# Channel Aware Decentralized Detection in Wireless Sensor Networks

Milad Kharratzadeh
milad.kharratzadeh@mail.mcgill.ca
Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

***Abstract*—We study the problem of decentralized detection in wireless sensor networks. First, we will review the classical framework for decentralized detection. The classical framework does not take into account the power constraints of wireless sensor networks and also the characteristics of wireless channels such as fading and noise. Next, we will review the algorithms that consider these constraints. We assume that these algorithms know the channel fading and noise statistics, and hence the term channel-aware. Two main approaches are approximating the fusion rule based on the value of SNR, and sensor censoring, where not all the peripheral nodes send their decisions to the central node. In the end, we will propose a fully decentralized channel aware algorithm for decentralized detection which is based on gossiping. We will show the validity of the proposed algorithm by proving the convergence and also performing some simulations.**

## I. Introduction

Wireless Sensor Networks consist of a set of sensors, which are capable of sensing, computation, and communication and are spatially distributed in order to cooperatively monitor physical or environmental conditions. The ability to detect events of interest is essential to the success of emerging sensor network technology. Detection often serves as the initial goal of a sensing system. For example in applications where we want to estimate attributes such as position and velocity, the first step is to ascertain the presence of an object. Moreover, in some applications such as surveillance, the detection of an intruder is the only purpose of the sensor system. In situations where the peripheral nodes do some preprocessing on their observation before sending data to a central node (called fusion center), the corresponding decision making problem is termed decentralized detection.

Assume that there are $M$ hypotheses on the state of the environment and each one of the sensors observes some relevant information about it. In a centralized scheme, each sensor transmits all of its observation (without any additional processing) to the fusion center, which solves a classical hypotheses testing problem and decides on one of the $M$ hypotheses. However, in a classical decentralized scheme, each sensor will do some preprocessing and send a summary of its observation, which is chosen from a finite alphabet, to the fusion center. Then, the fusion center will decide on one of the $M$ hypotheses, based on the messages it has received.

The described centralized and decentralized schemes, differ in some aspects. First, it is clear that the performance of the decentralized scheme is suboptimal in comparison with the centralized scheme due to loss of information in the nodes local preprocesses. On the other hand the communication requirements of decentralized scheme is much smaller than those of centralized one. The reason is that instead of sending raw voluminous observation, each node sends a summary of its observation, taken from a finite alphabet. So, in brief, the decentralized detection offers a great reduction in communication requirements, at the expense of some performance reduction. However, it turns out that the performance reduction is often negligible [1].

In the decentralized scheme, while the fusion center faces a classical hypotheses testing problem (if we look at the messages received from other nodes as its observations), the problem is more complex for the peripheral sensors. One may expect that each sensor should decide independently and make decision only based on its own observation and use its own likelihood ratio test. This is not true in general. When the detectors decide in a way to achieve a system-wide optimization, they often use different strategies than in cases where the joint costs of their decisions separates into a cost for each. Even under a conditional independence assumption (which means that the observations of different sensors are independent from each other under the same hypothesis), finding optimal decision-making algorithms (based on the observation) at the sensor nodes remains, in most cases, a difficult task. This optimization problem is known to be tractable only under restrictive assumptions regarding the observation space and the topology of the underlying network [1].

In section II we will explain the problem formulation and briefly review the works that study the classical framework [1], [2], [3]. Next, in section III we will review [4] and [5] where some channel aware algorithms for decentralized detection have been introduced. To be specific, the optimum likelihood ratio has been approximated in low and high SNR cases there. Section IV is devoted to the review of sensor censoring which is one of the main approaches in the litearture for interacting with channel constraints [6], [7]. Sensor censoring is a scheme in which, not all the nodes send their decisions to the fusion center and only the sensors with "informative" observations will communicate with the central node. Finally, as the original part of this work, we will propose a channel aware algorithm for decentralized detection in section V which is based on gossiping. We will prove the convergence of the algorithm in the absence of noise and fading and also show the validity of the proposed algorithm with channel constraints through simulations.

## II. Problem Definition and Classical Framework

Here, we will focus on binary hypotheses testing problem which is stated as follows. There are $N$ peripheral sensors each of which sends its message ("summary" of its observation which is chosen from a binary alphabet) to the fusion center directly (i.e. parallel or star topology). The two hypotheses are $H_0$ and $H_1$ with prior probabilities $\Pr(H_0)$ and $\Pr(H_1)$, respectively. Each of these hypotheses induces a different joint probability distribution for the observations of peripheral sensors:

$$\text{Under } H_1 \quad : \quad \Pr(r_i|H_1) \tag{1}$$

$$\text{Under } H_0 \quad : \quad \Pr(r_i|H_0) \tag{2}$$

where $r_i,\ i = 1, \ldots, N$ are the observations of peripheral nodes. Each sensor receives an observation $r_i$, which is a realization of a random variable whose pdf is known from (1) and (2), and evaluates a message $u_i = \gamma_i(r_i)$, and transmits it to the fusion center. We call the function $\gamma_i$, as the decision rule of the sensor $i$. The fusion center receives all these messages and using its own decision rule (which is also called fusion rule), $\gamma_0$, decides on one of the two possible hypotheses. The set of all these decision rules is called strategy.

