

# Providing Power Awareness to Wireless Microsensor Networks Using Tripwires

Caimu Tang

Department of Computer Science  
University of Southern California  
Los Angeles, CA 90089

Cauligi S. Raghavendra

Departments of EE-Systems and Computer Science  
University of Southern California  
Los Angeles, CA 90089

**Abstract**—Tripwire is a lightweight microsensor with limited processing capability. There are many new benefits of using tripwires in a sensor network. One of them is that sensor nodes can remain in sleep mode to conserve energy while tripwires can monitor a sensor field and wake up sensor nodes if further processing on the data is necessary. In this paper, we define two modes for sensor networks operating in a field, namely, monitoring mode and processing mode. We propose a collaborative two-stage detection scheme called DFAD which uses a group of tripwires to facilitate this. We evaluated the proposed scheme on field data sets using a wireless microsensor network with CYGNAL C8051 microcontrollers for detection processing. Our experimental results show that this scheme can provide power awareness to a sensor network with low overhead.

## I. INTRODUCTION

Wireless sensor networks have made many new applications possible [8]. Sensor nodes are constrained by various resources including energy. Algorithms and protocols which can relax these constraints and conserve energy are very useful for applications in this type of network. One approach to accomplish this is to use two kinds of nodes - signal processing sensor nodes (sensor nodes in short) and tripwire nodes (tripwires in short). The operating power for a sensor node is around an order of magnitude higher than that for a tripwire. Tripwire [5], [3], [2], [6] is one type of lightweight microsensor node which can perform some simple processing tasks, and it also has some limited communication capability. They can be deployed along with sensor nodes, and form a network to monitor a sensor field so that sensor nodes can be kept in sleep mode to conserve energy. Tripwires will wake up sensor nodes for various signal processing tasks only when necessary. By using tripwires, significant amount of energy can be saved since sensor nodes consume energy only for periods when events are happening. A network is made energy aware by using different operating powers based on field activities.

For energy conservation, we define two network operating modes as follows: (1) low power monitoring mode: sensor nodes are turned off, and tripwires remain in a sleep mode and switch to active mode on a regular basis to check field status; (2) high power processing mode: sensor nodes are woken up and process signals. The operating power difference between these two modes is significant, and it is at least an order of magnitude. Generally, it is a waste of energy to keep sensor nodes up longer than necessary. An important issue is how

accurately to match network operating modes to field activities without any loss of detection.

Let us assume event arrival to a sensor field follows Poisson process  $\{X(t), t \geq 0\}$  with instantaneous rate  $\lambda$ . We know that  $n$  inter-arrival periods  $a_1, \dots, a_n$  are i.i.d. random variables following geometrical distribution with mean  $\frac{1}{\lambda}$  and variance  $\frac{1}{\lambda^2}$ . Network idle time (i.e. no event periods) up to  $n$ -th arrival is a sum of  $n$  inter-arrival periods, which can be modeled as a random variable following Erlang distribution with p.d.f.  $f(x)$  as follows:

$$f(x) = 1 - \frac{\Gamma(n, x\lambda)}{\Gamma(n)},$$

where  $\Gamma(x)$  is a gamma function. Therefore, the mean network idle time is  $\frac{n}{\lambda}$ , and the variance of idle time is  $\frac{n}{\lambda^2}$ . Since both processing mode power and monitoring mode power can be modeled using uniform distribution with a small support interval, the average and worst case energy savings can be quantified using random variable derived from three independent random variables corresponding to idle time, monitoring operating power and processing operating power.

Alarm detection is performed by comparing a frame energy with a predefined energy threshold (in short, *threshold detection* is called for this detection in this paper). Tripwires may unnecessarily wake up sensor nodes by false alarms. After threshold detection, false alarms may occur in a sensor network due to a number of factors: (1) ambient noise, (2) measurement thermal noise, (3) A/D converter truncation error, (4) scattering/reflection and (5) multi-source. These make it not sufficient to simply use alarm detection. One common feature of these noises is that they are wideband in nature, and this feature is exploited in this paper for false alarm detection.

