Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks

Bhaskar Krishnamachari, *Member*, *IEEE*, and Sitharama Iyengar, *Fellow*, *IEEE*

Abstract—We propose a distributed solution for a canonical task in wireless sensor networks—the binary detection of interesting environmental events. We explicitly take into account the possibility of sensor measurement faults and develop a distributed Bayesian algorithm for detecting and correcting such faults. Theoretical analysis and simulation results show that 85-95 percent of faults can be corrected using this algorithm, even when as many as 10 percent of the nodes are faulty.

Index Terms—Fault tolerance, event detection, sensor fusion, Bayesian algorithms, wireless sensor networks.

1 Introduction

Wireless sensor networks are envisioned to consist of thousands of devices, each capable of some limited computation, communication, and sensing, operating in an unattended mode. According to a recent National Research Council report, the use of such networks of embedded systems "could well dwarf previous revolutions in the information revolution" [26]. These networks are intended for a broad range of environmental sensing applications from vehicle tracking to habitat monitoring [9], [13], [15], [26].

In general, sensor networks can be tasked to answer any number of queries about the environment [22]. We focus on one particular class of queries: determining event regions in the environment with a distinguishable characteristic. As an example, consider a network of devices that are capable of sensing concentrations of some chemical X; an important query in this situation could be "Which regions in the environment have a chemical concentration greater than λ units?" We will refer to the process of getting answers to this type of query as *event region detection*.

Event region detection is useful in and of itself as a useful application of a sensor network. While event region detection can certainly be conducted on a static sensor network, it is worthwhile pointing out that it can also be used as a mechanism for nonuniform sensor deployment. Information about the location of event regions can be used to move or deploy additional sensors to these regions in order to get finer-grained information.

Wireless sensor networks are often unattended, autonomous systems with severe energy constraints and low-end individual nodes with limited reliability. In such conditions, self-organizing, energy-efficient, fault-tolerant algorithms are required for network operation. These design themes will guide the solution proposed in this paper to the problem of event region detection.

To our knowledge, this is the first paper to propose a solution to the fault-event disambiguation problem in sensor networks. Our proposed solution, in the form of Bayesian fault recognition algorithms, exploits the notion that measurement errors due to faulty equipment are likely to be uncorrelated, while environmental conditions are spatially correlated. We show through theoretical and simulation results that the optimal threshold decision algorithm we present can reduce sensor measurement faults by as much as 85-95 percent for fault rates up to 10 percent.

We begin with a short introduction to some of the prior work in the area of wireless sensor networks before proceeding to discuss the event region detection problem and our solution in greater detail.

1.1 Wireless Sensor Networks

A number of independent efforts have been made in recent years to develop the hardware and software architectures needed for wireless sensing. The challenges and design principles involved in networking these devices are discussed in a number of recent works [1], [4], [12], [13], [26]. A good recent survey of sensor networks can be found in [34].

Self-configuration and self-organizing mechanisms are needed because of the requirement of unattended operation in uncertain, dynamic environments. Some attention has been given to developing localized, distributed, self-configuration mechanisms in sensor networks [10], [20] and studying conditions under which they are feasible [23].

Sensor networks are characterized by severe energy constraints because the nodes will often operate with finite battery resources and limited recharging. The energy concerns can be addressed by engineering design at all layers. It has been recognized that energy savings can be obtained by pushing computation within the network in the form of localized and distributed algorithms [4], [21], [22].

- *B. Krishnamachari is with the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089-2562. E-mail: bkrishna@usc.edu.*
- *S. Iyengar is with the Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803-4001. E-mail: iyengar@bit.csc.lsu.edu.*

Manuscript received 16 Sept. 2002; revised 10 Mar. 2003; accepted 3 June 2003.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 117358.

One of the main advantages of the distributed computing paradigm is that it adds a new dimension of robustness and reliability to computing. Computations done by clusters of independent processors need not be sensitive to the failure of a small portion of the network. Wireless sensor networks are an example of large scale distributed computing systems where fault tolerance is important. For large scale sensor networks to be economically feasible, the individual nodes necessarily have to be low-end inexpensive devices. Such devices are likely to exhibit unreliable behavior. Therefore, it's important to guarantee that faulty behavior of individual components does not affect the overall system behavior. Some of the early work in the area of distributed sensor networks focuses on reliable routing with arbitrary network topologies [17], [18], characterizing sensor fault modalities [5], [6], tolerating faults while performing sensor integration [19], and tolerating faults while ensuring sensor coverage [16]. A mechanism for detecting crash faults in wireless sensor networks is described in [25]. There has been little prior work in the literature on detecting and correcting faults in sensor measurements in an application-specific context. We now discuss the canonical problem of event region detection.

The optimal Bayesian decision algorithm we present in this paper is closely related to the classic voting algorithms studied in distributed applications [35], [36]. The basic idea in voting is to get a quorum of nodes to agree on an operation before commitment. In the context of sensor networks, voting algorithms (such as unanimous voting, majority voting, m-out-of-n voting, and plurality voting) have been recommended as a mechanism for fusing the decisions of multimodal sensors with low communication overhead [37].

2 Event Region Detection

Consider a wireless network of sensors placed in an operational environment. We wish to task this network to identify the regions in the network that contain interesting events. For example, if the sensors monitor chemical concentrations, then we want to extract the region of the network in which these concentrations are unusually high. It is assumed that each sensor knows its own geographical location, either through GPS or through RF-based beacons [27].

It is helpful to treat the trivial centralized solution to the event region detection problem first in order to understand the shortcomings of such an approach. We could have all nodes report their individual sensor measurements, along with their geographical location directly to a central monitoring node. The processing to determine the event regions can then be performed centrally. While conceptually simple, this scheme does not scale well with the size of the network due to the communication bottlenecks and energy expenses associated with such a centralized scheme. Hence, we would like a solution in which the nodes in an event region organize themselves and perform some local processing to determine the extent of the region. This is the approach we will take.

Even under ideal conditions, this is not an easy problem to solve due to the requirement of a distributed, self-organized approach. However, if we take into account the possibility of sensor measurement faults, there is an additional layer of complexity. Can unreliable sensors decide on their own if their measurement truly indicates a high event value or if it is a faulty measurement? In general, this is an intractable question. It is true, however, that the sensor measurements in the operation region are spatially correlated (since many environmental phenomena are), while sensor faults are likely to be uncorrelated. As we establish in this paper, we can exploit such a problem structure to give us a distributed, localized algorithm to mitigate the effect of errors in sensor measurements.

