

Sensor Support Systems for Asymmetric Threat Countermeasures

Chung-Ching Shen, *Student Member, IEEE*, Roni Kupershtok, Sanaz Adl, Shuvra S. Bhattacharyya, *Senior Member, IEEE*, Neil Goldsman, and Martin Peckerar, *Fellow, IEEE*

Abstract—In the past, primary focus has been given to novel sensor elements for deployment against urban terrorists and in limited force engagements. The issue explored in this paper is the adequacy of electronic system support for these new sensing elements. For example, ad hoc distributed networks must lie dormant for long periods of time and “come alive” when threats are nearby. This presents a unique challenge in the storage, generation, and management of power. In this paper, we demonstrate designs of processor algorithms and telecommunication protocols that alleviate current power-system shortcomings for these stationary networks. These advances include: 1) low-power protocols for data fusion and fault tolerance and 2) system-level energy modeling and analysis. As a concrete example, we define a distributed sensor support system for line crossing recognition. We demonstrate that threat detection is a system-level problem. Single elements of the system chain individually have small impact on overall performance. Through the development of a preamplifier/amplifier chain for optimum signal-to-noise (S/N) ratio, we show the degree to which system-level architecture can improve reliable detection. Specifically, the use of sensor redundancy to improve performance is analyzed from a statistical basis.

Index Terms—Distributed algorithms, low-power modeling, system-level developments, wireless sensor networks.

I. INTRODUCTION AND RELATED WORK

SENSOR support systems, such as wireless sensor networks (WSNs), address a great diversity of asymmetric defense and security applications. They include chem/bio threat detection, explosive detection, intrusive detection, and battlefield surveillance [1]. In many circumstances, sensor nodes are densely deployed in areas that are dangerous or, otherwise, inaccessible to humans. Thus, nodes must communicate with one another wirelessly through self-organizing protocols [2], [3]. Often, when designing such a distributed sensor system, the size of individual sensor nodes should be small enough so that they can easily be hidden in the environment. Issues of energy and power consumption are especially important due to the requirement of extended system lifetime [4]. A long

Manuscript received July 25, 2007; revised January 23, 2008. This work was supported in part by the Laboratory for Physical Sciences. The associate editor coordinating the review of this paper and approving it for publication was Dr. Janet Jensen.

C.-C. Shen, R. Kupershtok, S. S. Bhattacharyya, N. Goldsman, and M. Peckerar are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: ccshen@umd.edu; akuper@umd.edu; ssb@umd.edu, neil@umd.edu; peckerar@umd.edu).

S. Adl was with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA. She is now with Hughes Research Laboratories, Malibu, CA, USA (e-mail: sanaz.adl@gmail.com).

Digital Object Identifier 10.1109/JSEN.2008.922726

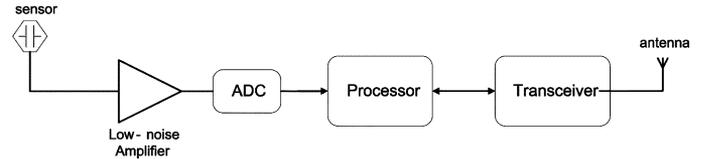


Fig. 1. Block diagram of a sensor support system platform.

autonomous system lifetime is an important evaluation metric for sensor support systems since any system is required to stay alive as long as possible. Also, fault tolerance features of the system are desired so that the system functionality can be reliably maintained.

Design of each element in a sensor support system is, in itself, a research topic. For example, Fig. 1 shows a typical block diagram for a system platform that consists of distinct single elements. In order to extend the system lifetime, low-power design of single elements for a sensor node platform has been extensively studied in recent years (e.g., see [5] and [6]). However, few of these design techniques for lifetime improvement have been discussed from a system-level point-of-view.

In this paper, we consider the design of sensor support systems for threat detection as a system-level problem. Here, the system-level problem is defined to be considered at the application level, protocol level, physical design level, and performance modeling level. For example, it includes algorithm streamlining, communication protocol configuration, and hardware/software implementation. We target system lifetime as the key system-level optimization objective and present our designs and experiments all aimed at lifetime improvement. This includes integration of a sensor support system for threat detection with single-element designs; system-level energy modeling and analysis; and simulation-based experimental results.

We start with the introduction of a distributed sensor network for line-crossing recognition as a threat detection application. This system is sensor supported with an acoustic sensor installed on each sensor node. Threats are detected via acoustic signals. The purpose of this system is to periodically reach consensus in deciding whether or not an object (“intruder”) has crossed a specific boundary (“line”) in a noisy environment that is continuously monitored. Furthermore, upon detecting an intrusion, the system determines where the line was crossed (i.e., between which nodes in the line). For example, in Fig. 2, the sensor nodes are placed in a circle inside a room. In this practical configuration, the system recognizes when and where a subject has crossed the circle through an integrated application, protocol, and system architecture development.

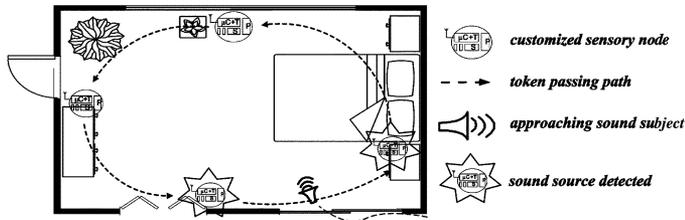


Fig. 2. An indoor environment scenario with the use of the threat detection system for line-crossing recognition.

When we consider lifetime improvement at the system level, the energy consumption of each sensor node in the system must be carefully optimized to increase system lifetime. Experimental results presented in [7] and [8] show that the power consumption for communication devices such as transceivers dominates overall power consumption on a sensor node. Therefore, for application-level algorithm development, we develop a lightweight distributed algorithm for line-crossing recognition so that the transceiver use time—in terms of the data size (decoded payload bits) to be communicated—can be minimized. Furthermore, for protocol-level consideration, all sensor nodes in such a threat detection application communicate with each other through an efficient, wireless time-division multiple-access (TDMA) protocol so that each node can transmit and receive at designated time slots, and can “sleep” during other times for energy savings. The packet routing path for such an application is based on a ring topology.

In [9], Hirschberg and Sinclair proved an upper bound of $O(\log N)$ on the number of bits that are sent by every node during a consensus task of N nodes arranged in a bidirectional ring topology. Every node that executes its algorithm has an initial input, and has no additional inputs during its execution. In [10], Dinitz *et al.* proved an upper bound of $O(N)$ on the number of bits that is sent by all nodes during a consensus task of N nodes arranged in a tree topology (a chain topology is a special case). This proof is based on the collection of information with feedback (CIF) algorithm. Every node that runs the CIF algorithm may have an initial input without additional inputs during its execution. After one round that includes two phases—the “collect” phase and “feedback” phase—all nodes reach consensus.

In our proposed distributed algorithm—in contrast to the approaches described above—each node can obtain many inputs (i.e., either from the received data or from the sensed data) during its execution, and based on these inputs, all of the nodes decide whether or not a subject is approaching and crossing the given line. Also, our algorithm has the property that either $O(\log C)$ or $O(\log N)$ data bits are needed depending on the protocol stage (i.e., synchronization stage or communication stage), instead of $O(N)$ bits. Furthermore, during most of its lifetime, our system communicates with only $O(\log C)$ data bits. Here, C —a design parameter—is the minimum number of nodes that must sense the subject in order to reach consensus that an intruder is approaching and crossing the line. Higher values of C provide higher system accuracy at the expense of higher communication requirements and higher recognition latency. Since energy consumption during transmission and re-

ception is high, our approach reduces energy consumption significantly by reducing the number bits that need to be communicated for overall system operation.