In this paper, we propose DFAD with two stages for wideband signal sources to make a sensor network power aware. Each node applies a threshold detection as the first-stage of detection. When a signal energy exceeds a predefined threshold, further detection processing is applied to eliminate false alarms. In order to detect a false alarm, a signal is first jointly transformed using a S-Transform [4] by a group of tripwires. DFAD assigns different tripwires to process different subtrees of a binary wavelet packet decomposition tree. These results in a form of detection predicates (i.e. false alarm or not false alarm) are fused by a designated tripwire. The final result of fusion processing is then used to determine whether to wake

up sensor nodes or not. For practical applications, as we are only interested in an energy distribution in subbands, there is no need to calibrate tripwire readings before a detection.

## II. COLLABORATIVE TRIPWIRE DETECTION

The proposed scheme has two separate stages: threshold detection and false alarm detection. Threshold detection is performed individually while false alarm detection is done collaboratively. In this section, we show the detection processing in one node followed by collaborative detection (details as how to eliminate some subbands using a reference frame and how to calibrate thresholds are omitted).

### A. Tripwire Detection in Single Node

In this study, we use the following formula *average squared amplitude* in dB to compute signal energy of a frame of fixed length in time domain for threshold detection:

$$W(k) = 10 \log \left( \frac{1}{n} \sum_{i=0}^n (x_i^k)^2 \right), \quad (1)$$

where,  $k$  is a frame index;  $n$  is the frame length and  $x_i^k$  is the  $i$ -th sample amplitude of  $k$ -th frame. For false alarm detection, the formula for computing a subband energy in wavelet domain is as follows:

$$W(u) = \sum_{i=0}^{n_u} (a_i^u - \bar{a}^u)^2, \quad (2)$$

where,  $u$  denotes a subband;  $n_u$  is the length of subband  $u$ ;  $\bar{a}^u$  is the average amplitude of subband  $u$ ;  $a_i^u$  denotes the  $i$ -th coefficient in wavelet subband  $u$ . Notice that the proposed algorithm works on zero-mean signal. In (2), we do not divide the sum by  $n_u - 1$  since we are interested in an energy ratio instead of subband energy (see (3)).

Threshold detection is done on a frame by frame basis. During the  $k$ -th frame period, a tripwire computes a frame energy using (1) to the end of this frame. It then compares the energy with a predefined threshold  $T_f$ . If  $W(k) \leq T_f$ , the tripwire continues to monitor the field; otherwise, it enters into the false alarm detection stage.

In order to explain technique for false alarm detection, let  $u$  be a non-leaf node in a pre-assigned wavelet subtree for a tripwire, and let us also denote two corresponding high-pass and low-pass subband transformed from  $u$  as  $u_2$  and  $u_1$ , respectively. We define the energy ratio as follows:

$$\mathcal{R}(u) = \frac{|\gamma_1 W(u_1) - \gamma_2 W(u_2)|}{|\gamma_1 W(u_1) + \gamma_2 W(u_2)|}, \quad (3)$$

where  $\gamma_1 W(u_1)$  and  $\gamma_2 W(u_2)$  are the energy for low-pass and high-pass subbands, respectively, of node  $u$ ;  $\gamma_1$  and  $\gamma_2$  are the normalization factors for a given transform for low-pass and high-pass subbands, respectively. When a signal has mean of zero, by Parseval theorem, the denominator in (3) actually equals to the energy of subband  $u$ . The normalization factors only depend on the basis and how the transform is actually performed (for S-Transform,  $\gamma_1$  is  $\sqrt{2}$  and  $\gamma_2$  is  $0.5\sqrt{2}$ ). Since

noise energy evenly spread across its frequency bands, small energy ratio can be a strong indicator for a false alarm.

The false alarm detection processing uses a greedy search method on a pre-assigned subtree. At each node of level  $i$  of wavelet decomposition (the decomposition indices start from 1), it first computes the energy ratio for a node using (3), and it then selects the node with a larger energy ratio to process in the next level. The detection processing iterates until either (1) a search reaches a predefined lowest level or (2) one node whose energy ratio is greater than threshold  $T_s$  is found. For case (1), a claim of false alarm denoted by  $H_a$  is generated, and for case (2), a claim of event denoted by  $H_0$  is generated. Figure 1 shows an example of case (2) where nodes in black are processed and it stops at the black node of the third level of the subtree. In Fig. 1, many nodes (in gray) are not processed

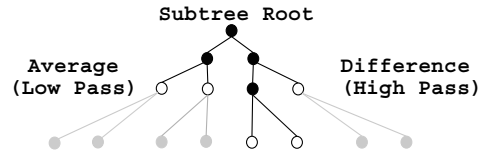


Fig. 1. Iteration Path of an  $H_0$  Predicate

since its energy ratio is smaller than the other child of the same parent. Nodes in circle are evaluated of the subband energy but no energy ratio is computed. In the greedy search, it looks ahead by one more level in order to determine the greater energy ratio. For a case of (1), the search stops at one of leaves. The memory required in this processing is not larger than half of original frame length, and memory required in next iteration is reduced by half.