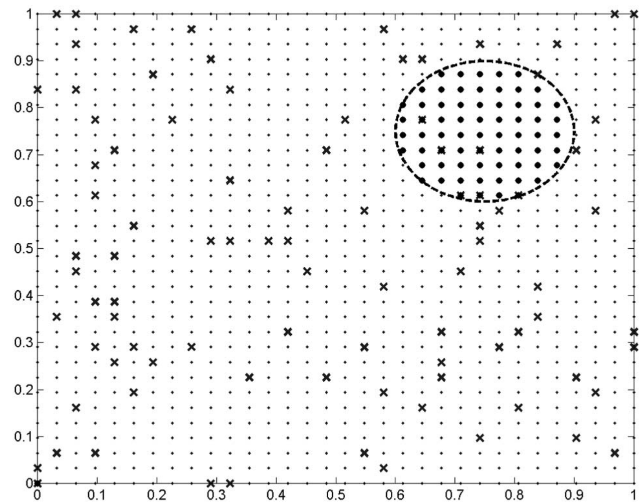


Fig. 1. Sample scenario: A distributed sensor network with uncorrelated sensor faults (denoted as "x") deployed in an environment with a single event region (dashed circle).

Fig. 1 shows a sample scenario. In this situation, we have a grid of sensors in some operational area. There is an event region with unusually high chemical concentrations. Some of the sensors shown are faulty in that they report erroneous readings.

The first step in event region detection is for the nodes to determine which sensor readings are interesting. In general, we can think of the sensor's measurements as a real number. There is some prior work on systems that learn the normal conditions over time so that they can recognize unusual event readings [28]. We will instead make the reasonable assumption that a threshold that enables nodes to determine whether their reading corresponds to an event has been specified with the query or otherwise made available to the nodes during deployment.

A more challenging task is to disambiguate events from faults in the sensor readings since an unusually high reading could potentially correspond to both. Conversely, a faulty node may report a low measurement even though it is in an event region. In this paper, we present probabilistic decoding mechanisms that exploit the fact that sensor faults are likely to be stochastically uncorrelated, while event measurements are likely to be spatially correlated. In analyzing these schemes, we will show that the impact of faults can be reduced by as much as 85-95 percent, even for reasonably high fault rates.

3 Fault Recognition

Without loss of generality, we will assume a model in which a particularly large value is considered unusual, while the normal reading is typically a low value. If we allow for faulty sensors, sometimes such an unusual reading could be the result of a sensor fault, rather than an indication of the event. We assume environments in which event readings are typically spread out geographically over multiple contiguous sensors. In such a scenario, we can disambiguate faults from events by examining the correlation in the reading of nearby sensors.

Let the real situation at the sensor node be modeled by a binary variable T_i. This variable $T_i = 0$ if the ground truth is that the node is a normal region and $T_i = 1$ if the ground truth is that the node is in an event region. We map the real output of the sensor into an abstract binary variable S_i. This variable $S_i = 0$ if the sensor measurement indicates a normal value and $S_i = 1$ if it measures an unusual value.

There are thus four possible scenarios: $S_i = 0, T_i = 0$ (sensor correctly reports a normal reading), $S_i = 0, T_i = 1$ (sensor faultily reports a normal reading), $S_i = 1, T_i = 1$ (sensor correctly reports an unusual/event reading), and $S_i = 1, T_i = 0$ (sensor faultily reports an unusual reading). While each node is aware of the value of S_i, in the presence of a significant probability of a faulty reading, it can happen that $S_i \neq T_i$. We describe below a Bayesian fault recognition algorithm to determine an estimate R_i of the true reading T_i after obtaining information about the sensor readings of neighboring sensors.

In our discussions, we will make one simplifying assumption: The sensor fault probability p is uncorrelated and symmetric. In other words,

$$P(S_i = 0 | T_i = 1) = P(S_i = 1 | T_i = 0) = p. \tag{1}$$

The binary model can result from placing a threshold on the real-valued readings of sensors. Let m_n be the mean normal reading and m_f the mean event reading for a sensor. A reasonable threshold for distinguishing between the two possibilities would be $0.5(m_n + m_f)$. If the errors due to sensor faults and the fluctuations in the environment can be modeled by Gaussian distributions with mean 0 and a standard deviation σ, the fault probability p would indeed be symmetric. It can be evaluated using the tail probability of a Gaussian, the Q-function, as follows:

$$p = Q\left(\frac{(0.5(m_f + m_n) - m_n)}{\sigma}\right) = Q\left(\frac{m_f - m_n}{2\sigma}\right). \tag{2}$$

We know that the Q-function decreases monotonically. Hence, (2) tells us that the fault probability is higher when $(m_f - m_n)$ is low, when the mean normal and event readings are not sufficiently distinguishable, or when the standard deviation σ of the sensor measurement errors. The assumption that sensor failures are uncorrelated is a standard, reasonable assumption because these failures are primarily due to imperfections in manufacturing and not a function of the nodes' spatial deployment. The algorithms and analysis presented in this paper may be extended to nonsymmetric errors in a straightforward manner; the symmetry assumption is made primarily for ease of exposition.

We also wish to model the spatial correlation of event values. Let each node i have N neighbors (excluding itself). Let's say the evidence $E_i(a,k)$ is that k of the neighboring sensors report the same binary reading a as node i, while $N - k$ of them report the reading $\neg a$, then we can decode according to the following model for using the evidence:

$$P(R_i = a | E_i(a,k)) = \frac{k}{N}. \tag{3}$$

Note that, in networks that are deployed with high densities, nearby sensors are likely to have similar event readings unless they are at the boundary of the event region. In this model, we have that a sensor gives equal weight to the evidence from each neighbor. More sophisticated models may be possible, but this model commends itself as a robust mechanism for unforseen environments.