Any sensor support system currently envisioned monitors threats in a noisy environment. False detection is inevitable at some level. This, in turn, creates unnecessary energy consumption especially if the monitored environment is severely noisy. We propose a novel design for a low-power preamplifier/amplifier chain (or called amplifier concisely) with ultra high signal-to-noise (S/N) ratio for preventing such a false detection problem and reducing redundant energy consumption. Our amplifier is designed to use low-noise read-out circuitry, including a preamplifier and shaping amplifier, to increase the S/N ratio. The shaping amplifier is used to achieve two conflicting goals. The first goal is to increase the S/N ratio by restricting the bandwidth. A large bandwidth will increase the noise without increasing the signal. The pulse shaper takes a narrow pulse and turns it into a broader, gradually rounded peak. The second goal is to limit the pulse width in order to measure consecutive signal pulses without pileup or overlap. A tradeoff exists because reducing the signal pulse width will increase the signal rate but at the expense of higher noise. Optimum shaping depends on the desired application. In this case, the goal is to increase the S/N ratio in detector sensors. Therefore, the main focus will be on limiting the bandwidth to achieve a higher S/N ratio. The readout array results show that the S/N ratio is significantly improved compared with previous work [11] done in this field.

References [8], [12], and [13] are preliminary summaries of the communication protocol, energy modeling, and preamplifier design aspects of this work, respectively. This paper presents these works in more depth by analyzing system-level energy consumption and improvement, and also presents the integration of these works into a complete, operational line crossing recognition system, which forms the main contribution of this paper. By developing such a complete system and the corresponding analysis, we demonstrate that sensor design is a system-level problem, and we examine how key system-level design decisions affect overall performance.

The rest of this paper is organized as follows. Section II introduces a distributed sensor support system for line-crossing recognition as a threat detection application, as well as an associated communication protocol for low-power operation, data fusion, and fault tolerance. Section III shows our novel design for a low-power, low-noise preamplifier/amplifier chain for high capacitance sensors. Section IV analyzes the system lifetime for our integrated system through a proposed system-level energy modeling approach. Experimental results for the discussed sensor support system are shown in Section V. Finally, Section VI concludes this paper.

II. DISTRIBUTED SENSOR SYSTEM FOR LINE-CROSSING RECOGNITION

A. Lightweight Distributed Algorithm for Line-Crossing Recognition

Our proposed distributed system uses a TDMA-based communication protocol that consists of two stages: syn-

```

0 function receiveMode() {
1  ( $F_{RX}$ ,  $CM$ ,  $ID$ ,  $C_{global}$  Or  $M_{sense}$  Or  $MAX_{sense}$ ) = receiveOp(period  $T_{RX}$ );
2  if ( $F_{RX}$  is true) then
3    switch  $CM$  then
4      "phase 0, round 0":
5        update  $C_{glo}$  from the least 3 bits of the received message.
6        if ( $F_{sen}$  is true) then
7          if ( $F_{app}$  is false and  $C_{sen}$  is 0) then
8            increment  $C_{glo}$ ;
9            set  $F_{inc}$  to true;
10           increment  $C_{sen}$ ;
11
12          if ( $F_{app}$  is false) then
13            if ( $C_{glo}$  is larger than  $C_C$ ) then
14              set  $F_{app}$  to true;
15            else
16              if ( $F_{sen}$  is false and  $F_{inc}$  is true) then
17                decrement  $C_{glo}$ ;
18                set  $F_{inc}$  to false;
19
20            if ( $C_{glo}$  equals to 0) then
21              update  $CM$  to "phase 1 and round 0";
22              set  $M_{sen}$  to  $C_{sen}$ ;
23
24          "phase 1, round 0":
25            if ( $phase$  equals to 0) then
26              update  $CM$  to "phase 1 and round 0";
27              if (the least 6 bits of the received message is less than
28                 $C_{sen}$ ) then
29                set  $M_{sen}$  to  $C_{sen}$ ;
30              else
31                set  $M_{sen}$  to the least 6 bits of the received message;
32            else
33              update  $CM$  to "phase 1 and round 1";
34              set  $MAX_{sen}$  to the least 6 bits of the received message;
35
36          "phase 1, round 1":
37            set  $MAX_{sen}$  to the least 6 bits of the received message;
38            if ( $round$  equals to 0) then
39              update  $CM$  to "phase 1 and round 1";
40
41            if ( $MAX_{sen}$  equals to  $C_{sen}$ ) then
42              "the subject crossed the line near node  $I$ ";
43              reinitialize the system;
44 }

```

```

function transmitMode() {
  switch  $CM$  then
    "phase 0, round 0":
      update  $CM$  to "phase 0, round 0"
      transmitOp( $CM$ ,  $ID$ ,  $C_{glo}$ );
    "phase 1, round 1":
      update  $CM$  to "phase 0, round 0"
      transmitOp( $CM$ ,  $ID$ ,  $M_{sen}$ );
    "phase 1, round 1":
      update  $CM$  to "phase 0, round 0"
      transmitOp( $CM$ ,  $ID$ ,  $MAX_{sen}$ );
}

```

Notation for distributed line-crossing algorithm

N: number of nodes
 F_{RX} : local received flag
 F_{app} : global approaching flag
 F_{inc} : local incremented flag for C_{global}
 F_{sen} : local sensed flag
phase: phase indicator
round: round indicator
CM: control mode message
 C_{glo} : global sensed counter
 C_{sen} : local sensed counter
 C_C : global sensed threshold
 M_{sen} : local maximum sensed count
 MAX_{sen} : global maximum sensed count
 T_{RX} : maximum waiting time for receiving message

receiveOp: hardware-controlled receiving operations
transmitOp: hardware-controlled transmitting operations

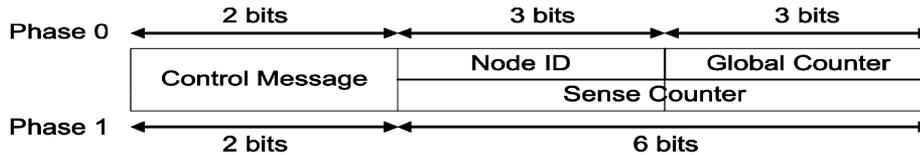


Fig. 3. Pseudocode specification for our distributed algorithm for line-crossing recognition and an illustration of data packet structure.

chronization and communication. The maximum number of nodes (upper bound) is determined according to the actual environment where the system needs to be deployed. Therefore, the number of nodes is fixed at design time. All node-to-node communications are based on a ring topology. Based on our experiments, we assume that at most two nodes may fail in the system. Define $(N + i - 1) \bmod N$ to be the neighbor of node $(N + i) \bmod N$. Assuming that there is no node failure, the only requirement is that every node "hears" (receives communication from) its neighbor. If there are at most two node failures (e.g., node $(N + i - 1) \bmod N$ and then $(N + i - 2) \bmod N$), the requirement is that node $(N + i) \bmod N$ hears nodes $(N + i - 1) \bmod N$, $(N + i - 2) \bmod N$, and then $(N + i - 3) \bmod N$, respectively. Therefore, under our assumption, it is not necessary that all nodes hear all other nodes. All N nodes within the system run the distributed algorithm, and reach a consensus based on local decisions of C nodes, while a subject is being detected ($C \leq N$). Here, C is a predetermined parameter that allows

the designer to control a tradeoff between recognition accuracy and communication requirements.