The classical framework, which has been extensively explored (e.g., see [1], [2], [3]), does not adequately take into account important features of the wireless sensor networks such as nodes power constraints and limits on the channel capacity. Also, it is assumed that the sent messages from the peripheral nodes will be received by the fusion center correctly, hence ignoring noise, intereference, and fading in communication channels. In this section we will briefly review the results of the classical framework, and in the following sections, we will study the effect of the mentioned constraints. Here, we will review the results from [1], [3].

In the Bayesian formulation, we are given a cost function

$$C(U_0, U_1, \ldots, U_N, H) \tag{3}$$

where $C(u_0, u_1, \ldots, u_N, H_i)$ is the cost associated with the event that hypothesis $H_i,\ i = 0, 1$ is true, the messages from the peripheral sensors are $u_1, \ldots, u_N$, and the fusion center decision is $u_0$. Our objective in Bayesian framework is to find a strategy, $\gamma$ that minimizes the expectation of the cost function: $J(\gamma) \triangleq E[C(U_0, U_1, \ldots, U_N, H)]$.

Note that the cost function defined above is a function of the fusion decision as well as the peripheral nodes messages. However, for different purposes we can define different cost functions that only depend on some of decisions. For example, if $C$ only depends on $U_0$, the performance of the system is judged on the basis of the fusion center's decision. The minimum probability of error criterion for the final decision lies in this type of cost functions. As another case, we may wish to interpret each sensor's message as a local decision, based on the true hypothesis. Then, with the suitable choice of $C$, we can penalize incorrect decisions by the fusion center as well as the peripheral sensors. As an extreme case, the cost function might be independent of the fusion center's decision and be only a function of the decisions of peripheral nodes and we only need to optimize with respect to $\gamma_i, i = 1, \ldots, N$. This can happen in the case of a priori fixed fusion decision rule.

The main result in [1] is the following theorem, which we include here without proof:

**Theorem 1.** *(a) Fix some $i \neq 0$ and suppose that $\gamma_j$ has been fixed for all $j \neq i$. Then $\gamma_i$ minimizes $J(\gamma)$ if and only if*

$$\gamma_i(r_i) = \operatorname*{argmin}_{d=0,1} \sum_{j=0}^{1} \Pr(H_j|r_i) \cdot a_i(H_j, d) \quad \text{w.p. } 1 \tag{4}$$

*where*

$$\begin{aligned} a_i(H_j, d) = E[C(\gamma_0(U_1, \ldots, U_{i-1}, d, U_{i+1}, \ldots, U_N) \\ , U_1, \ldots, U_{i-1}, d, U_{i+1}, \ldots, U_N, H_j)|H_j] \end{aligned} \tag{5}$$

*(b) Suppose that the decision rules for peripheral nodes have been fixed, then $\gamma_0$ minimizes $J(\gamma)$ if and only if*

$$\begin{aligned} \gamma_0(U_1, \ldots, U_N) = \operatorname*{argmin}_{d=0,1} \sum_{j=0}^{1} \Pr(H_j|U_1, \ldots, U_N) \\ \times C(d, U_1, \ldots, U_N, H_j) \quad \text{w.p. } 1 \end{aligned} \tag{6}$$

This means that with person-by-person optimization (fixing all the sensors decision rules except one, and try to minimize $J$ with respect to that sensor decision rule) we will get a set of likelihood ratio tests whose thresholds depend on each other. Thus, in order to find the person-by-person optimal strategy, we need to solve a system of $4N$ nonlinear equations in as many unknowns. Therefore, the achieved decision rule is equivalent to dividing the observations space into two regions (because we are considering binary hypotheses testing), and deciding to send $u_i = d,\ d = 0$ or $1$ to the fusion center if the vector of likelihood belongs to the corresponding region. Each region is specified by a set of linear equalities and therefore is a polyhedral. Note that this structure is the same as the structure of the optimal decision rule for the classical binary hypotheses testing.

The Neyman-Pearson formulation of the problem has been studied in [3] and similar results have been derived, that we do not mention them here. In the next section we will review the extension of the classical parallel fusion structure, by incorporating the fading channel layer that is present in wireless sensor networks.

## III. Decision Fusion Under Fading Channel Assumption

In this section, we will review [4] and [5] which explore decision fusion algorithms that take into account channel fading effects. Here we assume that we have flat fading channels and the fusion center knows the fading coefficient (envelope), and hence it is channel-aware. Although it is always possible to reduce the effects of noise and fading by increasing SNR in channel aware systems, in wireless sensor networks it is desirable to reduce the communication power (because WSNs are energy constrained). The problem formulation is almost similar to the previous section, except for the fact that the messages sent from the peripheral sensors to the fusion center pass through channels wirh flat fading and are corrupted by noise (see Fig. 1). Here, we do not look at the peripheral sensors decision rules and only modify the fusion center decision to cope with noise and fading.

