Distributed Decision Fusion With a Random-Access Channel for Sensor Network Applications

Yingqin Yuan, *Student Member, IEEE,* and Moshe Kam, *Fellow, IEEE*

***Abstract*—The envisioned use of sensor networks for military and civilian applications calls for the deployment of a large number of sensors for extended periods over wide geographical areas. Due to energy and bandwidth constraints, most sensor fields would be "dormant," initiating communication only when a threat is detected. Many architectures under consideration call for compression of locally sensed data into binary "target/no target" decisions. These schemes use a common channel for communication between local detectors (LDs) and the decision fusion center (DFC) that integrates the multiple detector decisions. Unlike most parallel fusion schemes, which use a dedicated communication channel between every LD and the DFC, sensor fields would employ a single shared, time-slotted communication link for all LDs to transmit decisions to the DFC. In this paper, we introduce two window-based methods for managing traffic over this channel. The first allows simultaneous messages to collide. The DFC uses the statistics of the channel states (i.e., the numbers of successful transmissions, idle slots, and collisions during a specified time period) to make its global target/no target decision. The second method uses a simple collision resolution algorithm (CRA), similar to slotted-ALOHA, with dynamically updated retransmission probability. With this scheme, the DFC makes its decision based on data communicated successfully within a specified time window. Overall performance of the two approaches are presented and compared. Simple rules are developed for assessing the conditions under which each one is preferred.**

***Index Terms*—Decision fusion, maximum a posteriori (MAP) estimation, multiple access channel, random access, sensor network.**

I. Introduction

TACTICAL military needs along with progress in sensor design have given rise to new designs of large sensor networks and sensor fields [1]–[10]. Typically such designs call for deployment of hundreds or even thousands of devices, which then cluster themselves and select local cluster leaders (e.g., [3]). Under normal circumstances sensor fields would be dormant, but upon the detection of external threats (such as a hostile target) the danger-sensing sensors would alert the cluster leader, and a group of cluster leaders may then alert higher level leaders in hierarchical order. Assuming that the local sensors act as binary "target/no target" detectors, we can view the cluster leader as a decision fusion center (DFC) , operating in a parallel decision fusion architecture. Such parallel decision fusion schemes [11]–[17] consist of a bank of local sensors/detectors, a DFC, and dedicated communication links between the local detectors (LDs) and the DFC. The DFC receives the LD decisions and integrates them into a global decision, according to some performance index, such as a Bayesian cost or a Neyman-Pearson criterion [11], [16]. To save bandwidth, the local sensor readings are often compressed, in the extreme into a 1-bit decision (target/no target).

In [13], Gini *et al.* studied the relation between bandwidth and performance of distributed detection schemes of this kind. Rago *et al.* [14] investigated a specific low-bandwidth scheme, where local likelihood ratios were transmitted to the DFC. Duman and Salehi [15] explored the effects of noise and interchannel interference in decision fusion architecture over multiple-access channels. In this paper, we consider an additional complication, namely the need of sensor field architectures to use a shared channel for communication between multiple LDs and the DFC. In large sensor fields, one can not dedicate a separate communication channel for each sensor, and typically all LDs share the same channel. Since this single channel serves a bank of asynchronous uncoordinated LDs, messages would occasionally collide with each other, and the use of a collision resolution algorithm (CRA) becomes necessary.

In our application, all LDs are connected to the common communication channel and access it using a random access protocol. At each time slot, the common channel is in one of three possible states, namely, *success*, *idle*, or *collision*. By *success* we mean that during the time slot one and only one local detector tried to transmit its local decision across to the DFC and the transmission was successful. By *idle* we mean that no LD attempted to transmit its local decision to the DFC during that time slot. By *collision* we mean that two or more LDs have attempted transmission of messages simultaneously during the time slot, and thus the transmitted local decisions collided with each other. The role of a CRA is to control the traffic and determine how transmission attempts that failed due to collisions should be retried so as to maximize the channel throughput [18]. However, before we turn to CRA design, we examine here what the DFC could do on the basis of knowing only the number of *success*, *idle*, and *collision* time slots in a time window, without a CRA. This base case would provide an upper bound on the performance of the common channel decision fusion architecture. A lower bound can be developed by considering the standard architecture, which dedicates a separate channel to each LD (e.g., [16], [17]). The performance of a system with a CRA will reside between these two bounds.

This paper is organized as follows. Following the introduction, we describe the system architecture and the main assumptions in Section II. In Section III, we describe the DFC computations for the two architectures (with or without CRA). We present the performance of the two schemes and compare it

Manuscript received June 15, 2003; revised March 25, 2004.

