In Proceedings of the International Symposium on Circuits and Systems, pages 2506-2509, New Orleans, Louisiana, May 2007

Compact, Low Power Wireless Sensor Network System for Line Crossing Recognition

Chung-Ching Shen, Roni Kupershtok, Bo Yang, Felice Maria Vanin, Xi Shao, Datta Sheth,
Neil Goldsman, Quirino Balzano, and Shuvra S. Bhattacharyya
Department of Electrical and Computer Engineering
University of Maryland, College Park 20742
{ccshen, akuper, boyang, felvanin, xshcn, sheth, neil, qbalzano, ssb}@umd.edu

***Abstract*— Many application-specific wireless sensor network (WSN) systems require small size and low power features due to their limited resources, and their use in distributed, wireless environments. In this paper, we present a light-weight distributed algorithm for line-crossing recognition, together with its analysis, implementation, and experimental evaluation within a prototype wireless sensor network platform. The algorithm is developed in conjunction with a TDMA-based communication protocol such that the proposed system provides for low duty cycle and energy efficient operation. An accurate lifetime model is proposed with consideration of detailed energy usage to analyze and estimate the system lifetime. Our experimental results demonstrate the accuracy of this lifetime model, and its utility in optimizing network implementation. The design and experimental evaluation of our prototype network demonstrates the compactness and functionality of the proposed distributed WSN system for line-crossing recognition.**

I. Introduction and Related Work

While considering the design of a distributed WSN system applied in an indoor environment, we propose a light-weight, distributed algorithm for line-crossing recognition. We implement this algorithm as part of an operational wireless sensor network, and develop a highly customized and streamlined printed circuit board (PCB) design to support the individual sensor nodes. Our complete system implementation involves application-specific optimizations at the algorithm level, network level, and node level, and for this implementation, we demonstrate the resulting features of small size and high energy efficiency.

The purpose of this distributed WSN 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 being 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 Figure 1, the sensor nodes are placed in a circle inside a room. In this practical configuration, the WSN system recognizes when and where a subject has crossed the circle. All sensor nodes 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.

To demonstrate the practical realization of the targeted application, we develop a customized system prototype using an emerging family of off-the-shelf system-on-chip (SoCs), and a custom-designed miniature antenna that we have developed. Each SoC component employed provides an embedded microprocessor and wireless communications transceiver together within a single integrated circuit. Our design is different from related wireless sensor network implementations (e.g., see [1], [2]) in that a single SoC is used to replace separate integrated circuits for microcontroller and transceiver components, and furthermore, a smaller antenna is designed and installed. Both design features contribute to significantly reducing PCB size and the size of each network node.

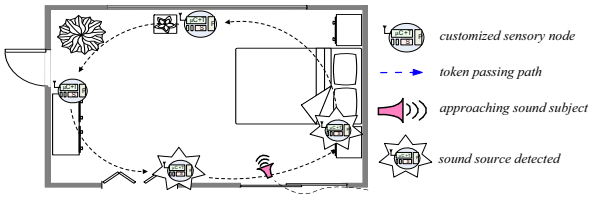

Fig. 1. An indoor environment scenario with the use of our distributed system for line-crossing recognition.

All N nodes within the system run the proposed distributed algorithm, and reach a consensus based on local decisions of C nodes while a subject is being detected ($C \leq N$). The number of data bits for the data that is communicated by each node is $O(log(C))$ (i.e., is logarithmically bounded with respect to C).

In [3], 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 their algorithm has an initial input, and has no additional inputs during its execution. In [4], 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). Their 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 requires 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 N. Furthermore, during most of the system lifetime, our implemented 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 is high, our approach reduces energy consumption significantly by reducing the number bits that need to be transmitted for overall system operation.