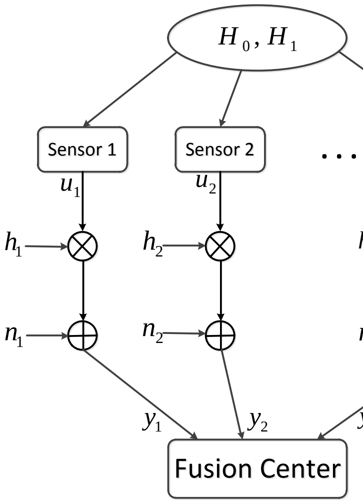

Fig. 1. Prallel fusion model in the presence of fading and noisy channel

Assume that the $k$th local sensor makes a binary decision $u_i \in \{+1, -1\}$ with $P_{fi} = \Pr(u_i = 1|H_0), P_{di} = \Pr(u_i = 1|H_1)$, called false alarm and detection probabilities respectively. If $u_i$ is transmitted from sensor $i$ through a fading channel, then the output of the channel (input to the fusion center) would be:

$$y_i = h_i u_i + n_i \tag{7}$$

where $h_i$ is the real valued (positive) fading coefficient (envelope) of the channel between sensor $i$ and the fusion center. $n_i$ is zero mean Gaussian noise with variance $\sigma^2$. In the following, we try to derive a fusion rule based on $y_i,\ i = 1, \ldots, N$, that can determine which hypothesis is true with the best achievable performance.

Assuming full knowledge of the fading channels, $h_i$, and the local sensors performance, $P_{fi}$ and $P_{di}$, we can write the likelihood ratio at the fusion center as:

$$\begin{aligned}\Lambda(\mathbf{y}) &= \frac{f(\mathbf{y}|H_1)}{f(\mathbf{y}|H_0)} \\ &= \prod_{i=1}^{N} \frac{P_{di} e^{-\frac{(y_i - h_i)^2}{2\sigma^2}} + (1 - P_{di}) e^{-\frac{(y_i + h_i)^2}{2\sigma^2}}}{P_{fi} e^{-\frac{(y_i - h_i)^2}{2\sigma^2}} + (1 - P_{fi}) e^{-\frac{(y_i + h_i)^2}{2\sigma^2}}} \end{aligned} \tag{8}$$

where $\mathbf{y}$ is the vector of received messages from all the sensors at the fusion center. Now, we try to simplify the above expression in two special cases: high SNR (i.e., $\sigma^2 \to 0$) and low SNR (i.e., $\sigma^2 \to \infty$).

*1) High SNR ($\sigma^2 \to 0$):* In this case the authors proposed a two-stage approximation in [4]. The likelihood ratio described in (8) considers the channel effects and the sensors specifications simultaneously. An alternative is to separate this into a two stage procedure. First, infer about $u_i$ using $y_i$, and then apply the optimum fusion rule based on the achieved value of $u_i$. Given the model introduced in (7), the maximum likelihood estimation of $u_i$ is:

$$\widetilde{u}_i = \text{sign}(y_i)$$

Now, if we rewrite (8), we have:

$$\begin{aligned}\Lambda = &\prod_{\text{sign}(y_i)=1} \frac{P_{di} e^{\frac{2y_i h_i}{\sigma^2}} + (1 - P_{di})}{P_{fi} e^{\frac{2y_i h_i}{\sigma^2}} + (1 - P_{fi})} \\ \times &\prod_{\text{sign}(y_i)=-1} \frac{P_{di} + (1 - P_{di}) e^{-\frac{2y_i h_i}{\sigma^2}}}{P_{fi} + (1 - P_{fi}) e^{-\frac{2y_i h_i}{\sigma^2}}}\end{aligned}$$

Because of the high SNR assumption ($\sigma^2 \to 0$), we have:

$$\lim_{\sigma^2 \to 0} \Lambda = \prod_{\text{sign}(y_i)=1} \frac{P_{di}}{P_{fi}} \prod_{\text{sign}(y_i)=-1} \frac{1 - P_{di}}{1 - P_{fi}}$$

In above derivation we used the fact that as $\sigma^2$ goes to zero, the exponential terms with a positive power over $\sigma^2$ will go to infinity and we can ignore other terms that are added to them. Taking logarithms from both sides, we will achieve the following simplified log-likelihood ratio:

$$\Lambda_1 = \lim_{\sigma^2 \to 0} \log \Lambda = \sum_{\text{sign}(y_i)=1} \log \left[\frac{P_{di}}{P_{fi}}\right] + \sum_{\text{sign}(y_i)=-1} \log \left[\frac{1 - P_{di}}{1 - P_{fi}}\right]$$

Note that $\Lambda_1$ does not depend on the channel statistics at all and is only a function of $P_{di}$ and $P_{fi}$ for $i = 1, \ldots, N$.

*2) Low SNR ($\sigma^2 \to \infty$):* The high SNR assumption is not realistic, because of the power constraint of the wireless sensor networks. So, in this case we consider the more realistic case of low SNR. Rewriting (8), we have:

$$\prod_{i=1}^{N} \frac{P_{di} + (1 - P_{di}) e^{-\frac{2y_i h_i}{\sigma^2}}}{P_{fi} + (1 - P_{fi}) e^{-\frac{2y_i h_i}{\sigma^2}}}$$