The full operation of the line-crossing recognition application, which runs on each node after the whole system is synchronized (i.e., at the communication stage), involves two phases of operation (phases 0 and 1). In phase 0, the nodes reach a consensus and decide whether the subject has crossed the line, and in phase 1, the nodes find the place where the subject crossed the line.

Fig. 3 shows a pseudocode representation of the proposed distributed algorithm, as well as an example to illustrate the message structure of the data packets used in the communication protocol. According to the use of different phases of operation, only the number of least significant bits is used in each phase. This is explained in our discussion of Fig. 3 later in this section. We implement such an algorithm at a node level in terms of mode operations (i.e., *Transmission* and *Reception*) in a TDMA-based protocol design. Note that in the algorithm operation, hardware-controlled receiving and transmitting opera-

tions are dependent on the targeted transceiver module. Since this paper is concerned primarily with system-level design, we do not address the details of these operations, which encapsulate lower level hardware configurations.

In phase 0 of the proposed algorithm (lines 4–22), the nodes use a counter C_{glo} to decide whether the subject is approaching based on local decisions of C nodes ($C \leq N$) that sense the subject. In its turn, each node receives C_{glo} from its previous neighbor (i.e., the node with ID i receives from node $(N + i - 1) \bmod N$, which is its “left” neighbor in the circular, virtual linkage of nodes based on their identifiers). When a node senses that the subject is approaching, it increments C_{glo} by one (line 8). Every node increments C_{glo} at most once. Therefore, at any moment during the first stage of phase 0, the value of C_{glo} indicates the number of nodes that sense the subject. When C_{glo} reaches C , all the nodes reach a consensus and decide that the subject is approaching. The node that increments C_{glo} to C is the first node to set its approaching flag (F_{app}) to 1 (line 14). In their respective turns, all of the other nodes set their F_{app} values to 1. Note that in this phase of operation, the execution of the algorithm relies on the count of sensing samples. The reliability of the sensing activities determines the system performance, which further affects the energy consumption for the overall system. Thus, a high S/N ratio device is preferred to help sensing elements improve performance. This will be discussed in more detail in the following sections.

Then, the second stage of phase 0 starts (line 15). During the second stage of phase 0, the subject is stepping away from the line. Every node has increased C_{glo} and stops sensing the subject, decrements C_{glo} by one (line 17). Since every node increments C_{glo} at most once, eventually all the nodes will stop sensing the subject. Then, C_{glo} will be decremented to zero. The first node, s , that decrements C_{glo} to zero, starts phase 1 (line 24).

In the message structure example shown in Fig. 3, the system is assumed to consist of at most eight nodes. Thus, in phase 0, the least significant 6 bits are used to represent a set of eight nodes using 3 bits, and to represent C_{glo} by using the other 3 bits. In phase 1, where it is not necessary to identify the nodes, the least significant 6 bits are used to represent C_{sen} .

In phase 1 of the proposed algorithm (lines 24–43), the nodes find the place where the subject crossed the line. This is done by finding which node sensed the subject the maximum number of times, which in turn is determined using a *sensing counter* C_{sen} in every node. Phase 1 consists of two rounds. In the first round (lines 24–34), the maximum sensed number is found. Node s transmits $C_{\text{sen}}(s)$ to its next neighbor. In its turn, each node i transmits the maximum of the number it received from node $(N + i - 1) \bmod N$ and $C_{\text{sen}}(i)$. At the end of the first round, node s receives the global maximum sensed count, $\text{MAX}_{\text{sen}} = \max\{C_{\text{sen}}(i)\}$. Afterward, node s starts the second round (line 36), where it transmits MAX_{sen} to its next neighbor. Every node j —in its turn—compares MAX_{sen} to $C_{\text{sen}}(j)$ (line 41). If there is a match, node j claims that it obtained the maximum number of senses, which means that the subject crossed the line near node j . Otherwise, node j transmits MAX_{sen} to its next neighbor. At the beginning of the second round, node s

holds MAX_{sen} , which is equal to $C_{\text{sen}}(k)$ ($0 \leq k \leq N - 1$). Therefore, the second round ends at node k . That is, the subject crossed the line near node k , and the system can be reinitialized. Suppose several nodes sense the approaching subject the same number of times which is equal to MAX_{sen} . In phase 1 round 0, the MAX_{sen} is found, and in phase 1 round 1, the first node out of these nodes that receives MAX_{sen} decides the approaching subject has crossed nearby.

From our algorithm, it can be observed that only $\log C$ are needed to represent C_{glo} and a maximum of $\log N$ bits are needed to represent C_{sen} . However, the time that the system operates in phase 1 is much less than the time of operation in phase 0. This is because the subject is stepping across the line continuously, and when phase 1 starts, the nodes can find the place where the subject crossed the line relatively quickly. Therefore, the critical number of data bits for transmitting and receiving is $O(\log C)$ instead of $O(\log N)$.

We determine the consensus threshold C based on noise in the actual environment. For example, if the environment is clear and there is minimal noise, we set $C = 1$. If the environment is noisy, we experiment with higher values of C so that multiple nodes must filter out noise, and agree about an approaching subject.

Based on a worse case analysis of the distributed algorithm, we give our fault tolerance protocol a chance of r rounds for every node before another neighbor is chosen. Suppose one round takes T time units and m nodes fail simultaneously. In this worse case, each node requires $T \cdot m \cdot r$ time units for a new neighbor to be found and assigned.

B. TDMA-Based Low-Power Protocol for Communication

TDMA-based communication protocols are often applied in small scale wireless communication systems [14], [15] due to their simplicity and low-power communication patterns. Furthermore, with TDMA, collision avoidance can be guaranteed throughout the system. Therefore, we have employed TDMA-based communication in the WSN system presented in this paper. The specific protocol that we have implemented consists of two stages: synchronization and communication. During the synchronization stage, the nodes are synchronized with each other. Whenever a node is powered on, it starts in the synchronization stage with a periodic communication pattern of transmitting and receiving one packet during each TDMA time frame, which consists of several predefined time lots. In each time frame, each powered node transmits one packet and receives at most one packet. Node i stays in the synchronization stage until it receives a packet from its previous node $(N + i - 1) \bmod N$. Afterward, it enters the communication stage for regularly communicating with other nodes in the network.

During the communication stage, every node transmits, receives, and idles periodically based on a predefined TDMA time schedule. In such a case, each node can power down its main computation and communication resources when the node is idle so that energy consumption is reduced to a minimal level. Fig. 4 shows a schedule for a four-node TDMA-based communication protocol with the ring topology routing scheme.

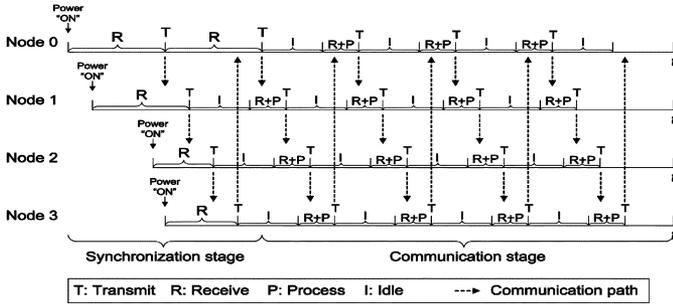


Fig. 4. TDMA-based communication pattern with a ring topology routing scheme for a four node example.

C. The Fault Tolerance Approach

We enhance the TDMA-based protocol mentioned above so that the fault tolerance feature is supported. This is preventing any node failures from translating into failures in the overall system. Here, a node failure means that a node stops processing and communicating, for example, due to lack of energy. Without considering fault tolerance, any node failure, in general, causes abnormal termination of all other nodes in the system.