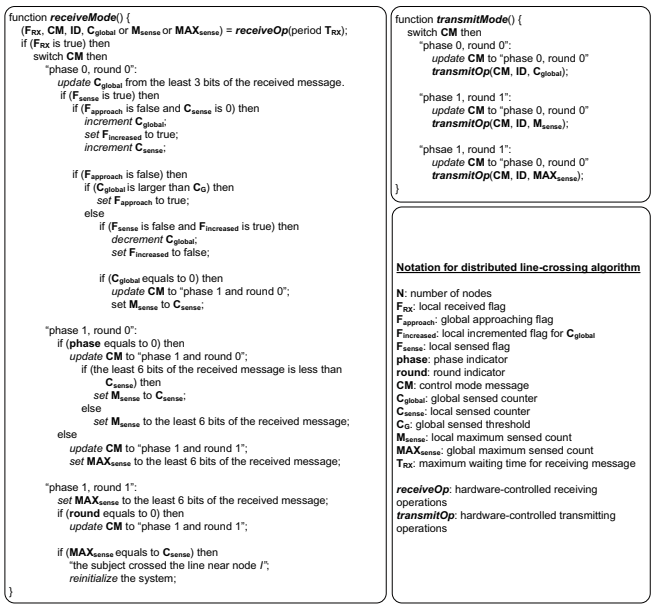

Fig. 2. Pseudo code specification for our distributed algorithm for line-crossing recognition.

II. Distributed Algorithm for Line-crossing Recognition

Our proposed algorithm, which runs on each node after the whole system is synchronized (i.e., at the communication stage), involves two phases of operation for line-crossing recognition. 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. We assume that each node has a unique identifier ID, where $0 \leq ID \leq N-1$. Figure 2 shows the pseudo code of the proposed distributed algorithm for line-crossing recognition, which each node runs the algorithm based on the TDMA-based communication pattern for *Transmission* and *Reception mode operations*. Notice that in the algorithm, hardware-controlled receiving and transmitting operations are dependent on targeted transceiver module. Therefore, we do not address the details of both functions due to the space limitation. The functionality of both functions encapsulates the lower level hardware configurations.

In phase 0 of the proposed algorithm, the nodes use a counter C_{global} to decide whether the subject is approaching based on local decisions of C nodes ($C \leq N$) that sensed the subject. In its turn, each node receives C_{global} from its previous neighbor (i.e., the node with ID i receives from node $i-1$). When a node senses that the subject is approaching, it increments C_{global} by one. Every node increments C_{global} at most once. Therefore, at any moment during the first stage of phase 0, the value of C_{global} indicates the number of nodes that sensed the subject. When C_{global} reaches C, all the nodes reach a consensus and decide that the subject is approaching. The node that increments C_{global} to C is the first node to set its *approaching flag* ($F_{approach}$) to 1. In their respective turns, all of the other nodes set their $F_{approach}$ values to 1.

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

In phase 1 of the proposed algorithm, 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_{sense} in every node. Phase 1 consists of 2 rounds. In the first round, the maximum sensed number is found. Node s transmits $C_{sense}(s)$ to its next neighbor. In its turn, each node i transmits the maximum of the number it received from node $i-1$ and $C_{sense}(i)$. At the end of the first round, node s receives the global maximum sensed count, $MAX_{sense} = max\{C_{sense}(i)\}$. Afterward, node s starts the second round, where it transmits MAX_{sense} to its next neighbor. Every node j — in its turn — compares MAX_{sense} to $C_{sense}(j)$. 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_{sense} to its next neighbor. At the beginning of the second round node, s holds MAX_{sense}, which is equal to $C_{sense}(l)$ ($0 \leq l \leq N-1$). Therefore, the second round ends at node l. That is, the subject crossed the line near node l, and the system can be reinitialized.

From our algorithm, it can be observed that only $log(C)$ bits are needed to represent C_{global} and a maximum of $log(N)$ bits are needed to represent C_{sense}. 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 $O(log(N))$.

III. TDMA-based Synchronization and Communication

TDMA-based communication protocols are often applied in small scale wireless communication systems [5], [6] 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. 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 $i-1$. 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 pre-defined 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. Figure 3 shows a schedule for a 4-node TDMA-based communication protocol. Figure 3 also illustrates the synchronization stage, where each node can be turned on at an arbitrary time. Based on the algorithm specification in Section II, our protocol requires $2 + log(C)$ ($O(log(C))$) bits to maintain system functionality during the communication stage, where the control information in a message packet occupies 2 bits, and the data occupies $log(C)$ bits. However, during the synchronization stage, nodes communicate with one another using 2 bits of control