Now, the task for each sensor is to determine a value for R_i given information about its own sensor reading S_i and the evidence $E_i(a,k)$ regarding the readings of its neighbors. The following Bayesian calculations provide the answer:

$$\begin{aligned} P(R_i = a | S_i = b, E_i(a,k)) &= \frac{P(R_i = a, S_i = b | E_i(a,k))}{P(S_i = b | E_i(a,k))} \\ &= \frac{P(S_i=b|R_i=a)P(R_i=a|E_i(a,k))}{P(S_i=b|R_i=a)P(R_i=a|E_i(a,k)) + P(S_i=b|R_i=\neg a)P(R_i=\neg a|E_i(a,k))} \\ &\approx \frac{P(S_i=b|T_i=a)P(R_i=a|E_i(a,k))}{P(S_i=b|T_i=a)P(R_i=a|E_i(a,k)) + P(S_i=b|T_i=\neg a)P(R_i=\neg a|E_i(a,k))}, \end{aligned} \tag{4}$$

where the last relation follows from the fact that R_i is meant to be an estimate of T_i. Thus, we have, for the two cases $(b = a)$, $(b = \neg a)$:

$$\begin{aligned} P_{aak} = P(R_i = a | S_i = a, E_i(a,k)) &= \frac{(1-p)\frac{k}{N}}{(1-p)\frac{k}{N} + p(1 - \frac{k}{N})} \\ &= \frac{(1-p)k}{(1-p)k + p(N-k)} \end{aligned} \tag{5}$$

$$\begin{aligned} P(R_i = \neg a | S_i = a, E_i(a,k)) &= 1 - P(R_i = a | S_i = a, E_i(a,k)) \\ &= \frac{p(N-k)}{(1-p)k + p(N-k)}. \end{aligned} \tag{6}$$

Equations (5), (6) show the statistic with which the sensor node can now make a decision about whether or not to disregard its own sensor reading S_i in the face of the evidence $E_i(a,k)$ from its neighbors.

Each node could incorporate randomization and announce if its sensor reading is correct with probability P_{aak}. We will refer to this as the randomized decision scheme.

An alternative is a threshold decision scheme, which uses a threshold $0 < \Theta < 1$ as follows: If $P(R_i = a | S_i = a, E_i(a,k)) > \Theta$, then R_i is set to a and the sensor believes that its sensor reading is correct. If the metric is less than the threshold, then node i decides that its sensor reading is faulty and sets R_i to $\neg a$.

The detailed steps of both schemes are depicted in Table 1, along with the optimal threshold decision scheme,

TABLE 1
Decision Schemes for Fault Recognition

Randomized Decision Scheme
1. Obtain the sensor readings S_j of all N_i neighbors of node i.
2. Determine k_i, the number of node i's neighbors j with $S_j = S_i$.
3. Calculate $P_{aak} = \frac{(1-p)k_i}{(1-p)k_i + p(N_i - k_i)}$.
4. Generate a random number $u \in (0,1)$.
5. If $u < P_{aak}$, set $R_i = S_i$ else set $R_i = \neg S_i$.
Threshold Decision Scheme
1. Obtain the sensor readings S_j of all N_i neighbors of node i.
2. Determine k_i, the number of node i's neighbors j with $S_j = S_i$.
3. Calculate $P_{aak} = \frac{(1-p)k_i}{(1-p)k_i + p(N_i - k_i)}$.
4. If $P_{aak} > \Theta$, set $R_i = S_i$, else set $R_i = \neg S_i$.
Optimal Threshold Decision Scheme
1. Obtain the sensor readings S_j of all N_i neighbors of node i.
2. Determine k_i, the number of node i's neighbors j with $S_j = S_i$.
3. If $k_i \geq 0.5N_i$, set $R_i = S_i$, else set $R_i = \neg S_i$.

which we will discuss later in the analysis. It should be noted that, with either the randomized decision scheme or the threshold decision scheme, the relations in (5) and (6) permit the node to also indicate its confidence in the assertion that $R_i = a$.

We now proceed with an analysis of these decoding mechanisms for recognizing and correcting faulty sensor measurements.

4 Analysis of Fault-Recognition Algorithm

In order to simplify the analysis of the Bayesian fault recognition mechanisms, we will make the assumption that, for all N neighbors of node i, the ground truth is the same. In other words, if node i is in an event region, so are all its neighbors and, if i is not in an event region, neither are any of its neighbors. This assumption is valid everywhere except at nodes which lie on the boundary of an event region. For sensor networks with high density, this is a reasonable assumption as the number of such boundary nodes will be relatively small. Table 2 sumarizes the notation we will use in our analysis. We will first present results for the randomized decision scheme.

Let g_k be the probability that exactly k of node i's N neighbors are not faulty. This probability is the same irrespective of the value of T_i. This can be readily verified:

$$\begin{aligned} g_k &= \binom{N}{k} P(S_i = 0 | T_i = 0)^k P(S_i = 1 | T_i = 0)^{(N-k)} \\ &= \binom{N}{k} P(S_i = 1 | T_i = 1)^k P(S_i = 0 | T_i = 0)^{(N-k)} \\ &= \binom{N}{k} (1-p)^k p^{(N-k)}. \end{aligned} \tag{7}$$

With binary values possible for the three variables corresponding to the ground truth T_i, the sensor measurement S_i, and the decoded message R_i, there are eight possible combinations. The conditional probabilities corresponding to these combinations are useful metrics in analyzing the performance of this fault recognition algorithm.

Consider first the probability $P(R_i = 0 | S_i = 0, T_i = 0)$. This is the probability that the algorithm estimates that

TABLE 2
Summary of Notation for Analysis of Fault-Recognition

Symbol	Definition
n	Total number of deployed nodes.
n_f	Number of nodes in the event region.
n_o	Number of other nodes $= n - n_f$.
N	The number of neighbors of each node
T_i	The binary variable indicating the ground truth at node i.
S_i	The binary variable indicating the sensor reading. Sensor is faulty $\iff S_i = \neg T_i$.
R_i	The binary variable with the decoded value. Decoding is correct $\iff R_i = T_i$
$E_i(a,k)$	The event that k of node i's N neighbors have the same sensor reading a.
P_{aak}	The conditional probablility $P(R_i = a \mid S_i = a, E_i(a,k))$.
p	The (symmetric) fault probability $P(S_i = 1 \mid T_i = 0) = P(S_i = 0 \mid T_i = 1)$.
g_k	The probability that k of node i's N neighbors are not faulty.
Θ	The decision threshold
α	The average number of errors after decoding
β	The average number of errors corrected
γ	The average number of errors uncorrected
δ	The average number of new errors introduced

there is no event reading when the sensor is not faulty and indicates that there is no event.