For $\sigma^2 \to \infty$ we have $e^{-(2y_i h_i)/\sigma^2} \to 1$ and thus we can approximate it by the first-order Taylor series expansion:

$$e^{-(2y_i h_i)/\sigma^2} \simeq 1 - \frac{2y_i h_i}{\sigma^2}$$

Therefore:

$$\begin{aligned}\lim_{\sigma^2 \to 0} \Lambda &= \lim_{\sigma^2 \to 0} \prod_{i=1}^{N} \frac{P_{di} + (1 - P_{di})(1 - \frac{2y_i h_i}{\sigma^2})}{P_{fi} + (1 - P_{fi})(1 - \frac{2y_i h_i}{\sigma^2})} \\ &= \lim_{\sigma^2 \to 0} \prod_{i=1}^{N} \frac{1 - (1 - P_{di}) \frac{2y_i h_i}{\sigma^2}}{1 - (1 - P_{fi}) \frac{2y_i h_i}{\sigma^2}}\end{aligned}$$

Taking logarithm from both sides:

$$\begin{aligned}\lim_{\sigma^2 \to 0} \log \Lambda = &\lim_{\sigma^2 \to 0} \sum_{i=1}^{N} \log \left[1 - (1 - P_{di}) \frac{2y_i h_i}{\sigma^2}\right] \\ &- \lim_{\sigma^2 \to 0} \sum_{i=1}^{N} \log \left[1 - (1 - P_{fi}) \frac{2y_i h_i}{\sigma^2}\right]\end{aligned}$$

Using the approximation $\log(1+x) \simeq x$ for $x \to 0$, and simplifying, we have:

$$\lim_{\sigma^2 \to 0} \log \Lambda = \sum_{i=1}^{N} (P_{di} - P_{fi}) \frac{2 y_i h_i}{\sigma^2}$$

If we assume that $P_{di} - P_{fi}$ is constant for all sensors, and eliminate it from the expression as well as the constant term $\sigma^2$, and also multiplying by a constant term $1/K$ we will achieve a likelihood ratio at the fusion center which is analogous to maximum ratio combiner (MRC):

$$\Lambda_2 = \frac{1}{K} \sum_{i=1}^{N} h_i y_i$$

In fact, $\Lambda_2$ is the first order approximation of the log-likelihood ratio based fusion rule and is asymptotically accurate for low SNR ($\sigma^2 \to \infty$). Note that the achieved expression for $\Lambda_2$ does not depend on the sensor specifications and is only a function of the channel statistics.

The validity of the above approximations and performance evaluation have been studied in [4], which we skip here. The same authors, also study the same problem for multi-hop wireless sensor networks in [5] and found similar approximations for the fusion rule. The derivations of formulas are similar to the ones reviewed above and thus we do not repeat them here. In the setup we reviewed in this section, we assumed that all the peripheral nodes send their decisions to the fusion center. In the next section we will review another approach used in literature for interacting with channel constraints which is called sensor censoring.

## IV. Sensor Censoring

Here we briefly review the sensor censoring approach. We just review the general concepts and do not go into details. The idea of censoring sensors was introduced by Rago, et al. in [6]. In this scheme, sensors are assumed to censor their observation so that each sensor sends to the fusion center only "informative" observations. How should we define informative to minimize the probabilty of error was studied in [6] and it was shown that with conditionally independent observations, the sensors should send their decisions to fusion center if and only if their local likelihood ratio do not fall into a certain single interval. The exact expressions for the intervals for different frameworks such as Neyman-Pearson and Bayesian are derived in [6]. The important fact is that these intervals depend on the available communication rates, so, when the available rate is low, the intervals that defines the "no-send" regions are larger, and only a few sensors will transmit to the fusion center. Also, it was shown, through experiments, that the performance of censoring scheme is very close to optimal even with quite severe communication rates.

The direct consequence of censoring is the saving in communication. Also, it will help the sensors save energy by not sending "uninformative" messages. Fusion of censored decisions transmitted over fading channels in wireless sensor networks is studied in [7], [8]. In these works it is assumed that the local sensors employ a sensor censoring scheme with known thresholds and the main focus is on developement of fusion algorithms while the channels have flat fading. Both cases, one assuming the knowledge of channel fading envelope and the other the fading statistics, are studied and the optimal likelihood ratio test is derived under each scenario.

In summary, to save power and also decrease communication rate, peripheral sensors can employ censoring schemes so that they do not transmit to fusion center if their likelihood ratio is in a certain interval [6]. Then, knowing these intervals and also the information about the channel, the fusion center can use its modified likelihood ratio test [7], [8] to decide on the true hypothesis. It is shown that the performance of this scheme is close to optimal.

## V. Gossip-Based Algorithm

In this section, we are going to propose a gossip-based fully decentralized detection algorithm. To the best of our knowledge, the proposed methods is novel. The algorithms that we studied until now, were mostly semi-decentralized, in the sense that all the peripheral sensors transmit a quantized function of their observations to a central node (fusion center). So, it is clear that we still have the problems of centralized systems such as single point of failure, data management (flooding more data to fusion center than it can process), and security. Also, in practical wireless sensor networks, because of the power constraints, nodes are only capable of short-range communications. Thus, each node can communicate with only a few other nodes that are close to it.

Our proposed gossip-based algorithm, tries to reach a consensus among all the nodes by only *local communications*. First, we will propose our method for the ideal channels between the nodes (i.e., no fading or noise) and show that it converges and the solution is exactly the same as the solution of the centralized scheme, and hence globally optimum. Then, we will modify it for the case where there is noise and channels between nodes have flat fading and show by simulation that our algorithm still converges and find the optimum solution.

