Tracking. *J. on Applied Signal Processing*, March 2003.

[17] L. Luo, T. Abdelzaher, T. He, and J. Stankovic. Design and Comparison of Lightweight Group Management Strategies in EnviroSuites. In *International Conference on Distributed Computing in Sensor Networks*, 2005.

[18] L. Luo, T. Abdelzaher, T. He, and J. A. Stankovic. EnviroSuite: An Environmentally Immersive Programming Framework for Sensor Networks. *ACM Transactions on Embedded Computing System, Special issue on Dynamically Adaptable Embedded Systems*, accepted.

[19] L. Luo and et. al. EnviroLog: Asynchronous Event Record and Replay Service for Wireless Sensor Network. In *in submission*.

[20] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. In *Operating Systems Design and Implementation*, December 2002.

[21] M.Maroti, B. Kusy, G. Simon, and A. Ledeczi. The Flooding Time Synchronization Protocol. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[22] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust Distributed Network Localization with Noise Range Measurements. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[23] J. Polastre and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[24] A. Savvides, C. C. Han, and M. B. Srivastava. Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors. In *Proc. of Mobile Computing and Networking (MOBICOM)*, 2001.

[25] R. Stoleru, T. He, and J. A. Stankovic. Walking GPS: A Practical Solution for Localization in Manually Deployed Wireless Sensor Networks. In *1st IEEE Workshop on Embedded Networked Sensors EmNetS-I*, October 2004.

[26] R. Szewczyk, A. Mainwaring, J. Anderson, and D. Culler. An Analysis of a Large Scale Habit Monitoring Application. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[27] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. In *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003.

[28] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A Wireless Sensor Network for Structural Monitoring. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[29] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom*, 2001.

[30] T. Yan, T. He, and J. Stankovic. Differentiated Surveillance Service for Sensor Networks. In *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003.

[31] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM*, 2002.

[32] G. Zhou, T. He, and J. A. Stankovic. Impact of Radio Irregularity on Wireless Sensor Networks. In *The Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2004.