$$P(R_i = 0 \mid S_i = 0, T_i = 0) = \sum_{k=0}^{N} P(R_i = 0 \mid S_i = 0, T_i = 0, E_i(0,k)) = \sum_{k=0}^{N} P_{aak} g_k. \tag{8}$$

In a similar manner, we can derive the following expressions for all these conditional probabilities:

$$P(R_i = a \mid S_i = a, T_i = a) = 1 - P(R_i = \neg a \mid S_i = a, T_i = a) = \sum_{k=0}^{N} P_{aak} g_k \tag{9}$$

$$P(R_i = \neg a \mid S_i = \neg a, T_i = a) = 1 - P(R_i = a \mid S_i = \neg a, T_i = a) = \sum_{k=0}^{N} P_{aak} g_{N-k.} \tag{10}$$

These metrics suffice to answer questions such as the expected number of decoding errors α, obtained by marginalizing over values for S_i.

$$\alpha = P(R_i = 1 \mid T_i = 0) n_o + P(R_i = 0 \mid T_i = 0) n_f = \left(1 - \sum_{k=0}^{N} P_{aak}(g_k - g_{N-k})\right) n. \tag{11}$$

The reduction in the average number of errors is therefore $(np - \alpha)/np$.

We can also now talk meaningfully about β, the average number of sensor faults corrected by the Bayesian fault recognition algorithm. The conditional probabilities in (9) and (11) tell us about this metric:

$$\beta = \left(1 - \sum_{k=0}^{N} P_{aak} g_{N-k}\right) np. \tag{12}$$

A related metric is γ, the average number of faults uncorrected:

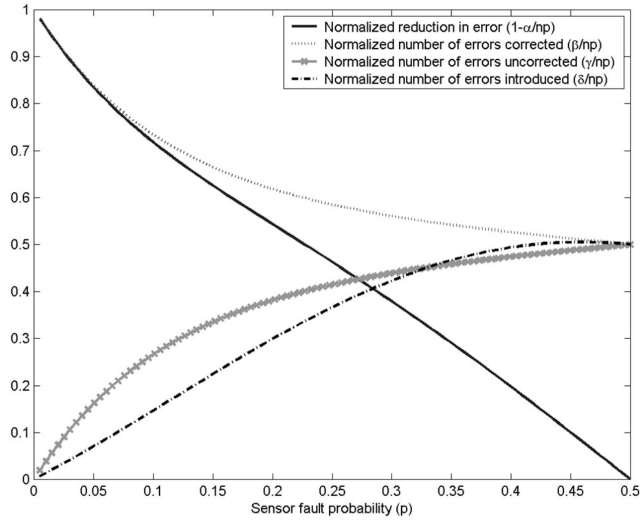


Fig. 2. Metrics for the Bayesian fault recognition algorithm with randomized decision scheme (N = 4).

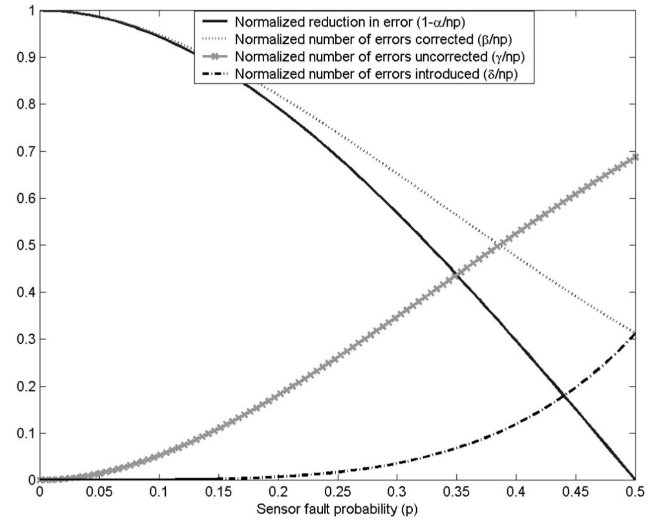


Fig. 3. Metrics for the Bayesian fault recognition algorithm with optimal threshold decision scheme (N = 4).

$$\gamma = \left(\sum_{k=0}^{N} P_{aak} g_{N-k} \right) np. \tag{13}$$

The Bayesian fault recognition algorithm has one setback —while it can help us correct sensor faults, it may introduce new errors if the evidence from neighboring sensors is faulty. This effect can be captured by the metric δ, the average number of new errors introduced by the algorithm:

$$\begin{aligned} \delta &= P(R_i = 1 | S_i = 0, T_i = 0)(1-p)n_o \\ &\quad + P(R_i = 0 | S_i = 1, T_i = 1)(1-p)n_f \\ &= \left(1 - \sum_{k=0}^{N} P_{aak} g_k \right)(1-p)n. \end{aligned} \tag{14}$$

These metrics are shown in Fig. 2 with respect to the sensor fault probability p. While it can be seen that, for $p < 0.1$ (10 percent of the nodes being faulty on average), over 75 percent of the faults can be corrected. However, the number of new errors introduced δ is seen to increase steadily with the fault-rate and starts to affect the overall reduction in errors significantly after about $p = 0.1$.

Let us now consider the threshold decision scheme. Fig. 3 shows the corresponding metrics for the threshold decision scheme. The following theorem tells us that we can view the threshold scheme from an alternate perspective.

Theorem 1. *The decision threshold scheme with Θ is equivalent to picking an integer k_{min} such that node i decodes to a value $R_i = S_i = a$ if and only if at least k_{min} of its N neighbors report the same sensor measurement a.*

Proof. Recall that, in this scheme, $R_i = a \Longleftrightarrow P_{aak} > \Theta$. It suffices to show that P_{aak} increases monotonically with k since, in this case, for each Θ, there is some k_{min} beyond which R_i is always set to a. We can rewrite (5) as follows:

$$P_{aak} = \frac{(1-p)k}{k(1-2p) + pN}. \tag{15}$$

The monotonicity can be shown by taking the derivative of this with respect to a continuous version of the variable k:

$$\Rightarrow \frac{d(P_{aak})}{dk} = \frac{p(1-p)N}{(k(1-2p)+pN)^2} > 0. \tag{16}$$

Specifically, k_{min} is given by the following expression, derived by relating (15) to the parameter Θ:

$$k_{min} = \left\lceil \frac{pN\Theta}{1 - p - (1-2p)\Theta} \right\rceil. \tag{17}$$