Assume that we have a wireless sensor network, with $N$ sensors, and without any central node. Also, assume that we have a binary hypotheses testing problem. Sensor $S_i$ can communicates only with a few other nodes that are in its communication range. We call these nodes the neighbors of sensor $S_i$ and denote them by the set $V_i$. The notation introduced in section II still holds. Our proposed algorithm for fully decentralized detection is briefly as follows:

1- Each sensor receives an observation and based on that, computes its likelihood ratio. Let us denote this likelihood ratio by $\Lambda^{(0)}$:

$$\Lambda_i^{(0)} = \frac{f(r_i|H_1)}{f(r_i|H_0)}, \qquad i = 1, \ldots, N$$

2- The sensors make initial decisions based on their observations. Note that they make their decisions under either Bayesian or Neyman-Pearson criterion. Also, note that, since we have a binary hypothesis testing, under either of the frameworks, all the nodes implement a likelihood ratio test.

3- **Gossiping:** After calculating the likelihood ratios and making initial decisions, the nodes perform several rounds of gossiping. In each round, the nodes communicate two-by-two and update their likelihood ratios. If the nodes $i$ and $j$ communicate with each other (or "gossip") at iteration $k$, then, the update procedure is as follows:

$$\Lambda_i^{(k)} = \sqrt{\Lambda_i^{(k-1)} \cdot \Lambda_j^{(k-1)}} \tag{9}$$

$$\Lambda_j^{(k)} = \sqrt{\Lambda_i^{(k-1)} \cdot \Lambda_j^{(k-1)}} \tag{10}$$

where $\Lambda_i^{(k)}$ denotes the value of the likelihood ratio of sensor $S_i$ at iteration $k$ of the algorithm ($0 \leqslant k$).

4- Based on their new likelihood ratios, the sensors update their decisions. As mentioned in step 2, all the nodes implement a likelihood ratio test with fixed thresholds. In our algorithm, the likelihood ratios are updated and the thresholds of the tests are remained fixed and also are the same for all the sensors ($\lambda^{1/N}$).

5- Steps 3 and 4 are repeated until all the sensors decide on the same hypothesis (denoted as "convergence").

In the following, first, we will show that the mentioned algorithm will converge to the same decision as the centralized scheme in the case where there is no noise or fading. Then we will modify our algorithm for the case where there exists noise and channels have flat fading.

### A. Analysis of Proposed Algorithm

Having $N$ conditionally independent observations $(r_1, \ldots, r_N)$, and assuming a Bayes cost or Neyman-Pearson performance criterion, the optimum (centralized) test for the binary hypotheses testing is well known to be:

$$\prod_{i=1}^{N} \Lambda(r_i) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda \tag{11}$$

where in NP framework, $\lambda$ can be found based on the probability of error criterion, and in Bayesian framework it can be found based on the prior probabilities of the two hypotheses and the costs. By optimum decision, we mean a decision which minimizes the probability of the error.

In a fully decentralized sensor network, each node is only aware of its own likelihood ratio and can perform a test like:

$$\Lambda_i(r_i) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda_i$$

In our method we have equal thresholds for all the nodes, and choose them in a way such that their product is equal to the appropriate $\lambda$ in (11). Therefore, $\lambda_i = \lambda^{1/N}, \quad i = 1, \ldots, N$. These thresholds remain fixed through the algorithm. However, each node updates its likelihood ratio in each iteration of the algorithm. The procedure is as follows: in each iteration, nodes gossip two-by-two with each other and update their likelihood ratios as described in (9) and (10). Note that although the individual likelihood ratios change, the product of the likelihood ratios of all nodes remain fixed through the algorithm. In other words:

$$\prod_{i=1}^{N} \Lambda_i^{(k)} = \prod_{i=1}^{N} \Lambda_i^{(k-1)}, \quad 1 \leqslant k \tag{12}$$

It is clear that only $\Lambda_i^{(0)}$ has the classical definition of the likelihood ratio for sensor $S_i$, and hence equals to $\frac{f(y_i|H_1)}{f(y_i|H_0)}$. However, we still call the updated versions, $\Lambda_i^{(k)}$for $1 \leqslant k$, likelihood ratios for ease of notation.

Then, based on the updated versions of likelihood ratios, the nodes revise their decisions. The algorithm will converge if all the nodes have the same decisions. Assume that at stage $k$ all the nodes have the same decisions. Without loss of generality, assume that they all decide on $H_1$. Thus:

$$\Lambda_i^{(k)} > \lambda_i = \lambda^{1/N}, \quad i = 1, \ldots, N$$

Multiplying all the inequalities, we have:

$$\prod_{i=1}^{N} \Lambda_i^{(k)} > \lambda$$

$$\Rightarrow \prod_{i=1}^{N} \Lambda_i^{(0)} > \lambda \qquad (\text{using } (12))$$

Which shows that $H_1$ is also the solution of the centralized test, and hence optimum (minimizes the probability of error). Thus, if all the nodes "agree" with each other and reach a consensus decision, then their decision matches the centralized decision exactly. However, a big question remains; Does this algorithm converge or not?