The approach that we use in the synchronization stage is to have all of the nodes separately determine whether or not the system is synchronized, in addition to the initial node clock synchronization process discussed previously. The system is synchronized only if all of the functioning nodes are synchronized and agree on this situation. Once the system is synchronized, the algorithm of system failure prevention at the communication stage can be activated.

During the synchronization stage, there are only functioning (powered-on) nodes and powered-off nodes. Moreover, powered-on nodes will be synchronized and will enter the communication stage in a relatively short period of time. Our algorithm for system failure prevention is not incorporated in the synchronization stage. This is because a node might consider a powered-off neighbor as a failed node. Whenever a node fails in the communication stage, the distributed system will be reorganized automatically within the time period of a single TDMA frame.

The fault tolerance algorithm that we employ in the synchronization stage operates in the following way. Suppose that there are N nodes in the system, and every node has a unique identifier (ID) i such that $0 \leq i \leq N - 1$. Initially, once node i is turned on, it periodically transmits a packet to node $(i + 1) \bmod N$, every s seconds. As mentioned previously, such a packet includes a control message field, an id field, and a global counter field (C_{glo}) with an initial value of 1. In the synchronization stage, C_{glo} counts the number of nodes that are on and are synchronized. In addition, node i keeps an internal synchronization flag (F_{sync}), which is set to zero when the node is powered-on. F_{sync} indicates that node i knows whether all the nodes in the system are synchronized. Note that during the synchronization stage, some nodes might be on while others might be off. Therefore, the overall distributed system is not necessarily synchronized at this time.

While node i transmits a packet every s seconds, where s denotes the duration of a TDMA time frame, it continuously

tries to receive a packet from node $(N + i - 1) \bmod N$ (i.e., from its “left” neighbor in the circular, virtual linkage of nodes based on their identifiers). Whenever node i receives a packet from node $(N + i - 1) \bmod N$ with $C_{glo} \leq N - 1$, it reads the associated value C_{glo} , increments this value by 1, and transmits it within a data packet to node $(i + 1) \bmod N$.

Whenever a node j ($0 \leq j \leq N - 1$) is the first to receive a packet from its predecessor (i.e., from node $(N + i - 1) \bmod N$) with $C_{glo} = N - 1$, node j sets its internal F_{sync} to 1. Then, node j transmits a packet to node $(j + 1) \bmod N$ with a unique control message α , and C_{glo} is set to 0. That is, at this point, node j knows that all the nodes are on and that they are synchronized with their neighbors. Therefore, node j starts the process of informing all other nodes in the system that all the nodes are on and are synchronized. It does this by transmitting the control message α to its neighbor.

Now, suppose that a node k ($0 \leq k \leq N - 1$) and $k \neq j$) whose F_{sync} value is 0 receives a packet from node $(N + k - 1) \bmod N$ with the control message α . Then, node k sets its F_{sync} value to 1, and transmits a packet to node $(k + 1) \bmod N$ with the control message α , and with a C_{glo} value of 0. When node j (i.e., the first node that set its F_{sync} value to 1) receives a packet from node $(N + j - 1) \bmod N$ with the control message α , node j knows that all the nodes in the system have received α . Then, node j starts the communication stage by transmitting a packet to node $(j + 1) \bmod N$ with the control message β and with a C_{glo} value of zero. At this point, all the nodes are synchronized in the system.

Note that α is transmitted as long as there are still nodes that are not synchronized, and β is transmitted once all the nodes are on and are synchronized. Moreover, the time period over which the whole system remains in the synchronization stage must be larger than the time difference between when the first and the last nodes join the system plus an additional s seconds.

If our TDMA-based distributed system consists of N nodes, the TDMA time frame of s seconds is divided into N time slots, and each such slot lasts for a period of s/N seconds. During slot i ($0 \leq i \leq N - 1$), node i transmits a packet, and node $(i + 1) \bmod N$ receives that packet in its receiving window. Here, the receiving window is defined as the longest time period allowed for receiving packets within a given TDMA time slot.

When node i does not receive packets from node $(N + i - 1) \bmod N$ (i.e., if node $(N + i - 1) \bmod N$ has failed), then node i starts a process to find a new neighbor. Based on the predefined TDMA schedule, node i shifts its receiving window from slot $(N + i - 1) \bmod N$ to slot $(N + i - 2) \bmod N$ and tries to receive a packet from node $(N + i - 2) \bmod N$ during the next TDMA time frame. If node i succeeds in receiving a packet from some node $(N + i - \delta) \bmod N$ ($0 \leq \delta \leq N - 2$), then δ becomes the new neighbor of i in the new, fault-adapted, virtual linkage structure of the remaining $N - \delta$ functional nodes. All the nodes between nodes i and $(N + i - \delta) \bmod N$ (noninclusive) are considered from this point onward as being nonfunctioning nodes. That is, the corresponding TDMA time schedule on each node is predefined initially, and is adapted as execution evolves based on changes in system status.

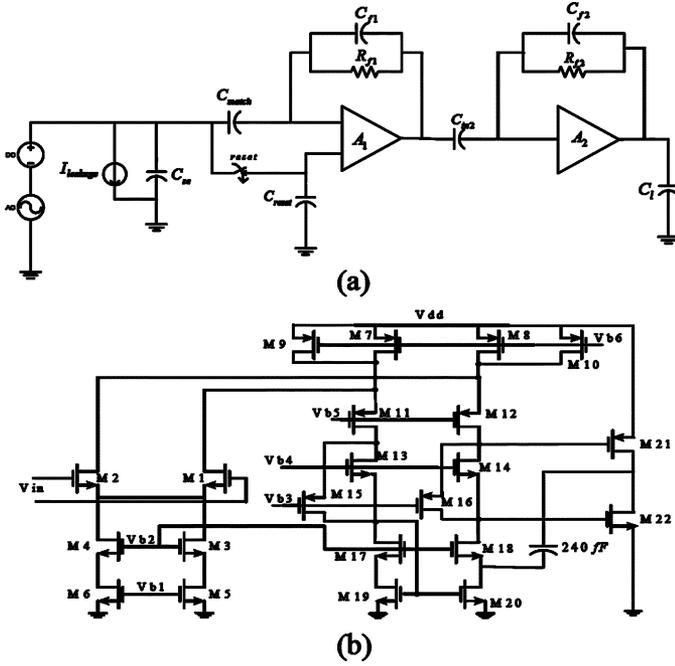


Fig. 5. (a) Preamplifier/amplifier chain block diagram. (b) Folded cascode amplifiers schematic for each stage.

In our proposed system, acoustic sensors will require some amplifier and preprocessing. Optimizing the analog preprocessing saves power, as shown in the next section.

III. LOW-NOISE PREAMPLIFIER/AMPLIFIER CHAIN FOR HIGH CAPACITANCE SENSORS

A low-noise amplifier/preamplifier chain is proposed for high capacitance sensors. The large sensor capacitance increases the charge uncertainty on the sensor node after reset. Due to the fact that a large capacitance enhances the reset noise component of the system's S/N ratio, innovative techniques need to be incorporated in readout arrays to increase the S/N ratio. Fig. 5(a) portrays the block diagram of the proposed readout array for high capacitance sensors. The readout circuitry consists of a preamplifier and amplifier. Each stage is a folded cascode with a class AB push-pull output stage, as shown in Fig. 5(b). A large capacitor along with a leakage current source is utilized for electronic simulation of the detector. The detector capacitance is simulated by a 100 nF capacitor. The capacitor used for simulating the detector has a capacitance that is five orders of magnitude more than that of capacitors used in previous techniques. The sensed signal is the voltage stored on the sensor capacitance.