□

The first question this previous theorem allows us to answer is how the metrics described in (8)-(14) change for the decision threshold scheme. In this scheme, we have that, if $k \geq k_{min}$ of its neighbors also read the same value a, the node i decides on $R_i = a$. Thus, we can replace P_{aak} in (8)-(14) with a step function U_k, which is 1 for $k \geq k_{min}$ and 0 otherwise. This is equivalent to eliminating the P_{aak} term and summing only terms with $k \geq k_{min}$. Thus, for the decision threshold scheme, we have that:

$$\begin{aligned} P(R_i = a | S_i = a, T_i = a) &= 1 - P(R_i = \neg a | S_i = a, T_i = a) \\ &= \sum_{k=k_{min}}^{N} g_k \end{aligned} \tag{18}$$

$$\begin{aligned} P(R_i = \neg a | S_i = \neg a, T_i = a) &= 1 - P(R_i = a | S_i = \neg a, T_i = a) \\ &= \sum_{k=k_{min}}^{N} g_{N-k} \end{aligned} \tag{19}$$

$$\alpha = \left(1 - \sum_{k=k_{min}}^{N} (g_k - g_{N-k}) \right) n \tag{20}$$

$$\beta = \left(1 - \sum_{k=k_{min}}^{N} g_{N-k}\right) np \tag{21}$$

$$\gamma = \left(\sum_{k=k_{min}}^{N} g_{N-k}\right) np \tag{22}$$

$$\delta = \left(1 - \sum_{k=k_{min}}^{N} g_k\right)(1-p)n. \tag{23}$$

The following is a strong result about the optimal threshold decision scheme.

Theorem 2. *The optimum threshold value which minimizes α, the average number of errors after decoding, is $\Theta^* = (1-p)$. This threshold value corresponds to $k^*_{min} = 0.5N$.*

Proof. As the goal is to find the k_{min} and Θ which minimize α, it is helpful to start with the definition of α. From (23), we have that:

$$\begin{aligned}\alpha &= \left(1 - \sum_{k=k_{min}}^{N} (g_k - g_{N-k})\right) n \\ &= \left(1 - \sum_{k=k_{min}}^{N} \binom{N}{k}\left((1-p)^k p^{(N-k)} - p^k(1-p)^{(N-k)}\right)\right) n.\end{aligned} \tag{24}$$

We examine the behavior of the expression in the summand:

$$\begin{aligned}&\left((1-p)^k p^{(N-k)} - p^k(1-p)^{(N-k)}\right) \\ &= p^k(1-p)^k\left(p^{(N-2k)} - (1-p)^{(N-2k)}\right).\end{aligned} \tag{25}$$

For $p < 0.5$, this expression is negative for $N > 2k$, zero for $N = 2k$, and positive for $N < 2k$. In the expression for α, as we vary k_{min} by decreasing it by one at a time from N, we get additional terms with negative contributions while $k_{min} > 0.5N$ and positive contributions once $k_{min} < 0.5N$. It follows that α achieves a minimum when $k_{min} = k^*_{min} = 0.5N$.

To determine what value of Θ this corresponds to, we can utilize (17). We have that

$$\begin{aligned}&\frac{pN\Theta^*}{1-p-(1-2p)\Theta^*} = 0.5N \\ &\Rightarrow p\Theta^* = 0.5(1-p-(1-2p)\Theta^*) \\ &\Rightarrow \Theta^*(p-p+0.5) = 0.5(1-p) \\ &\Rightarrow \Theta^* = (1-p).\end{aligned} \tag{26}$$

□

The above theorem says that the best policy for each node (in terms of minimizing α, the average number of errors after decoding) is to accept its own sensor reading if and only if at least half of its neighbors have the same reading. This is an intuitive result, following from the equal-weight evidence model that we are using (3). This means that the sensor nodes can perform an optimal decision without even having to estimate the value of p. This makes the optimal-threshold decision scheme presented in Table 1 an extremely feasible mechanism for minimizing the effect of uncorrelated sensor faults.

5 Simulation Results

We conducted some experiments to test the performance of the fault recognition algorithms. The scenario consists of $n = 1,024$ nodes placed in a 32×32 square grid of unit area. The communication radius R determines which neighbors each node can communicate with. R is set to $\frac{1}{\sqrt{n}-1}$ so that each node can only communicate with its immediate neighbor in each cardinal direction. All sensors are binary: They report a "0" to indicate no event and a "1" to indicate that there is an event. The faults are modeled by the uncorrelated, symmetric, Bernoulli random variable. Thus, each node has an independent probability p of reporting a "0" as a "1" or vice versa. We model correlated events by having l single point-sources placed in the area and assuming that all nodes within radius S of each point-source have a ground truth reading of 1, i.e., detect an event if they are not faulty. For the scenario for which the simulation results are presented here, $l = 1, S = 0.15$.

We now describe the simulation results. The most significant way in which the simulations differ from the theoretical analysis that we have presented thus far is that the theoretical analysis ignored edge and boundary effects. This can play a role because, at the edge of the deployed network, the number of neighbors per node is less than that in the interior and, also, the nodes at the edge of an event region are more likely to erroneously determine their reading if their neighbors provide conflicting information. Such boundary nodes are the most likely sites of new errors introduced by the fault recognition algorithms presented above. In general, because of this, we would expect the number of newly introduced errors to be higher than that predicted by the analysis.

Fig. 4 shows a snapshot of the results of a sample simulation run. The sensor nodes are depicted by dots; the nodes indicated with bold dots are part of the circular event region. An "x" indicates a faulty node (before the fault recognition algorithm), while an "o" indicates a node with erroneous readings after fault recognition. Thus, nodes with both an "x" and "o" are nodes whose errors were not corrected, while nodes with an "x" but no "o" are nodes whose errors were corrected and nodes with no "x" but an "o" are nodes where a new error has been introduced by the fault-recognition algorithm. It can be seen that many of the remaining errors are concentrated on the boundaries of the event region on the top right.

Figs. 5, 6, and 7 show the important performance measures for the fault recognition algorithm with the optimal threshold decision scheme from both the simulation as well as the theoretical equations. The key conclusion from these plots is that the simulation matches the theoretical predictions closely in all respects except the statistic of newly introduced errors, where, understandably, the border effects in the simulation result in higher values. More concretely, these figures show that well over 85-95 percent of the total faults can be corrected even when the fault rate is as high as 10 percent of the entire network.