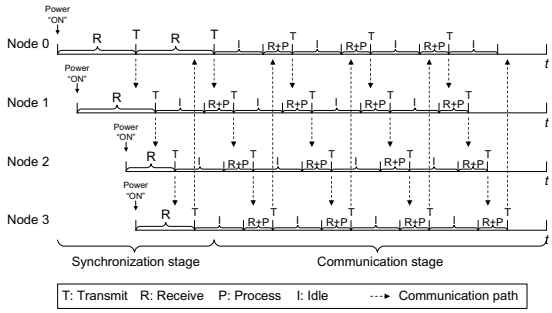

Fig. 3. TDMA-based communication pattern of distributed line-crossing recognition system for a four node example.

information and $log(N)$ bits of data (for node identifiers). Thus, $O(log(N))$ bits are needed for synchronization. Because the time that the system operates in the synchronization stage is transient and relatively short compared to the time of operation in the communication stage, the number of bits required to communicate is $O(log(C))$ in steady state system operation.

Now let T_{frame} denote the overall time of each TDMA communication frame (schedule period), and let T_{busy} denote the time during which a node is not idle in one frame. Then the *duty cycle* of the TDMA-based communication pattern illustrated in Figure 3 is

$$\frac{T_{busy}}{T_{frame}}, \text{where } T_{busy} \leq T_{frame}. \quad (1)$$

The duty cycle can be reduced at a node either by reducing the amount of non-idle time or by extending T_{frame}. Our developed distributed algorithm and communication protocol can exploit such a benefit since only small amounts of computation and communication are required to achieve the goals of sensing and of reaching consensus across the whole system.

IV. System Lifetime Analysis

The system (network) lifetime model in Eq. 2 below is derived from the analysis of pre-scheduled TDMA-based communication operations along with the characteristics of key hardware components, including the sensor, microcontroller, and transceiver. According to the communication protocol discussed in Section III, all operations within a TDMA frame (T_{frame}) appear periodically, frame-by-frame until the system fails (i.e., a node runs out of energy). Thus, in our lifetime model, we estimate the energy consumption for a single communication time frame, and then based on this estimation, the lifetime of the overall system can be estimated using data on the total energy stored in the available batteries:

$$T_{lifetime} = \frac{E_{total}}{P_{total}}, \text{and } P_{total} = \sum_{i=1}^{L} P_i \frac{T_i}{T_{frame}}, \quad (2)$$

where E_{total} denotes the total energy stored in a given battery; L denotes the number of devices considered in the power model; P_i denotes the average operational power consumption of the ith device; and T_i denotes the estimated time for which the ith device operates within a given frame.

We consider details of devices belonging to various hardware components for estimating the energy consumption. For example, the power consumption of A/D converters, hardware timers, and the CPU core are all considered whenever the microprocessor is used. Within distinct modes of operation for our TDMA-based communication protocol, the power consumption for devices can be

Symbols	Description	Value	Symbols	Description	Value
$t_{radioOn_{tx/rx}}$	startup time from power-on to valid transmit/receive	$50\mu s$	R	data rate	$500kbps$
$I_{radio_{tx/rx}}$	TX/RX mode current consumption	$31mA$ / $22mA$	M	data length	O(log(C)) or O(log(N)) *bits*
$I_{static/dynamic}$	static/dynamic current consumption in Active mode	$500\mu A$ / $270\ \mu A/MHz$	$t_{compute}$	CPU computation time	$10ms$
I_{idle}	Idle mode current consumption	$0.8\ \mu A$	I_{timer}, $I_{A/D}$, I_{sensor}, $I_{I/O}$	peripheral current consumption for timer, A/D converter, sensor, and I/O	$260\mu A$, $0.9mA$, $0.2mA$, $0.2mA$

Fig. 4. Notations for power modeling with reference values.