Capacitive matching is also used at the amplifier input to reduce the reset noise at the sensor node. This method calls for an additional capacitor to be placed at the input of the preamplifier/amplifier chain. The capacitance of this capacitor must be equivalent to the capacitance seen at the preamplifier input. The capacitance seen at the input of the preamplifier is the feedback capacitor, C_{f1} , multiplied by the open loop gain of the preamplifier, A_1 . The capacitor referred to as C_{match} in Fig. 5(a) is used at the amplifier input for the purpose of matching, in order to increase the S/N ratio at the input of the preamplifier/amplifier chain. The matching capacitor is chosen to be equal to

TABLE I
NOTATIONS AND EXPERIMENTAL VALUES FOR SYSTEM-LEVEL ENERGY MODELING

Symbols	Description	Values
V_{cc}	system supply voltage	3.3V
R	transceiver data rate	500Kbps
f_{clk}	processor clock frequency	26MHz
M	data length	8bits
N_{ck}	number of cycles for processing	320(RX), 105(TX)
N_{pt}	number of sensed samples	1 – 5000
t_{sens}	sensing time per sample	1 μ s
t_s	TDMA slot time	50ms
t_p	CPU processing time	4 μ s(RX), 12.3 μ s(TX)
$t_{tran-st}$	transceiver startup time	200 μ s
$t_{tran-on}$	transceiver turn-on time	0.22ms
t_{ADC-c}	ADC conversion time/sample	58.4 μ s
t_{ADC-on}	ADC turn-on time	58.4 N_{pt} μ s
t_{tmr-on}	timer turn-on time	50ms

the product of C_{f1} and A_1 . By using capacitive matching, we are following the principle of impedance matching, which maximizes the signal power transmission into the amplifier system.

IV. ENERGY MODELING AND LIFETIME ANALYSIS

Based on the design schematic of a sensor support system platform shown in Fig. 1 and the designed scheme of TDMA operations in the network, we integrate various fine-grained energy models to evaluate energy consumption for the whole system. Moreover, the system-level energy model is used to evaluate the system lifetime in the experiments and to demonstrate how single elements of the system chain impact overall performance.

A. System-Level Energy Modeling

The proposed energy model estimates energy consumption from a system-level point of view, i.e., considering the energy consumption from the use of all system devices (including software and hardware components) within the framework of our TDMA-based protocol control scheme. Therefore, the system energy model for which the system is running at various protocol modes can be generalized to

$$E_m = E_{sens} + E_{amp} + E_{ADC} + E_{mcu} + E_{tran} \quad (1)$$

where E_{sens} , E_{amp} , E_{ADC} , E_{mcu} , and E_{tran} , respectively, denote the energy consumed by the sensor, preamplifier/amplifier chain, analog-to-digital converter, microcontroller, and transceiver when the system runs in different protocol modes—i.e., transmission, reception, and idle modes. Table I lists all notations with their descriptions for system-level energy modeling, as well as experimental values that are being used in our tests, where $t_{ADC-on} = N_{pt} \cdot t_{ADC-c}$, $t_p = N_{clk}/f_{clk}$, and $t_{tran-on} = t_{tran-st} + M/R$.

Specifically, when we consider the time to use various devices on a sensor node platform, the energy consumption for which

the system is in either transmission, reception, or idle mode is modeled as

$$E_m = N_{pt} \cdot [(P_{\text{sens}} + P_{\text{amp}})t_{\text{sens}} + P_{\text{ADC}}t_{\text{ADC-on}}] + P_{\text{tmr}}t_{\text{tmr-on}} + P_{\text{proc}}t_p + P_{\text{tran}}t_{\text{tran-on}} + P_{lp}t_{lp}. \quad (2)$$

Here, $N_{pt} \cdot [(P_{\text{sens}} + P_{\text{amp}})t_{\text{sens}} + P_{\text{ADC}}t_{\text{ADC-on}}]$ is considered as the energy consumption for all sensing elements. $P_{lp}t_{lp}$ denotes the energy consumption for the processor when the processor stays in its low-power state. For example, in transmission (reception) mode, the platform will turn to a low-power state after finishing its processing and communicating tasks, and therefore $t_{lp} = t_s - (t_{\text{tran-on}} + t_p)$. However, in idle mode, each sensor node platform stays in a low-power state during the whole time slot without executing processing and communicating tasks. That is, $t_p = t_{\text{tran-on}} = 0$ and $t_{lp} = t_s$ when $E_m = E_{\text{idle}}$. We define the sensing time, t_{sens} , as the time that a sensor and a sensor support amplifier complete a signal processing operation for one sample. For example, our designed amplifier takes approximately $1 \mu\text{s}$ to complete such a signal processing operation for one sample in terms of its RC time constant. Note that the timer device is turned on all the time due to the execution of runtime TDMA protocol control and synchronization, and therefore $t_{\text{tmr-on}} = t_s$.

B. System Lifetime Analysis

We analyze the average system lifetime according to the system-level energy models described above. The system lifetime is represented in (3), where the lifetime analysis is derived based on the prescheduled TDMA-based communication operations along with the characteristics of hardware and software components on each sensor node. According to the communication protocol discussed in Section II along with fault tolerance, all operations within a TDMA frame (T_{fr}) are executed periodically, frame-by-frame, until the system fails (i.e., all nodes run out of energy). Thus, in our lifetime model, we first estimate the average energy consumption for each sensor node within a single communication time frame, and then based on this estimation, the average lifetime of the overall system (i.e., T_{sys}) can be estimated in terms of total energy stored in the available batteries for each sensor node

$$T_{\text{sys}} = \frac{1}{N} \cdot \sum_{n=1}^N \frac{E_{\text{bat}}(n)}{P_{fr}(n)}$$

and

$$P_{fr} = \frac{1}{t_s} \left[N_{\text{tx}} \sum_{m \in \text{tx}} E_m + N_{\text{rx}} \sum_{m \in \text{rx}} E_m + N_{\text{idle}} \sum_{m \in \text{idle}} E_m \right] \quad (3)$$

where N denotes the number of sensor nodes in the system, $E_{\text{bat}}(n)$ denotes the total energy stored in a given battery, and P_{fr} denotes the power consumption for each TDMA time frame for node n , respectively; N_{tx} , N_{rx} , and N_{idle} denote the number of transmission (tx), reception (rx), and idle (idle) mode occurrences in a T_{fr} , respectively.

The energy model shown in this section can be used to estimate energy consumption in both the synchronization stage and the normal communication stage. However, the time of the synchronization stage is relatively short (e.g., $t_{\text{sens}} = t_{\text{ADC-on}} = 0$, and $t_{\text{tmr-on}}$, t_p , $t_{\text{tran-on}}$ are active at most once per T_{fr}) compared with the time spent for the normal communication, so the energy consumption during this stage can be omitted with minimal loss in accuracy. Our experiments corresponding to the use of system-level energy modeling outlined in this section will be demonstrated in Section V.

C. Fidelity Analysis

We calculate the fidelity of the system-level energy model based on the results of simulated versus measured energy consumption. Our reason for using the metric of fidelity here is that the design of sensor network systems depends on various cross-layer configurations, and therefore, it is difficult to obtain perfect accuracy of simulated energy for the purposes of fast energy consumption evaluation and associated design space exploration considerations. For this estimation, we use the fidelity metric to determine the trend of energy estimation based on our energy model as an alternative to physical measurement. In this way, the proposed energy model can be shown, in a quantitative way, to have high accuracy when compared with actual measured results. Thus, the model can be applied with high confidence to evaluate our sensor support system—and in particular, to compare alternative system configurations—in terms of lifetime for arbitrary network size. Here, by network size, we mean the number of sensor nodes in the network.