The authors are with the Data Fusion Laboratory, Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104 USA (e-mail: yingqin,kam@minerva.ece.drexel.edu).

Digital Object Identifier 10.1109/TIM.2004.830598

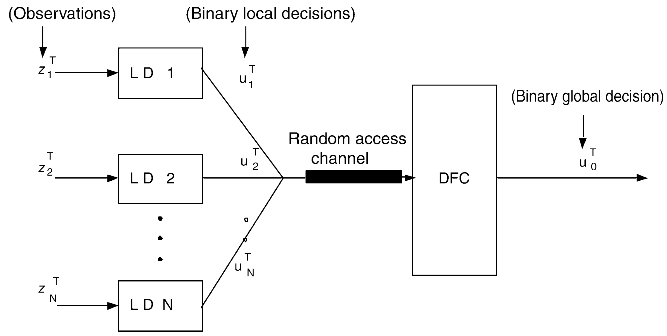

Fig. 1. Parallel decision fusion with a common LD-DFC channel.

to that of a fully connected parallel decision fusion architecture ([16]). In Section IV, we conclude that in general the scheme with a CRA is preferred, but identify special circumstances where the system operates better without one.

II. System Structure

The decision fusion structure studied here is shown in Fig. 1. The task of this scheme is binary hypothesis testing. H_0 and H_1 denote the two hypotheses. H_0 represents null hypothesis and H_1 represents the alternative hypothesis. The *a priori* probabilities of H_0 and H_1 are assumed constant and known, and denoted P_0 and P_1, respectively. As shown in Fig. 1, there are N local detectors in the system, all connected to the same channel. The local decision rules are fixed and every local sensor is characterized by its known false alarm probability $P_{fi} = P(\text{detector } i \text{ accepts } H_1|H_0)$ and its missed detection probability $P_{mi} = P(\text{detector } i \text{ accepts } H_0|H_1)$ $(i = 1, 2, \ldots, N)$. We use the time slot of the communication channel as the time unit of the system. One slot is considered long enough for each sensor to sample the environment, make a local decision, and transmit this decision to the DFC if the channel is available. W successive time slots constitute a "W window." All LDs are synchronized so that they sample the environment and make decisions at the beginning of each W window.[1] In our scheme, every LD that decided in favor of H_1 tries to transmit its decision to the DFC by picking up a random number between 0 and 1 at the beginning of each time slot during this W window. If the random number happens to be less than or equal to a certain $r \in [0, 1]$, called the *retransmission probability*, then the LD sends its decision at this time slot to the multiple access channel. The parameter r is tunable and could depend on feedback from the channel.

When we do not use a CRA, the LDs repeat this process regardless of their success or failure in previous time slots. When we use a CRA, LDs, that were successful in transmitting their decisions at any time slot within the window, become inactive until the end of that current W window. In either method, LDs that favor H_0 make no attempt to transmit their opinions.

The DFC updates its global decision at the end of the W window based on all available information. Earlier updates are also possible, using all available information up to that point.

[1]The problem of synchronization of sensor networks has been studied intensively, with promising results [6]–[8].

In Fig. 1, $z_i^T, i = 1, 2, \ldots, N, T = 1, 2, \ldots$ is the observation made by the ith local sensor at the beginning of the Tth window. Temporal and spatial statistical independence are assumed here for all observations of all N sensors, conditioned on the hypothesis. We use $u_i^T (i = 1, 2, \ldots, N, T = 1, 2, \ldots)$ to denote the local decision made by the ith LD at the beginning of the Tth W window. $u_i^T = 1$ is used to represent acceptance of H_1 and $u_i^T = 0$ to represent acceptance of H_0 by the ith LD. We use u_o^T to denote the global decision made by the DFC at the end of Tth window.

III. DFC Decision Rule

A. Decision Rule Based on the Channel States; No CRA

To derive the DFC decision rule when no CRA is employed, we have to determine the distribution of the channel states within a W window under the two hypotheses. For simplicity, we assume in this paper that all sensors have the same false alarm probability denoted P_f, and the same missed detection probability, P_m.