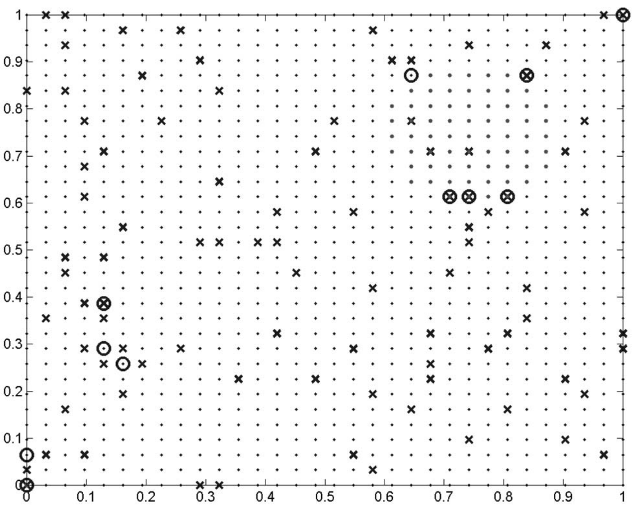


Fig. 4. A snapshot of the simulator showing the errors before and after fault recognition with optimal threshold (p = 0.1).

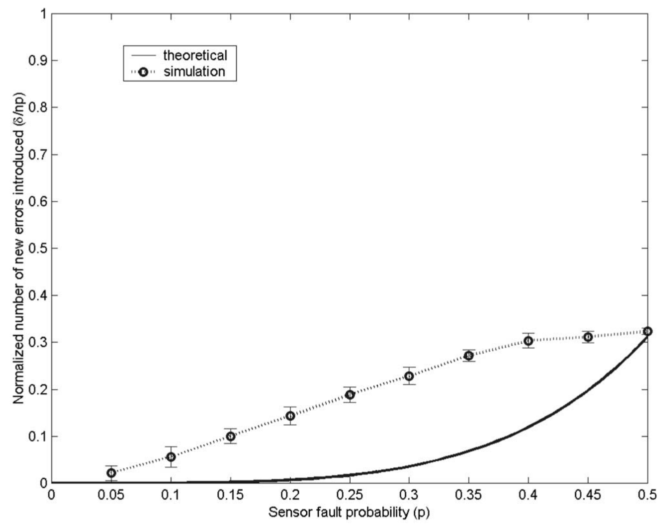


Fig. 6. Normalized number of new errors introduced with the optimal threshold decision scheme.

Fig. 8 illustrates the performance of the threshold decision scheme with respect to the threshold value Θ. Again, the simulation and theoretical predictions are in close agreement. The optimal value of the threshold Θ is indeed found to correspond to a k_{min} of $0.5N$.

As mentioned before, many detection errors occur at the boundary of the event region. This is because the assumption that all neighbors should have the same reading fails, by definition, at this border. Now, the relative number of nodes at the boundary of an event region (compared to the number of nodes within the region) decreases as the size of the event region increases. We should therefore expect to see the algorithm perform better as the event region increases. This is shown by Fig. 9, which measures the improvement in the average fraction of errors corrected in the event region.

Finally, we comment on the impact of the parameter N, the number of neighbors of each node, on the performance of the algorithms we have proposed. There are essentially two ways to increase N—by increasing the sensor density or by increasing the communication range of each node. All other factors remaining the same, increasing N by increasing the deployed density of sensors can significantly improve the performance of the algorithm. This is because this would allow a greater sampling of a spatially correlated event. However, increasing N by keeping the density the same and increasing the communication range can have the opposite effect. Increasing the communication range can effectively increase the number of nodes that are at the "boundary" of the event region—potentially increasing the number of nodes at which incorrect decisions are made by the Bayesian algorithms. Both these effects were observed in our simulations.

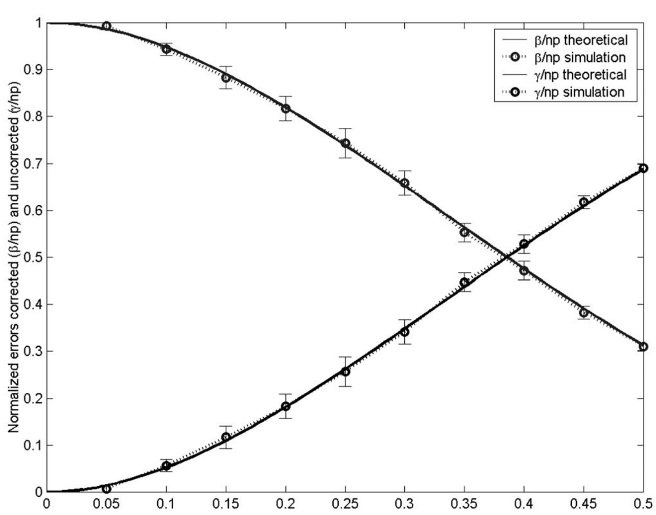


Fig. 5. Normalized number of errors corrected and uncorrected with the optimal threshold decision scheme.

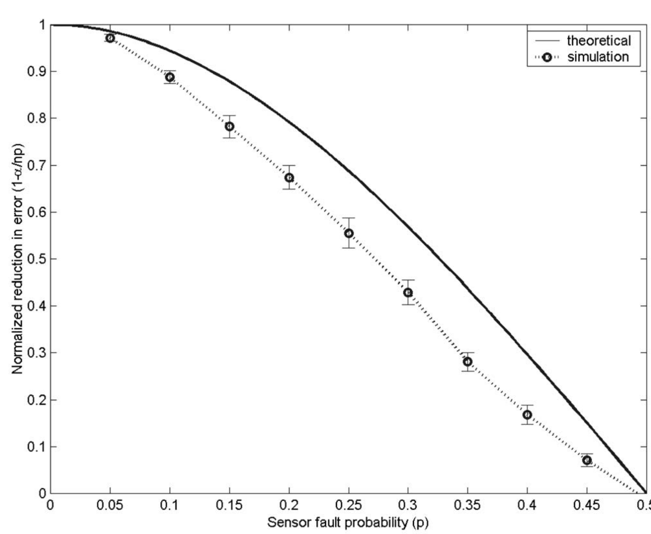


Fig. 7. Normalized reduction in average number of errors for the optimal threshold decision scheme.