Note that, in order to prove that the mentioned algorithm converges, it is enough to show that

$$\text{as } \; k \to \infty : \quad \Lambda_i^{(k)} \to \left[ \prod_{j=1}^{N} \Lambda_j^{(0)} \right]^{1/N}, \quad i = 1, \ldots, N \tag{13}$$

The reason is that in this case, all the likelihood ratios become equal. Also, we know that the thresholds are equal by definition. Therefore, all the nodes perform the same likelihood ratio test and thus get the same result (convergence). The type of convergence that we are concerned about in (13), is convergence in expectation. In the following theorem, we prove that (13) is true and hence, our proposed algorithm converges.

**Theorem 2.** *Assume that we have $N$ nodes with initial values $\Lambda_i^{(0)}, \quad i = 1, \ldots, N$. Consider the following gossip algorithm: At each iteration, nodes choose one of their neighbors at random and gossip with each other. When two nodes gossip with each other, they update their values according to the following procedure:*

$$\Lambda_i^{(k)} = \sqrt{\Lambda_i^{(k-1)} \cdot \Lambda_j^{(k-1)}} \tag{14}$$

$$\Lambda_j^{(k)} = \sqrt{\Lambda_i^{(k-1)} \cdot \Lambda_j^{(k-1)}} \tag{15}$$

*where $\Lambda_i^{(k)}$ denotes the value of the likelihood ratio of sensor $S_i$ at iteration $k$ of the algorithm ($0 \leqslant k$). This gossip algorithm converges in expectation and we have:*

$$\lim_{k \to \infty} E\left[\Lambda_i^{(k)}\right] = \left[ \prod_{j=1}^{N} \Lambda_j^{(0)} \right]^{1/N}, \quad i = 1, \ldots, N \tag{16}$$

The proof can be found in Appendix A.

Another important question is: How fast does the algorithm converge? *or* How many messages do we need to send among sensors in order to converge? This can highly affect the energy consumption of the sensors (the more transmitted messages, the more battery consumption), and also the delay of the system. Actually there are many fast gossip algorithms that we can employ. For example, the algorithm proposed in [9], achieves $\epsilon$-accuracy with high probability after $O\big(n \log\log n \log \frac{kn}{\epsilon}\big)$ messages.

### B. Considering Channel Constraints

Until now, we did not consider the effects of noise and fading in our algorithm. Now, we are going to consider these effects. Since, the nodes are communicating only two-by-two, we only need to consider the following case: Assume that nodes $i$ and $j$ communicate with each other at iteration $k$ over a channel with flat fading and additive white Gaussian noise (see Fig. 2). Also assume that node $i$ sends its likelihood ratio, $\Lambda_i^{(k)}$, to node $j$. If we name the message received at sensor $j$ from sensor $i$ as $y_{ij}^{(k)}$, then we can write:

$$y_{ij}^{(k)} = h_{ij}\Lambda_i^{(k)} + n$$

where $h_{ij}$ is the fading coefficient (envelope) of the channel between sensors $i$ and $j$, and $n$ is Gaussian noise with zero mean and variance $\sigma^2$. Upon receiving $y_{ij}^{(k)}$, sensor $j$ tries to estimate $\Lambda_i^{(k)}$. Since $h_{ij}$ is known to sensor $j$ (cahnnel aware), the maximum likelihood of $\Lambda_i^{(k)}$ is simply:

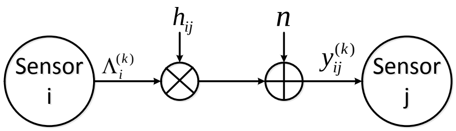

Fig. 2. Gossiping with channel constraints (AWGN and flat fading channel)

$$\widetilde{\Lambda}_i^{(k)} = \frac{y_{ij}^{(k)}}{h_{ij}} \tag{17}$$

So, the algorithm is exactly as before, except for the update procedure, which now changes to:

$$\Lambda_i^{(k)} = \sqrt{\Lambda_i^{(k-1)} \cdot (y_{ji}^{(k-1)}/h_{ji})} \tag{18}$$

$$\Lambda_j^{(k)} = \sqrt{(y_{ij}^{(k-1)}/h_{ij}) \cdot \Lambda_j^{(k-1)}} \tag{19}$$

Note that the only difference from the simple case is that $\Lambda_j^{(k-1)}$ and $\Lambda_i^{(k-1)}$ are replaced by their ML estimates for the other sensor, which is derived as in (17). Proving theoretically that this algorithm converges to the same decision as the centralized scheme is more complicated in this case. However, in the next subsection we will show this by some simulations.

### C. Performance Evaluation

For evaluating the performance of the proposed algorithm, we performed it on random geometric graph with 25 nodes. In our experiment the two hyotheses have equal prior probabilities. At the begining, each node takes an observation which is corrupted by noise and computse its log-likelihood ratio. Then all nodes try to reach a consus by gossiping their likelihood ratios in several rounds (in our program, we performed 25000 iterations). We considered two measure of performance: probability of detection, $\Pr(H_1|H_1)$, and probability of false alarm, $\Pr(H_1|H_0)$. The former shows the probability of detecting the $H_1$ hypotheses and we want it to be as high as possible. The latter is the probability of detcting $H_1$ while $H_0$ is true, and clearly we want it to be as low as possible. We plotted these two measures for different SNRs. Actually, we computed the average of these measures over all the nodes and for several repetitions of the algorithm (5000 times). It is clear from Fig. 3 and Fig. 4 that as SNR increases, we have the higher probability of detection and lower probability of false alarm. And, of course, if the SNR is large enough, the algorithm works perfectly (Pr(detection) = 1 and Pr(false alarm) = 0).