### B. Distributed Tripwire Detection

After single node detection, the detection result (in a form of predicate) is sent to a designated tripwire. The designated tripwire makes a fusion decision using a boolean function on these received predicates. The final predicate will indicate  $H_a$  when all predicates indicate  $H_a$ , and the final predicate indicates  $H_0$  if at least one predicate indicates  $H_0$ .

In order to perform a distributed detection, we need to assign each node to a subtree of wavelet packet decomposition tree. We assume that nodes are sequentially indexed. We use a pyramidal wavelet decomposition scheme since underlying signals could be of low-pass, high-pass or band-pass type. The tree structure corresponding to a wavelet packet basis [7] of a pyramidal wavelet decomposition scheme is predefined based on application signal characteristics, and partition of a tree is also predefined based on various application requirements (e.g. delay, accuracy). As an example for packet basis selection, for a case of vehicle acoustic tracking, a basis selection may have to take into account what type of engine (e.g. Turbo or non-Turbo). In the case of Turbo engine, the basis should include more low-pass subbands. In the experimental study, we have both high-pass and low-pass acoustic signal data sets, so we select a packet basis for these types of acoustic signals as explained in Sec. III. With the partition information

made available to a tripwire, each tripwire selects a subtree to process based on its index.

As a general guideline on tree partition, all nodes in a wavelet packet decomposition tree should be covered by at least one tripwire when there is no priori knowledge of underlying signal available to tripwires, nodes in a given frequency band should be covered by at least one tripwire when underlying signal is most possibly in that frequency band (p priori knowledge). As for basis selection, an evenly balanced binary wavelet packet decomposition tree should be employed when no priori knowledge of underlying signal is available. However, coarser decomposition (i.e. more level of decomposition) in a frequency band should be performed when underlying signal is most possibly in that frequency band, and the corresponding tree is unbalanced (as one such example tree is used in Sec. III). We omit the detailed analysis and selection criteria on tree partition and packet basis due to space limit.

At the end of detection processing in each node, a detection predicate indicating  $H_a$  or  $H_0$  is generated corresponding to a node's subtree. Each node broadcasts a message containing its predicate and index. The message contains one bit for a predicate and  $\lceil \log(N) \rceil$  bits for node index where  $N$  is the total number of nodes in a field. A fusion node, which is also determined based on its index (e.g. its index is a multiple of 3 for the case of three tripwires per detection), collects these messages, and produces a final detection predicate using a boolean function. Let  $p_i$  be the predicate from node  $i$ , where  $p_i = 1$  for  $H_0$  and  $p_i = 0$  for  $H_a$ , and  $\mathcal{P}$  denotes the final predicate,  $\mathcal{P}$  is computed by the pseudocode in Fig. 2.

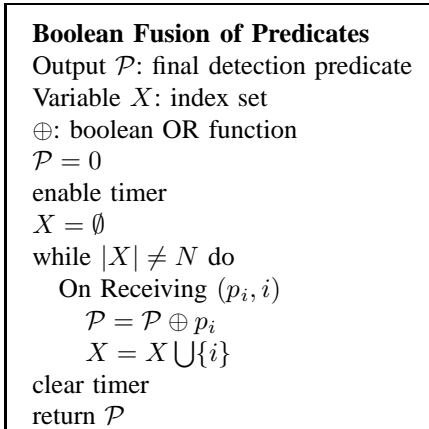


Fig. 2. Pseudocode of Boolean Predicate Fusion

As an intrinsic problem associated with tripwire is unreliability due to its cost-effective commodity components, and any single tripwire could be down or malfunction at any time. Under DFAD, to increase detection robustness, two tripwires can be assigned to two overlapped subtrees. Especially when some priori knowledge about field signals is available, two tripwires can be assigned to overlapped subtrees corresponding to frequency bands in which signals of interest are most possible. To accommodate these cases, the exit criterion in the pseudocode should be modified slightly.