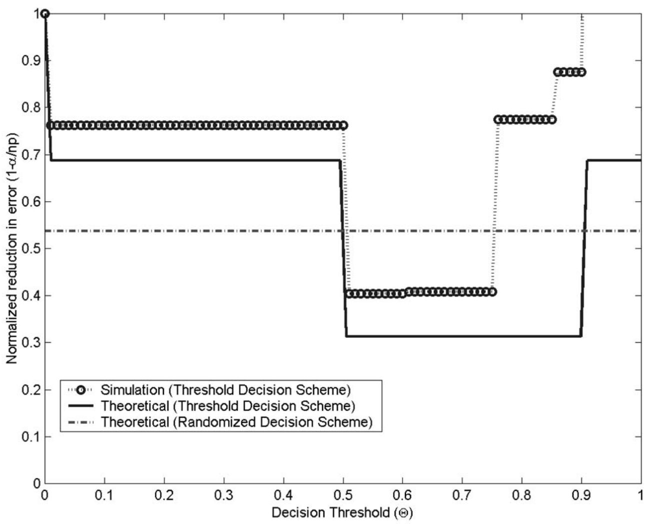


Fig. 8. Normalized reduction in average number of errors with respect to the threshold value in the threshold decision scheme (p = 0.25, $\Theta * = 1 - p = 0.75$).

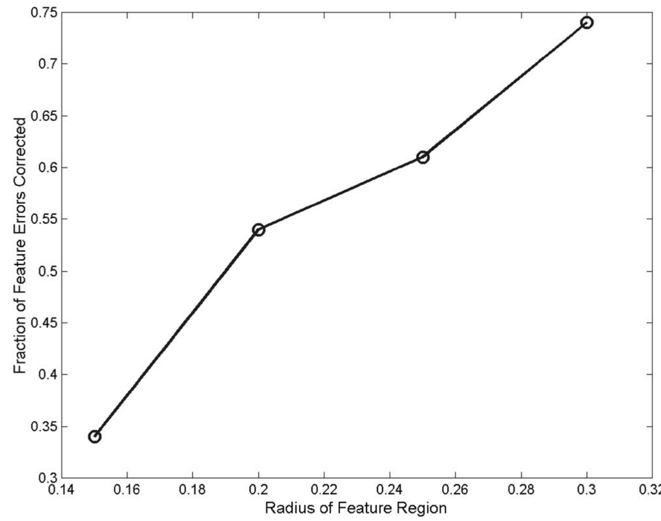


Fig. 9. Average fraction of errors corrected in the event region, with respect to the size of the event region.

6 Conclusions

With recent advances in technology, it has become feasible to consider the deployment of large-scale wireless sensor networks that can provide high-quality environmental monitoring for a range of applications. In this paper, we developed a solution to a canonical task in such networks—the extraction of information about regions in the environment with identifiable events.

One of the most difficult challenges is that of distinguishing between faulty sensor measurements and unusual environmental conditions. To our knowledge, this is the first paper to propose a solution to the fault-event disambiguation problem in sensor networks. Our proposed solution, in the form of Bayesian fault-recognition algorithms, exploits the notion that measurement errors due to faulty equipment are likely to be uncorrelated, while environmental conditions are spatially correlated.

We presented two Bayesian algorithms, the randomized decision scheme and the threshold decision scheme, and derived analytical expressions for their performance. Our analysis showed that the threshold decision scheme has better performance in terms of the minimization of errors. We also derived the optimal setting for the threshold decision scheme for the average-correlation model. The proposed algorithm has the additional advantage of being completely distributed and localized—each node only needs to obtain information from neighboring sensors in order to make its decisions. The theoretical and simulation results show that, with the optimal threshold decision scheme, faults can be reduced by as much as 85 to 95 percent for fault rates as high as 10 percent.

We should note that the extension to nonsymmetric fault probabilities is straightforward and does not affect the basic conclusions of this paper. There are a number of other directions in which this work on fault recognition and fault tolerance in sensor networks can be extended. We have dealt with a binary fault-event disambiguation problem here. This could be generalized to the correction of real-valued sensor measurement errors: Nodes in a sensor network should be able to exploit the spatial correlation of environmental readings to correct for the noise in their readings (the noise models would be different from the binary 0-1 failures considered in this work). Another related direction is to consider dynamic sensor faults where the same nodes need not always be faulty. Much of the work presented here can also be extended to dynamic event region detection to deal with environmental phenomena that change location or shape over time. We would also like to see the algorithms proposed in this paper implemented and validated on real sensor network hardware in the near future.

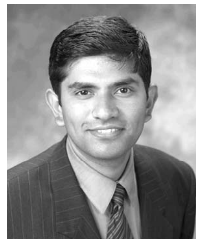
Acknowledgments

This research was funded in part by US Defense Advanced Research Projects Agency N66001-00-1-8946, US Office of Naval Research N00014-01-0712, and the US Department of Energy through Oak Ridge National Lab.

References

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the World with Wireless Sensor Networks," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing (ICASSP 2001),* May 2001.
[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networks (MobiCom 2000),* Aug. 2000.
[3] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," *Proc. 22nd Int'l Conf. Distributed Computing Systems (ICDCS '02),* July 2002.
[4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networks (MobiCom '99),* Aug. 1999.
[5] K. Marzullo, "Implementing Fault-Tolerant Sensors," TR89-997, Dept. of Computer Science, Cornell Univ., May 1989.
[6] L. Prasad, S.S. Iyengar, R.L. Kashyap, and R.N. Madan, "Functional Characterization of Fault Tolerant Interation in Distributed Sensor Networks," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 21, no. 5, Sept./Oct. 1991.
[7] N. Lynch, *Distributed Algorithms.* Morgan Kauffman, 1997.