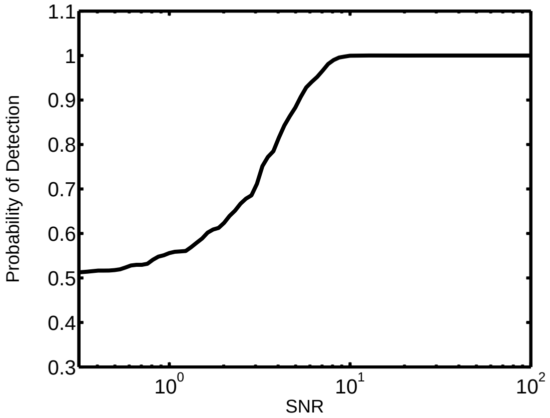

Fig. 3. Probability of detection versus SNR. As SNR increases, the probability of detection approaches 1

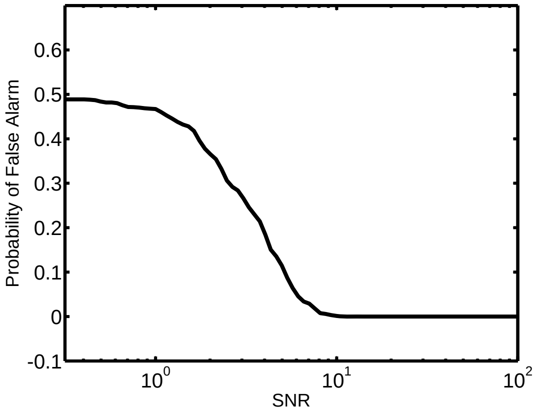

Fig. 4. Probability of false alarm versus SNR. As SNR increases, the probability of false alarm approaches 0

The result is that for large enough SNR (about 10 here), the algortihm works well and the event of interest is detected with probability one. Also, the probability of wrong detection goes to zero as we increase the SNR. So, we showed that even with channel constraints our proposed algorithm converges to the same decision as the centralized scheme.

### D. Comparison With Other Algorithms

The main difference of our proposed algorithm with the previous ones is that it is fully decentralized. In other words, there is no central node (fusion center) in our algorithm, while in the other ones, all the nodes send their messages to the fusion center and the final detection process is performed centrally. Also, after performing the gossip-based algorithms, all the nodes are aware of the final decision while in the other algorithms only central node knows about the detected hypothesis. However, in our algorithm all the nodes should be aware of the channel fading coefficients, while in the other ones only the fusion center needs to have such information about the channel. In case of performance, our algorithm performs slightly better than the previous ones. For example the algorithms proposed in [4] achieve same probabilities of detection as our proposed algorithm, with higher SNRs.

## VI. Conclusion

In this project, we reviewed the problem of channel aware decentralized detection in wireless sensor networks. First, we reviewed the classical framework, where we do not consider power and channel constraints and showed that with person-by-person optimization we will get a set of likelihood ratio tests whose thresholds depend on each other. Then, we reviewed some channel aware algorithms that take into account the constraints of the wireless sensor networks in real world, such as power constraints, noise, and fading. First, we derived some approximations for the fusion center likelihood ratio in the presence of noise and flat fading channels in the cases of large and low SNR. Then, we briefly reviewed the sensor censoring approach in which nodes only transmit to the fusion center if their observations are informative (their likelihood ratio do not belong to a certain interval). Finally, we proposed a totally decentralized and channel aware algorithm for decentralized detection, which is based on gossiping. We proved the optimality and convergence of the algorithm in the case where there is no noise or fading. The performance of the proposed algorithm for the case of AWGN and flat fading channels have been evaluated through simulations and it was shown that for large enough SNRs, the algorithm performs perfectly ($\Pr$(detection)=1 and $\Pr$(false alarm)=0).

## References

[1] J. N. Tsitsiklis, "Decentralized Detection," *Advances in Signal Processing*, vol. 2, pp. 297–344, 1993.

[2] R. Tenney and N. Sandell, "Detection with distributed sensors," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-17, no. 4, pp. 501 –510, 1981.

[3] Q. Yan and R. Blum, "Distributed signal detection under the Neyman-Pearson criterion," *Information Theory, IEEE Transactions on*, vol. 47, pp. 1368 –1377, May 2001.

[4] B. Chen, R. Jiang, T. Kasetkasem, and P. Varshney, "Channel aware decision fusion in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 52, no. 12, pp. 3454 – 3458, 2004.

[5] Y. Lin, B. Chen, and P. Varshney, "Decision fusion rules in multi-hop wireless sensor networks," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 41, no. 2, pp. 475 – 488, 2005.

[6] C. Rago, P. Willett, and Y. Bar-Shalom, "Censoring sensors: a low-communication-rate scheme for distributed detection," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 32, no. 2, pp. 554 –568, 1996.

[7] R. Jiang and B. Chen, "Fusion of censored decisions in wireless sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 4, no. 6, pp. 2668 – 2673, 2005.