### III. EXPERIMENTAL STUDIES

In this section, we show our experimental results. These results include detection performance, processing times, energy cost of a tripwire and a fusion node as compared to a non-distributed detection (all processing is done by a single tripwire with a whole pramidal wavelet decomposition tree).

We tested the scheme based on the wavelet packet tree partition shown in Fig. 3, where three tripwires are used in each detection. The subbands in light gray to the left of the tree are assigned to tripwire 1; the subtree in black in the middle part of the tree is assigned to tripwire 2; the subtree in dark gray to the right of the tree is assigned to tripwire 3; the subtree in the middle in light gray is ignored. However, we are not limited to a given tree partition, other tree partition can be used provided that it fits well to underlying signal characteristics (either low-pass smooth signal or high-pass signal). Notice that

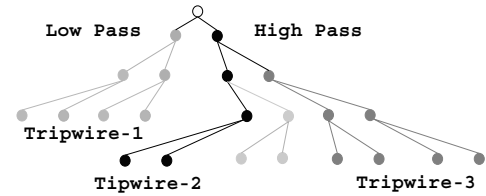


Fig. 3. A Three-Tripwire Case: Subtree Assignment

one subtree of the partition shown in Fig. 3 is not processed, and this is due to the fact that the field signals are not of band-pass type.

In our implementation, as we can spread processing to multiple tripwires in DFAD, the processing can be completely kept on-chip on CYGNAL C8051 F020 [1] at each tripwire. This microcontroller has 4352 bytes on-chip memory and 64K off-chip memory. Access to off-chip memory is close to 7 times more costly than on-chip memory access, and this situation is common in microsensors. The energy dissipation is kept at a minimum level when all memory access is kept on-chip and the tripwire is clocked by an internal oscillator. For DFAD, the implementation is done with only on-chip memory access with internal oscillator set at 16 mHz while the non-distributed detection requires off-chip memory access and an external oscillator due to processing delay constraint.

The left plot and right plot of Fig. 4 show subband energy ratios of an event case and a false alarm case, respectively. From the left plot, we can see that there are a few peaks which correspond to some energy cluster of event-signal spectrum, and these phenomena do not present in the right plot for the false alarm signal.

The left plot and right plot of Fig. 5 show time domain representations of an event signal and false alarm signal (ambient noise), respectively, for a period of 20 seconds. The left plot and right plot of Fig. 6 show the energy per frame of the event signal and false alarm signal, respectively, for the same period. Once signal energy in a frame (256 samples, i.e. quarter second resolution for threshold detection) is greater than  $-30$  dB ( $T_f = -30dB$ ), tripwires are alarmed and they

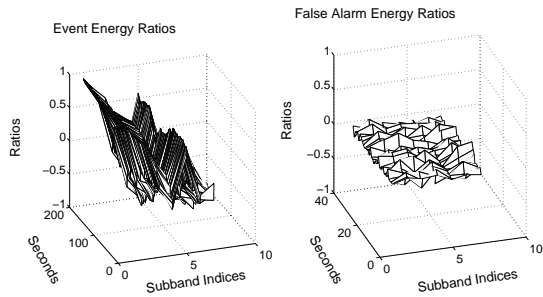


Fig. 4. Energy Ratio Comparison of Event Signal and False Alarm Signal

start false detection processing on succeeding frames of length 1024.