To measure the accuracy of our approach to system-level energy modeling, we use the estimation fidelity metric defined by

$$\text{fidelity} = \frac{2}{n(n-1)} \left(\sum_{i=1}^{n-1} \sum_{j=i+1}^n f_{ij} \right) \quad (4)$$

where $f_{ij} = 1$ if $\text{sign}(M_i - M_j) = \text{sign}(S_i - S_j)$, and $f_{ij} = 0$, otherwise. Here, $M_i(M_j)$ denotes the measured results and $S_i(S_j)$ denotes the simulated results, respectively.

For experimenting with fidelity calculation for the proposed system-level energy modeling approach, we generated 20 testing points for configuring the sensor node platforms with different combinations of supply voltages. The supply voltages across the different platforms in our experiments could be homogeneous or heterogeneous. For each configuration of supply voltages, we observed the energy consumption variation (simulated results from the energy model versus measured results) on the nodes corresponding to the voltage changes. Our fidelity experiments are summarized in Fig. 6, where the fidelity value is determined to be 0.91. Note that a fidelity value of 1 corresponds to perfect fidelity.

V. EXPERIMENTS

A. Experimental Setup

Demonstration of the discussed sensor support system for line-crossing recognition is presented in [12]. Here, a highly customized printed circuit board (PCB) is designed to support individual sensor nodes. All required components (as shown in

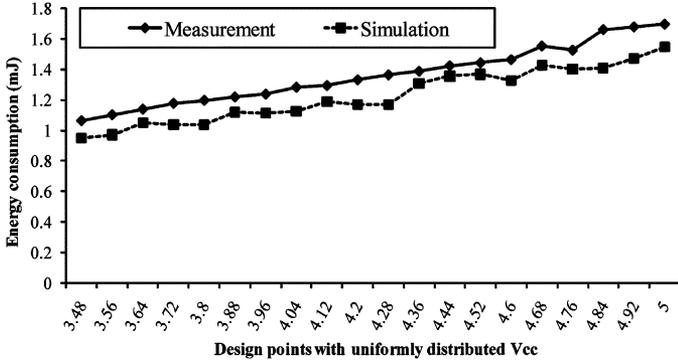


Fig. 6. Fidelity analysis for energy modeling.

Fig. 1) including an acoustic sensor are integrated on this sensor support system platform. The platform executes the proposed algorithm for line-crossing recognition. Moreover, the energy consumption results for fidelity analysis are measured from such a sensor node platform.

Based on the design of the preamplifier/amplifier chain, we construct a simulation-based experiment for estimating the system lifetime. An arbitrary number of sensor nodes in the system can be chosen for this simulation-based experimental environment. According to the fidelity analysis in the previous section, the analyzed results from this simulation-based experiment are similar to the results that we observed when we implemented the system in actual hardware with the same configurations.

For the experimental configurations in this paper, we set up ten sensor nodes for the integrated sensor support system such that each sensor node takes 0.5 s to pass around information across the whole system. That is, in this configuration, we set $T_{fr} = 0.5$ s, $N = 10$, $t_{slot} = 50$ ms, $N_{tx} = N_{Tx} = 1$, and $N_{idle} = 8$. Based on these system-level configurations, we conduct simulations for estimating the system lifetime of the integrated sensor support system in the following context.

B. Simulation Results for Preamplifier/Amplifier Chain

We first present the simulated performance results for our designed preamplifier/amplifier chain. These results were presented originally in [13]. The preamplifier/amplifier chain is designed in 0.13 μ m CMOS8RF IBM technology using the Cadence design package (Assura). The circuit has a gain of 63 dB. The frequency response is shown in Fig. 7(a).

The input to the preamplifier/amplifier chain from the capacitive sensor is simulated by a pulse with a pulse width of 0.5 μ s and amplitude of 1 mV. The pulse contains a DC offset of 1.8 V to provide the necessary input DC bias voltage for the signal processing chain. The equivalent output noise of the signal processing chain is shown in Fig. 7(b). The transient input and output waveforms for the preamplifier/amplifier chain are shown in Fig. 8. The amplifier has a maximum equivalent output noise of 280 μ V/ $\sqrt{\text{Hz}}$. The S/N ratio for the preamplifier/amplifier chain is about 5300 V/V. This value is 80 times larger than the S/N ratio of previously reported signal processing chains [13].

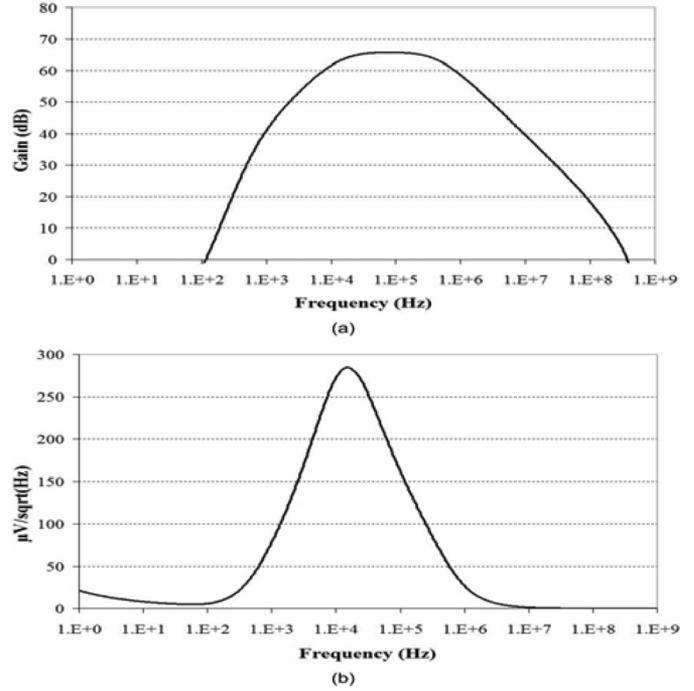


Fig. 7. (a) Frequency response of the preamplifier/amplifier using correlated double sampling and capacitive matching. (b) Equivalent output noise of the preamplifier/amplifier using correlated double sampling and capacitive matching.

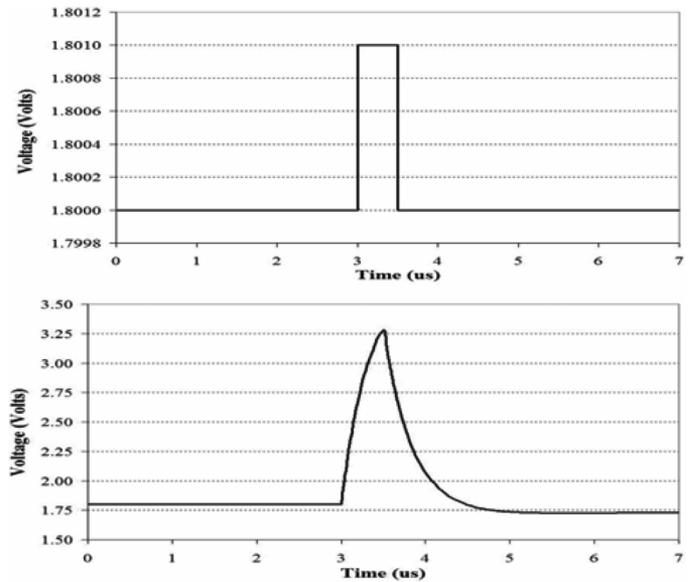


Fig. 8. Input and output transient waveforms.