[8] R. Jiang and B. Chen, "Decision fusion with censored sensors," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 2, pp. ii – 289–92 vol.2, May 2004.

[9] K. Tsianos and M. Rabbat, "Fast decentralized averaging via multi-scale gossip," *Proc. IEEE Distributed Computing in Sensor Systems*, June 2010.

[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2508–2530, June 2006.

## Appendix A
## Proof of Theorem 2

We use the result of [10] to prove Theorem 2. It was shown in [10] that:

**Lemma 1.** Assume that we have $N$ nodes with initial values $x_i^{(0)}, \quad i = 1, \ldots, N$. Consider the following gossip algorithm: At each iteration, nodes choose one of their neighbors at random and gossip with each other. When two nodes ($i$ and $j$) gossip with each other, they update their values according to the following procedure:

$$x_j^{(k)} = x_i^{(k)} = \frac{x_i^{(k-1)} + x_j^{(k-1)}}{2} \tag{20}$$

where $x_i^{(k)}$ denotes the value of the sensor $i$ at iteration $k$ of the algorithm ($0 \leqslant k$). This gossip algorithm converges in expectation and we have:

$$\lim_{k \to \infty} E\left[x_i^{(k)}\right] = \frac{\left[\sum_{j=1}^{N} x_j^{(0)}\right]}{N}, \quad i = 1, \ldots, N \tag{21}$$

The proof of the lemma can be found in [10]. Now, if we do the following replacement:

$$x_i^{(k)} \triangleq \log(\Lambda_i^{(k)}) \quad \text{for} i = 1, \ldots, N \quad \text{and} \quad 0 \leqslant k \tag{22}$$

then Lemma 1 transforms to Theorem 2, because the update procedure in Theorem 2:

$$\Lambda_i^{(k)} = \sqrt{\Lambda_i^{(k-1)} \cdot \Lambda_j^{(k-1)}} \tag{23}$$

is equivalent to

$$\log\left(\Lambda_i^{(k)}\right) = \frac{\log\left(\Lambda_i^{(k-1)}\right) + \log\left(\Lambda_j^{(k-1)}\right)}{2} \tag{24}$$

which is the update rule of Lemma 1. Also, we have:

$$\log\left(\left[\prod_{j=1}^{N} \Lambda_j^{(0)}\right]^{1/N}\right) = \frac{\sum_{j=1}^{N} \log\left(\Lambda_i^{(0)}\right)}{N} \tag{25}$$

which shows that Theorem 2 is equivalent to Lemma 1 in the log domain and thus the proof is complete.

# Appendix B
# Matlab Codes

```
% Wireless Project
% Gossip-Based Decentralized Detection
% Milad Kharratzadeh
% April, 2011

% Initializations
n = 25;
SNR = logspace(-1,2,200);
K=5000;
A = RGG(n,0.4); % Adjacency Matrix
H = abs(raylrnd(ones(n,n))) .* A;
LLR = zeros(n,1); % Local Log-Likelihood Ratio
U = zeros(n,1); % Local Sensors Decisions
d = sum(A,2)';
siz = size(A);
edges = find(A == 1);
nel = numel(edges);
l=1;
P_Detection = zeros(200,1);
P_FalseAlarm = zeros(200,1);

% Performing Algorithm
for snr=SNR
    detected = 0;
    falseAlarm = 0;
    hyp1 = 0;
    hyp0 = 0;
    sigma = 1/sqrt(snr);
    for i=1:K
        hyp = rand>0.5;
        r = hyp*ones(n,1) + sigma*randn(n,1); % Observation at sensors
        LLR = (2*r-1)/(2*sigma^2); % Calculating local Log-LRs
        for j=1:25000
            [rand_i, rand_j] = ind2sub(siz, edges(randi(nel)));
            temp = LLR(rand_i)+LLR(rand_j);
            % Updating log-likelihood ratios
            LLR(rand_i)=(temp+(sigma*randn)/H(rand_j,rand_i))/2;
            LLR(rand_j)=(temp+(sigma*randn)/H(rand_i,rand_j))/2;
        end
        U = (LLR>0); % Making final decisions
        if hyp==1
            detected = detected + sum(U==1);
            hyp1 = hyp1 + 1;
        else
            falseAlarm = falseAlarm + sum(U==1);
            hyp0 = hyp0 + 1;
        end
    end

    P_Detection(l) = detected / (hyp1*n);
    P_FalseAlarm(l) = falseAlarm / (hyp0*n);
    l = l+1;
end

semilogx(SNR,P_Detection);
hold on;
semilogx(SNR,P_FalseAlarm);
hold off;
figure;
plot(sort(P_FalseAlarm),sort(P_Detection));
```

```
% Producing a random geometric graph and returning its adjacency matrix
function A = RGG(n,r)

A = zeros(n,n);
X = rand(n,1);
Y = rand(n,1);

% Plotting the graph:

% scatter(X,Y,'red','filled');
% hold on;

for i=1:n
    for j=i:n
        if (((X(i)-X(j))^2 + (Y(i)-Y(j))^2) ⩽ r^2) && (i≠j)
            A(i,j) = 1;
%             line([X(i) X(j)], [Y(i) Y(j)],'Color','k','LineWidth',1.5);
%             hold on;
        end
    end
end

% hold off;
A = A + A';
```