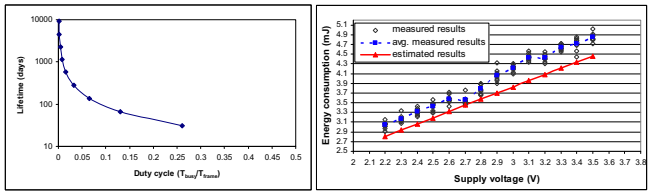

Fig. 5. (a) Analysis of duty cycle versus lifetime using $3V \cdot 950mAh$ capacity battery. (b) Fidelity analysis.

modeled carefully based on the *node-level* power models presented in [7]. For estimating the lifetime of the proposed WSN using TDMA-based communication, the total power consumption P_{total} is modeled by the following equation:

$$\begin{aligned} P_{total} = V_{cc}\{&(\frac{t_{radioOn_{tx}} + \frac{M}{R}}{T_{frame}})(I_{radio_{tx}} + \frac{P_{out}}{V_{cc}}) + \\ &(\frac{t_{radioOn_{rx}} + \frac{M}{R}}{T_{frame}})I_{radio_{rx}} + (\frac{t_{timer}}{T_{frame}})I_{timer} + \\ &(\frac{t_{compute}}{T_{frame}})(I_{static} + I_{dynamic}) + \\ &(\frac{T_{frame} - t_{radioOn_{tx/rx}} - t_{compute} - t_{timer} - \frac{2M}{R}}{T_{frame}})I_{idle} + \\ &(I_{A/D} + I_{sensor} + I_{I/O})\}. \end{aligned}$$

Figure 4 lists notations for the power model above with reference values (e.g., see [8]). For further details on the symbols and terms in this equation, we refer the reader to [7].

For demonstrating how configuration of the communication protocol affects the lifetime of the system, we evaluate different duty cycles of the protocol by changing T_{frame}. The idle time is varied along with T_{frame}, and the associated results of lifetime calculation are shown in Figure 5(a). Here, we can quantify how lower duty cycles lead to greater network longevity through the increasing proportion of idle time. Reduction of the duty cycle is limited by the latency constraint of the application, and thus lifetime can be maximized when the duty cycle is reduced to the lowest point at which the latency constraint can be reliably satisfied.

V. Fidelity Analysis

For presenting the accuracy of lifetime analysis using the estimator employed in our model, we calculate the fidelity of the estimator by comparing the estimated energy consumption to the average measured energy consumption within a TDMA frame T_{frame} (Figure 5(b)). The estimated energy consumption is based on the models described in Section IV. The measured energy consumption is tested from the hardware of the system prototype with real batteries (see Section VI-A). In this way, we can demonstrate in a quantitative manner the high accuracy of our lifetime estimation method. Based on this accuracy, the lifetime modeling, and the associated energy consumption models, can be applied with high confidence to analyze and optimize the

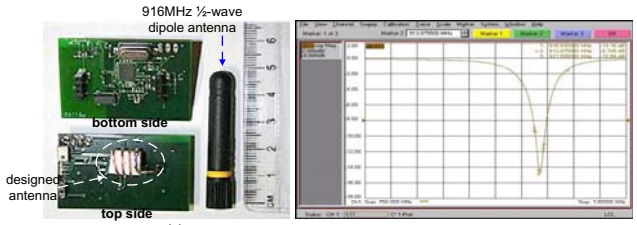

Fig. 6. (a) System prototype. (b) Return loss (S11) of the designed miniature antenna.

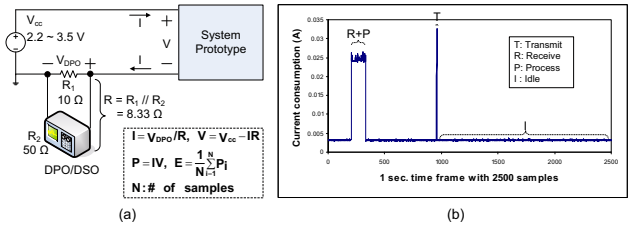

Fig. 7. (a) Experimental platform for voltage measurement. (b) Current consumption result within one communication frame period (T_{frame}), with T_{frame} set to one second.

lifetime of the WSN system. By applying 14 different voltage settings to the sensor node platform, and comparing the resulting measured vs. estimated energy consumption values, we obtain an estimation fidelity of *0.978* (a fidelity value of 1 corresponds to perfect fidelity). Due to space limitations we refer the reader to [9] for full details on the general model employed for deriving the fidelity of an estimator in terms of measured data. In Figure 5(b), we note that the estimated energy consumed is lower than the measured one on each working point. This is because the current leakage on the PCB board and the power consumed by other peripheral devices such as LEDs are not considered in the model. A more complete mixed-signal model for energy consumption will be explored in our future work.