[8] B.A. Forouzan, *Data Communications and Networking.* McGraw-Hill, 2001.
[9] A. Cerpa et al., "Habitat Monitoring: Application Driver for Wireless Communications Technology," *Proc. 2001 ACM SIGCOMM Workshop Data Comm. in Latin America and the Caribbean,* Apr. 2001.
[10] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies," *Proc. INFOCOM,* 2002.
[11] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building Efficient Wireless Sensor Networks with Low-Level Naming," *Proc. 18th ACM Symp. Operating Systems Principles,* Oct. 2001.
[12] W.W. Manges, "Wireless Sensor Network Topologies," *Sensors Magazine,* vol. 17, no. 5, May 2000.
[13] G.J. Pottie and W.J. Kaiser, "Wireless Integrated Network Sensors," *Comm. ACM,* vol. 43, no. 5, pp. 551-558, May 2000.
[14] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad-Hoc Networks," *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networks (MOBICOM '98),* pp. 76-84, Oct. 1999.
[15] J. Warrior, "Smart Sensor Networks of the Future," *Sensors Magazine,* Mar. 1997.
[16] K. Chakrabarty, S.S. Iyengar, H. Qi, and E.C. Cho, "Grid Coverage of Surveillance and Target Location in Distributed Sensor Networks," *IEEE Trans. Computers,* vol. 51, no. 12, pp. 1448-1453, Dec. 2002.
[17] S.S. Iyengar, M.B. Sharma, and R.L. Kashyap, "Information Routing and Reliability Issues in Distributed Sensor Networks," *IEEE Trans. Signal Processing,* vol. 40, no. 2, pp. 3012-3021, Dec. 1992.
[18] S.S. Iyengar, D.N. Jayasimha, and D. Nadig, "A Versatile Architecture for the Distributed Sensor Integration Problem," *IEEE Trans. Computers,* vol. 43, no. 2, Feb. 1994.
[19] L. Prasad, S.S. Iyengar, R.L. Rao, and R.L. Kashyap, "Fault-Tolerant Sensor Integration Using Multiresolution Decomposition," *Physical Rev. E,* vol. 49, no. 4, Apr. 1994.
[20] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Personal Comm.,* vol. 7, no. 5, pp. 16-27, Oct. 2000.
[21] M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks," *Proc. Int'l J. High Performance Computing Applications,* 2002.
[22] P. Bonnet, J.E. Gehrke, and P. Seshadri, "Querying the Physical World," *IEEE Personal Comm.,* vol. 7, no. 5, Oct. 2000.
[23] B. Krishnamachari, R. Bejar, and S.B. Wicker, "Distributed Problem Solving and the Boundaries of Self-Configuration in Multi-Hop Wireless Networks," *Proc. Hawaii Int'l Conf. System Sciences (HICSS-35),* Jan. 2002.
[24] R. Min et al., "An Architecture for a Power-Aware Distributed Microsensor Node," *Proc. IEEE Workshop Signal Processing Systems (SiPS '00),* Oct. 2000.
[25] S. Chessa and P. Santi, "Crash Faults Identification in Wireless Sensor Networks," *Computer Comm.,* vol. 25, no. 14, pp. 1273-1282, Sept. 2002.
[26] D. Estrin et al., "Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers," Nat'l Research Council Report, 2001.
[27] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm.,* vol. 7, no. 5, pp. 28-34, Oct. 2000.
[28] R.A. Maxion, "Toward Diagnosis as an Emergent Behavior in a Network Ecosystem," *Emergent Computation,* S. Forrest, ed. MIT Press, 1991.
[29] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. INFOCOM,* 1997.
[30] N. Megiddo, "Linear Time Algorithm for Linear Programming in R3 and Related Problems," *SIAM J. Computers,* vol. 12, no. 4, pp. 759-776, 1983.
[31] M.E. Dyer, "Linear Time Algorithms for Two and Three-Variable Linear Programs," *SIAM J. Computers,* vol. 13, no. 1, pp. 31-45, 1984.
[32] B. Cain, T. Speakman, and D. Towsley, "Generic Router Assist (GRA) Building Block Motivation and Architecture," RMT Working Group, Internet-Draft < draft-ietf-rmt-gra-arch-01.txt> , work in progress, Mar. 2000.
[33] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, and G.J. Minden, "A Survey of Active Network Research," *IEEE Comm. Magazine,* vol. 35, no. 1, pp. 80-86, Jan. 1997.
[34] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Comm. Magazine,* vol. 40, no. 8, pp. 102-114, Aug. 2002.
[35] D.K. Gifford, "Weighted Voting for Replicated Data," *Proc. ACM Symp. Operating Systems Principles,* 1979.
[36] R.H. Thomas, "A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases," *ACM Trans. Database Systems,* vol. 4, pp. 180-209, 1979.
[37] L.A. Klein, *Sensor and Data Fusion Concepts and Applications.* SPIE, Apr. 1993.



Bhaskar Krishnamachari received the Bachelors degree in electrical engineering with a four-year full-tuition scholarship from The Cooper Union for the Advancement of Science and Art in 1998. He received the Master's degree and the PhD degree in electrical engineering from Cornell University in 1999 and 2002, under a four-year university graduate fellowship. Since Fall 2002, he has been an assistant professor in the Department of Electrical Engineering at the University of Southern California (USC) and holds a joint appointment in the Department of Computer Science. His previous research has included work on critical density thresholds in wireless networks, data centric routing in sensor networks, mobility management in cellular telephone systems, multicast flow control, heuristic global optimization, and constraint satisfaction. His current research is focused on the discovery of fundamental principles and the analysis and design of protocols for next generation wireless sensor networks. He is a member of the IEEE.

Sitharama Iyengar (IEEE Fellow, ACM Fellow, AAAS Fellow) has been involved with research in high-performance algorithms, data structures, sensor fusion, data mining, and intelligent systems since receiving the PhD degree (in 1974 from Mississippi State University) and the MS degree from the Indian Institute of Science (1970). He is the chairman and Roy Paul Daniels Chaired Professor of Computer Science at Louisiana State University and is also Satish Dhawan Chaired Professor at the Indian Institute of Science. His publications include 13 widely used textbooks and more than 280 research papers in refereed journals and conferences. He has been a visiting professor at the Jet Propulsion Laboratory-California Institute of Technology, Oak Ridge National Laboratory, the Indian Institute of Science, and the University of Paris. He has served as associate editor and as guest editor for several journals including the *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Systems, Man, and Cybernetics*, *IEEE Transactions on Software Engineering*, *Journal of Theoretical Computer Science*, *Journal of Computer and Electrical Engineering*, *Journal of the Franklin Institute*, *Journal of American Society of Information Science*, and the *International Journal of High Performance Computing Applications*. He is a series editor for Neuro-Computing of Complex Systems for CRC Press. In 1998, he won the IEEE Computer Society Technical Achievement Award for outstanding contributions to data structures and algorithms in image processing and sensor fusion problems. He was awarded the Distinguished Alumnus Award at the Indian Institute of Science in March 2003.