C. Simulation Results for System Lifetime Analysis

In our experiments for lifetime analysis, we consider the case where all sensor nodes in the system execute tasks for line-crossing recognition frame-by-frame periodically without additional power-off mechanisms. As a practical example, Fig. 9(a) first presents power consumption comparison between each single element used in the sensor node platform, where the electrical specifications except for the designed amplifier are obtained from [16] and [17]. Note that in [17], we only consider

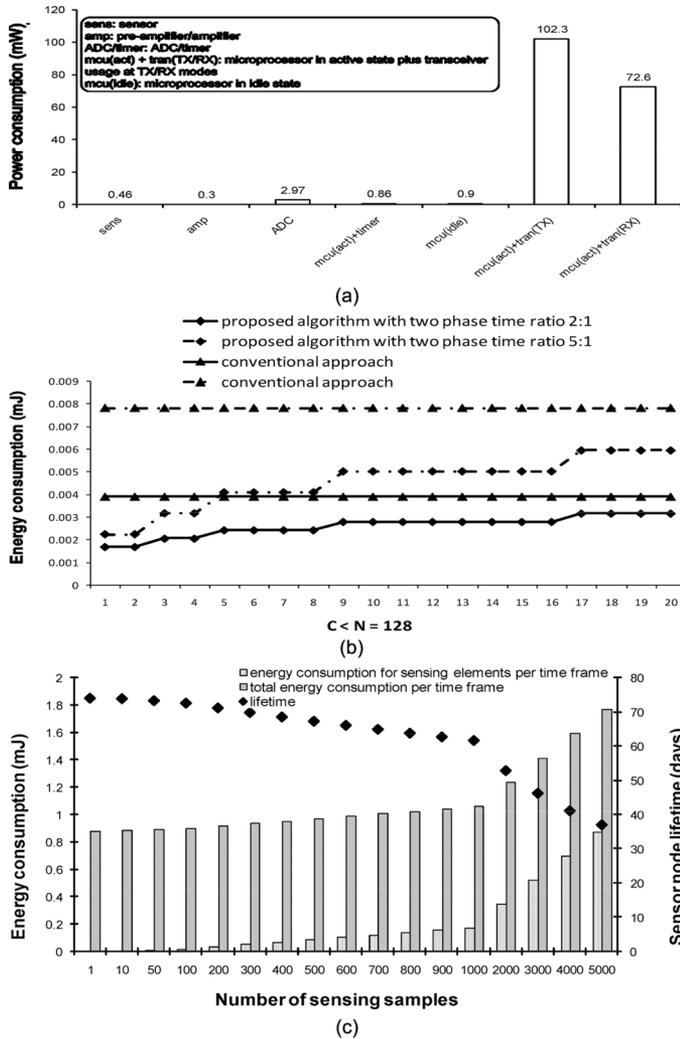


Fig. 9. (a) Power consumption comparison among all devices on a sensor node platform. (b) Energy consumption comparison depending on C . (c) Energy consumption and lifetime comparison, where simulated results are calculated in terms of the specifications in [13], [16], [17].

the power consumption of the sensor part for our experiments according to its provided gain information. In Fig. 9(a), it shows explicitly that the communication element (i.e., the transceiver) dominates overall power consumption on a sensor node platform. Therefore, we need to consider the lifetime problem at the system level so that designs of single elements become appropriately tailored for integration into the overall system. Fig. 9(b) shows how energy depends on the number C of nodes needed to reach consensus, where N is assumed to be 128. In Fig. 9(b), we use two experiments to compare energy consumption for data communication between our proposed distributed algorithm with two-phase operation and a conventional approach with single-phase operation, where the time ratios of the proposed algorithm spent in phase 0 and phase 1 are assumed to be 2:1 and 5:1, respectively. We observe from Fig. 9(b) that the longer the duration of system operation, the more the potential for energy savings by the proposed algorithm (compared to the single-phase approach) for a smaller value of C . Note that the data to be communicated in a conventional approach only depends on N and is independent of C .

To consider the lifetime problem at the system level, in Fig. 9(c), we first demonstrate a comparison between the total energy consumption and the energy consumption for sensing elements on a node platform. This comparison is done based on energy consumption values during a single TDMA time frame. With this system-level consideration, we reduce the transceiver use time by minimizing the data that is required to be communicated across sensor nodes. This minimization is based on our proposed algorithm for the targeted threat detection application. Moreover, a TDMA-based communication protocol is used to schedule the tasks among processing, communicating, and idling so that the processing and communicating elements can remain in idle states when their tasks are complete.

As the statistical results show in Fig. 9(c), once the number of required sensing samples increases, the energy consumption for sensing elements increases and significantly impacts overall energy consumption for the system. When using a traditional signal processing chain with lower S/N ratio for processing sensing signals, multiple sensing activities are required under different circumstances with different noise levels, because noise interference is not predictable during the system runtime. Many false sensing detections may occur when producing samples from the received signals. As a result, it is difficult to formulate a general, exact solution for the number of required sensing samples.

Our new amplifier with high S/N ratio solves this problem when it is integrated into the system. Regardless of interference at any noise level, the noise will be canceled through the correlated double sampling technique in our design. Therefore, only one sensing task is required for producing each sample from the received signal.

We also demonstrate the system lifetime comparison in Fig. 9(c) in terms of the number of sensing samples required in our system. In this lifetime experiment, a 3.3 V · 950 mAh capacity lithium battery is provided in each sensor node platform so that a practical result in terms of days of useful operation is shown in the figure. From the lifetime results, we observe that by using our designed amplifier, the energy consumption for sensing elements is not critical anymore because there is no redundant sensing activity executed for maintaining the system functionality, and one sensing activity only consumes 0.02% of the overall energy consumption within a TDMA time frame. In this case, our system can last 75 days since all sensor nodes are powered-on.

However, without using our optimized amplifier, the system lifetime drops since we require too many undesirable sensing activities. For example, in the worse case statistical results, the system lifetime will drop to approximately 37 days when 5000 sensing samples within a TDMA time frame are required. In this case, the energy consumption for sensing elements may have equivalent influence to computation and communication elements on overall energy consumption—for example, 49.2% of overall energy consumption is consumed by the sensing elements in our system when a conventional amplifier is used. Note that in many practical applications where tamper-resistant deployment is important, the size of each sensor node has to be limited so that the nodes cannot be easily found or manipulated. In such cases, batteries with very limited capacity must be used,

and the system lifetime may drop significantly (e.g., well below a single day) if redundant sensing activities cannot be avoided.

VI. CONCLUSION

In this paper, we have emphasized that threat detection should be considered as a system-level problem. Single elements of the system chain individually have small impact on overall performance. We have presented a complete system design methodology for a practical application of distributed line-crossing recognition, including algorithm streamlining, communication protocol configuration, hardware/software implementation, and lifetime modeling. Our proposed distributed algorithm is useful in reducing the amount of data that must be communicated across nodes in the network. Furthermore, the communication protocol that we employ carefully manages the duty cycle to achieve further improvements in energy efficiency. Our designed protocol is fault tolerant so that node failures are prevented from translating into failures in the overall system; the capability to add nodes dynamically into our distributed system is being considered in our ongoing and future work.

Our design of the preamplifier/amplifier signal processing chain includes the important feature of ultra high S/N ratio. Based on a series of simulation-based experiments, and our analysis of the results of these experiments, we employ various design techniques to improve the system lifetime. From the experimental results, we observe the importance of considering the system lifetime problem at the system level. We also observe that full system integration for sensor support systems plays an important role in influencing the design for individual components (“elements”) that are to be used on sensor node platforms.