VI. System Demonstration and Experimental Results

A. System Prototype

To implement the complete wireless sensor network system — based on the algorithms and protocols discussed in the previous sections — we have developed a streamlined, miniaturized antenna [10], and employed an emerging family of system-on-chip (SoC) devices [8] as an integrated, single-chip device for performing computation and communication tasks as well as a acoustic sensor for sensing tasks. The PCB design of our sensor node prototype includes a 4-layer layout, and provides placement of components for minimizing the board size and enhancing antenna performance. Figure 6 illustrates our sensor node prototype and our miniature antenna with its performance characteristics, as well as a comparison in size with a commercial 916MHz dipole antenna. The reflection coefficient at the feeding point of the antenna is measured through the Agilent Network Analyzer (PNA Series 8364B). The center frequency of our miniature antenna is 916MHz, with a return loss of 20dB and bandwidth of 13MHz.

B. Experimental Platform and Performance Results

Figure 7(a) represents a simple experimental platform for measuring voltage variations (i.e., V_{DPO}) and calculating the associated current, power and energy consumption for the system prototype. Figure 7(b) shows the current consumption result while the system prototype is running the proposed distributed algorithm and communication protocols over one TDMA frame T_{frame}, where T_{frame} is set to one second. The voltage variations are measured through the Tektronix 4GHz digital phosphor oscilloscope (TDS 7404). All of the measured/calculated values for current consumption and voltage are converted to corresponding energy consumption results for the fidelity analysis. This is illustrated in Figure 5(b).

VII. Conclusion

This paper has presented a complete system design — including algorithm streamlining, communication protocol configuration, hardware/software implementation, and lifetime modeling — for a compact and low power, distributed, sensor network system for line crossing recognition. Our proposed distributed algorithm for line crossing recognition 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 approach to modeling system lifetime provides an accurate method for estimating how long the network can operate with a given amount of battery capacity for each node. A customized PCB design of the system prototype is demonstrated to compactly support the developed sensor node platform. The sensor node platform in turn employs emerging, off-the-shelf, system-on-chip technology, and a custom-designed miniature antenna. Our ongoing and future work aims for further miniaturization and energy efficiency for the targeted line crossing recognition application by developing a fully-customized design, including a streamlined ASIC implementation of the mixed-signal SoC.

References

[1] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, pp. 12–24, November 2002.

[2] M. Kohvakka, M. Hannikanen, and T. Hamalainen, "Wireless sensor prototype platform," in *Proceedings of the IEEE International Conference for Industrial Electronics, Control and Instrumentation*, November 2003, pp. 1499–1504.

[3] D. S. Hirschberg and J. Sinclair, "Decentralized extrema-finding in circular configurations of processors," in *Commun. ACM 23 (ll)*, 1980, pp. 627–628.

[4] Y. Dinitz, S. Moran, and S. Rajsbaum, "Exact communication costs for consensus and leader in a tree," in *Journal of Discrete Algorithms 1(2)*, April 2003, pp. 167–183.

[5] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," in *IEEE/ACM Transactions on Networking*, June 2004, pp. 493–506.

[6] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, November 2003, pp. 171–180.

[7] C.-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 *Proceedings of the International Workshop on Computer Architecture for Machine Perception and Sensing*, Montreal, Canada, September 2006.

[8] *CC1110 Preliminary Datasheet (Rev. 1.01) SWRS033A*, Chipcon AS, 2006.

[9] N. K. Bambha and S. S. Bhattacharyya, "A joint power/performance optimization technique for multiprocessor systems using a period graph construct," in *Proceedings of the International Symposium on System Synthesis*, Madrid, Spain, September 2000, pp. 91–97.

[10] B. Yang, F. Vanin, C.-C. Shen, X. Shao, Q. Balzano, N. Goldsman, and C. Davis, "A low profile f-inverted compact antenna (fica) for wireless sensor networks," *submitted to International Symposium, Antenna and Propagation Society*, June 2007.