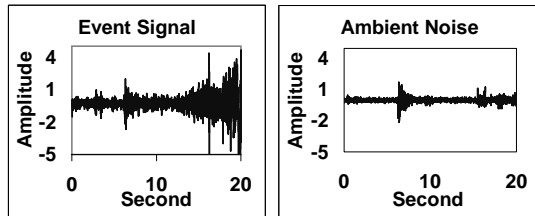


Fig. 5. Data Sources: Event Signal vs. Ambient Noise in Time Domain

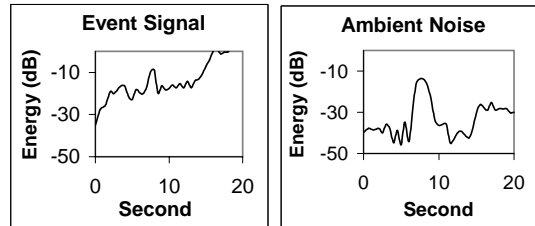


Fig. 6. Frame Energy for Threshold Detection

The left plot and right plot of Fig. 7 show energy ratios of these three tripwire for 20 seconds when threshold detection results are positive. Notice that in Fig. 7 tripwire 1 has a subtree of height 4, and tripwire 2 and 3 have a subtree of height 5. To plot these curves in Fig. 7, at each frame, we select the maximum of 3 ratios for Tripwire 2 and Tripwire 3, while we use the ratios, which exceed  $T_s$  for the first time in the subtree, for Tripwire 1. In Fig. 7,  $T_s$  is set at 0.2 which is pre-determined offline based on object types. If  $T_s$  is set too small, some false alarms may be missed; however, it may delay or could miss signal processing on an event if the threshold is set too large. Since the minimum energy ratio of signal subband is much greater than the maximum energy ratio of noise subbands based on the studied data sets, a proper threshold for a particular application can be easily found using a training data set offline. From the plots in Fig. 7 and the subtree assignment, we can see this signal is a low-pass type; however, in general it may not be the case.

Table I shows the average processing delay and energy consumption per detection under DFAD and non-distributed detection using a single tripwire (shown in last row). The

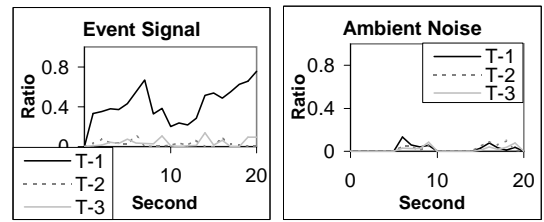


Fig. 7. Detection Results Per Tripwire

TABLE I  
ENERGY AND TIME COMPARISONS ON CYGNAL C8051 F020  $\mu$ C

	T. (ms)	Comm. T. (ms)	Energy (mJ)
T-1	80	45	1.35
T-2	90	43	1.33
T-3	130	57	1.97
Total	300	145	4.7
Single	370	0	4.5

communication time is for the duration when the transceiver is on, and the actual transmission time is less than 1 ms using RFM TR1000 radio. It is clear that the total time needed by DFAD is much less than that needed by non-distributed detection. With added communication cost, the overall energy of DFAD is still comparable to that of non-distributed detection. We must note that the overall delay of each tripwire under DFAD is much less than that of a single tripwire under non-distributed detection.

#### IV. CONCLUSIONS

In this paper, we presented a distributed detection scheme DFAD to provide power awareness to a sensor network. It is suitable for energy efficient operations of sensor networks.

#### ACKNOWLEDGMENTS

This research is partially supported by the DARPA under PAC/C project, U.S. Air Force Research Laboratory, U.S. Air Force Material Command and USAF. The views and conclusions contained herein are solely those of the authors.

#### REFERENCES

- [1] Silicon Laboratories Inc., "C8051 Microcontroller [online]," Available: <http://www.silabs.com>.
- [2] Crossbow Technology Inc., "Mica Motes [online]," Available: <http://www.xbow.com>.
- [3] H. Abrach, S. Bhatti, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Shucker, J. Deng, and R. Han, "MANTIS: System Support For Multimodal Networks of In-situ Sensors," 2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA) 2003, pp. 50-59.
- [4] A. R. Calderbank, I. Daubechies, W. Sweldens, and D. Yeo, "Wavelet Transforms That Map Integers to Integers", Applied and Computational Harmonic Analysis, vol. 5, no. 3, pp. 332-369, July 1998.
- [5] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets and T. Tuan, "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design", Proceedings of the International Solid-State Circuits Conference, San Francisco, CA, February 3-7, 2002.
- [6] S. Roundy, P. K. Wright and J. Rabaey, "Energy Scavenging for Wireless Sensor Networks", Kluwer Academic Publishers, 2003.
- [7] M. V. Wicherhauser, "INRIA Lectures on Wavelet Packet Algorithms", <http://www.math.wustl.edu/~victor/papers/lwpa.pdf>, Nov. 1991.
- [8] C.S. Raghavendra, K. M. Sivalingam and T. Znati, "Wireless Sensor Networks" (edited), Kluwer Academic Publishers, 2004.