REFERENCES

- [1] M. Kuorilehto, M. Hannikainen, and T. D. Hamalainen, “A survey of application distribution in wireless sensor networks,” *EURASIP J. Wireless Commun. Netw.*, pp. 774–788, 2005.
- [2] M. Kohvakka, M. Hannikainen, and T. Hamalainen, “Wireless sensor prototype platform,” in *Proc. IEEE Int. Conf. Industrial Electron., Control, Instrumentation*, Nov. 2003, pp. 1499–1504.
- [3] J. Hill and D. Culler, “Mica: A wireless platform for deeply embedded networks,” *IEEE Micro*, pp. 12–24, Nov. 2002.
- [4] K. Romer and F. Mattern, “The design space of wireless sensor networks,” *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 54–61, Dec. 2004.
- [5] K. Bhagavathula, A. H. Titus, and C. S. Mullin, “An extremely low-power CMOS glare sensor,” *IEEE Sensors J.*, vol. 7, pp. 1145–1151, Aug. 2007.
- [6] P. D. Dimitropoulos, D. P. Karampatzakis, G. D. Panagopoulos, and G. I. Stamoulis, “A low-power/low-noise readout circuit for integrated capacitive sensors,” *IEEE Sensors J.*, vol. 6, pp. 755–769, Jun. 2006.
- [7] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin, “Networking issues in wireless sensor networks,” *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 799–814, Jul. 2004.
- [8] C. Shen, C. Badr, K. Kordari, S. S. Bhattacharyya, G. L. Blankenship, and N. Goldsman, “A rapid prototyping methodology for application-specific sensor networks,” in *Proc. IEEE Int. Workshop on Comput. Architecture for Machine Perception and Sensing*, Montreal, ON, Canada, Sep. 2006.
- [9] D. S. Hirschberg and J. Sinclair, “Decentralized extrema-finding in circular configurations of processors,” *Commun. ACM*, vol. 23, no. 11, pp. 627–628, 1980.
- [10] Y. Dinitz, S. Moran, and S. Rajsbaum, “Exact communication costs for consensus and leader in a tree,” *J. Discrete Algorithms*, vol. 1, no. 2, pp. 167–183, Apr. 2003.
- [11] W. R. Cook, J. A. Burnham, and F. A. Harrison, “Low-noise custom VLSI for CdZnTe pixel detectors,” in *Proc. SPIE*, Nov. 1998, vol. 3445, pp. 347–354.

- [12] C. Shen, R. Kupershtok, B. Yang, F. M. Vanin, X. Shao, D. Sheth, N. Goldsman, Q. Balzano, and S. S. Bhattacharyya, “Compact, low power wireless sensor network system for line crossing recognition,” in *Proc. Int. Symp. Circuits Syst.*, New Orleans, LA, May 2007, pp. 2506–2509.
- [13] S. Adl and M. Peckerar, “A low-noise pre-amplifier/amplifier chain for high capacitance sensors,” in *Proc. IEEE Sensors Appl. Symp.*, San Diego, CA, Feb. 2007.
- [14] T. van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in *Proc. 1st Int. Conf. Embedded Networked Sensor Syst.*, Nov. 2003, pp. 171–180.
- [15] W. Ye, J. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” *IEEE/ACM Trans. Networking*, pp. 493–506, Jun. 2004.
- [16] CC1110 Data Sheet. SWRS033A, Texas Instruments, 2006.
- [17] SP0103 Microphone Sensor, Knowles Acoustics, 2003.



Chung-Ching Shen (S’08) received the B.S. degree in computer and information science from National Chiao Tung University, Hsin-Chu, Taiwan, in 2001 and the M.S. degree in electrical and computer engineering from the University of Maryland, College Park, in 2004. He is currently working towards the Ph.D. degree in electrical and computer engineering at the University of Maryland, College Park.

His research interests include energy-driven code-sign for distributed embedded systems, signal processing, and low-power VLSI design of embedded systems for wireless sensor network applications.



Roni Kupershtok received the B.A. degree in mathematics from the University of Haifa, Haifa, Israel, and the M.S. and Ph.D. degrees from the Technion—Israel Institute of Technology, Haifa.

In 2002, he joined the group of Prof. U. Vishkin at the University of Maryland, College Park, and led the development of a compiler and a simulator for the XMT parallel architecture. He is a Postdoctoral Researcher in the group of Prof. N. Goldsman at the University of Maryland, College Park, and developed an efficient distributed algorithm that minimizes the

total energy consumption used by a set of independent nodes that communicate one with each other in a unidirectional ring topology.



Sanaz Adl received the B.S. degree in electrical and computer engineering from the Sharif University of Technology, Tehran, Iran, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 2003 and 2007, respectively.

She is a Research Staff Member with Hughes Research Laboratories, Malibu, CA. Her research interests include analog and mixed-mode circuit design and low-noise and low-temperature signal processing chains.



Shuvra S. Bhattacharyya (S’92–M’95–SM’01) received the B.S. degree from the University of Wisconsin, Madison, in 1987 and the Ph.D. degree from the University of California, Berkeley, in 1994.

He is a Professor in the Department of Electrical and Computer Engineering, and at the Institute for Advanced Computer Studies (UMIACS), University of Maryland, College Park. He is also an Affiliate Associate Professor in the Department of Computer Science. He is coauthor of two books and the author or coauthor of more than 60 refereed technical articles.

His research interests include signal processing, embedded software, and hardware/software codesign. He has held industrial positions as a Researcher at the Hitachi America Semiconductor Research Laboratory, San Jose, CA, and as a Compiler Developer at Kuck & Associates, Champaign, IL.



Neil Goldsman received the Ph.D. degree in electrical engineering from Cornell University, Ithaca, NY, in 1989.

He is a Professor in the Electrical and Computer Engineering Department, University of Maryland, College Park. He is an originator of the Legendre Polynomial/Spherical Harmonic Boltzmann approach to device simulation. Recently, he has focused on mixed-signal VLSI, radio-frequency CMOS circuit design, wide-bandgap semiconductors, and nanoscale devices, including carbon

nanotubes. He directs the Mixed-Signal VLSI Design Laboratory and the Semiconductor Device Simulation Laboratory at the University of Maryland. His main research interests are in the areas of device, material and circuit modeling and design. He has written two educational texts in electronics used at the University of Maryland, and has published more than 100 technical papers.

Dr. Goldsman is the recipient of the National Science Foundation's Research Initiation Award, the University of Maryland IEEE Professor of the Year Award, the George Corcoran Award for Contributions to Education, and the IEEE Benjamin Dasher Award.



Martin Peckerar (F'93) received the B.S. degree from Stony Brook University, Stony Brook, NY, and the M.S. and Ph.D. degrees from the University of Maryland, College Park.

In 1976, he joined the Westinghouse Advanced Technology Laboratory, where he led a group working in advanced MOS process development. At Westinghouse, he developed the deep-depletion CCD for X-ray and for IR imaging. In 1981, he became head of the Nanoelectronics Processing Facility at the Naval Research Laboratory (NRL) and,

subsequently, head of the Surface and Interface Sciences Branch. He is now a Professor of Microelectronic Engineering at the University of Maryland. He is an expert in the area of imaging system technology. Among his achievements in this area is the development of a maximum-entropy image reconstruction chip. This chip represented the first application of entropy regularization to integrated circuit signal processing. He has developed fast-Fourier chips based on neural net principles and a tomographic imager chip (also based on maximum entropy principles). He is coauthor of the textbook *Electronic Materials: Science and Technology*, editor of the book *Synthetic Microstructures in Biological Research*, and coauthor of the book *The Korean Electronics Industry*.