We define the state (ready to transmit) RTT_i^k for the ith LD at the beginning of time slot k. $\mathrm{RTT}_i^1 = 1$ if $u_i^T = 1$ (the LD accepts H_1) in the beginning of a W window. Otherwise $\mathrm{RTT}_i^1 = 0$. For $2 \leq k \leq W$, $\mathrm{RTT}_i^k = 1$ if $\mathrm{RTT}_i^1 = 1$. Otherwise, $\mathrm{RTT}_i^k = 0$. We note that

$$\begin{aligned} P\left(\mathrm{RTT}_i^1 = 1|H_0\right) &= P_f, \text{ and} \\ P\left(\mathrm{RTT}_i^1 = 1|H_1\right) &= 1 - P_m = P_d. \end{aligned} \tag{1}$$

We denote $K_{\mathrm{RTT}}(k) = \sum_{i=1}^N \mathrm{RTT}_i^k$ and note that

$$\begin{aligned} P(K_{\mathrm{RTT}}(1) = n|H_0) &= \binom{N}{n} P_f^n (1 - P_f)^{N-n} \\ P(K_{\mathrm{RTT}}(1) = n|H_1) &= \binom{N}{n} P_d^n (1 - P_d)^{N-n}. \end{aligned} \tag{2}$$

Given that there are n LDs in state $\mathrm{RTT} = 1$ at the beginning of a W window, the probabilities that any of the W time slots will be a *success*, *idle* or *collision* slot ($P_{\mathcal{S}|n}$, $P_{\mathcal{I}|n}$ and $P_{\mathcal{C}|n}$, respectively) are

$$\begin{aligned} P_{\mathcal{S}|n} &= nr(1 - r)^{n-1}, \\ P_{\mathcal{I}|n} &= (1 - r)^n, \text{ and} \\ P_{\mathcal{C}|n} &= 1 - P_{\mathcal{S}|n} - P_{\mathcal{I}|n}. \end{aligned} \tag{3}$$

Now, if we let $\mathcal{N}_\mathcal{S}$, $\mathcal{N}_\mathcal{I}$, and $\mathcal{N}_\mathcal{C}$ be the number of *success* slots, *idle* slots and *collision* slots in a W window, then the probability that $\mathcal{N}_\mathcal{S} = n_\mathcal{S}$, $\mathcal{N}_\mathcal{I} = n_\mathcal{I}$ and $\mathcal{N}_\mathcal{C} = n_\mathcal{C}$, given that $K_{\mathrm{RTT}}(1) = n$ (namely n LDs wanted to transmit at the beginning of the window), is

$$P(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}|n) = \binom{W}{n_\mathcal{S}} \binom{W - n_\mathcal{S}}{n_\mathcal{I}} * P_{\mathcal{S}|n}^{n_\mathcal{S}} P_{\mathcal{I}|n}^{n_\mathcal{I}} P_{\mathcal{C}|n}^{n_\mathcal{C}} \tag{4}$$

where $n_\mathcal{S} + n_\mathcal{I} + n_\mathcal{C} = W$. The joint probabilities that $\mathcal{N}_\mathcal{S} = n_\mathcal{S}$, $\mathcal{N}_\mathcal{I} = n_I$ and $\mathcal{N}_\mathcal{C} = n_C$ in a W window, under the hypotheses H_0 and H_1 (which we denote as $P_0(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C})$ and $P_1(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C})$, respectively) are

$$P_0(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}) = \sum_{n=0}^{N} P(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}|n) * P(K_{\mathrm{RTT}}(1) = n|H_0)$$

and

$$P_1(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}) = \sum_{n=0}^{N} P(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}|n) * P(K_{\mathrm{RTT}}(1) = n|H_1). \quad (5)$$

The DFC rule for minimizing the probability of error[2] is

$$\mathcal{L}(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}) = \frac{P_0(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C})}{P_1(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C})} \underset{H_1}{\overset{H_0}{\gtrless}} \frac{P_1}{P_0} = \tau \quad (6)$$

where $\mathcal{L}(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C})$ is the likelihood ratio and τ is the decision threshold. We partition the set $\Omega = (n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}|0 \le n_\mathcal{S} \le W, 0 \le n_\mathcal{I} \le W - n_\mathcal{S}, n_\mathcal{C} = W - n_\mathcal{S} - n_\mathcal{I})$ into two subsets, $\Omega_0 = (n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}|\mathcal{L}(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}) > \tau)$ and $\Omega_1 = (n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}|\mathcal{L}(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}) \le \tau)$. The DFC will make a decision of $u_o = 0$ if $(n_\mathcal{I}, n_\mathcal{S}, n_\mathcal{C}) \in \Omega_0$ and $u_o = 1$ otherwise. The global false alarm probability and the missed detection probability of the DFC, $(P_F)_0$ and $(P_M)_0$, are

$$(P_F)_0 = \sum_{(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}) \in \Omega_1} P_0(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}),$$
$$(P_M)_0 = \sum_{(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}) \in \Omega_0} P_1(n_\mathcal{S}, n_\mathcal{I}, n_\mathcal{C}). \quad (7)$$

The global error probability $(P_{\mathrm{err}})_0$ is

$$(P_{\mathrm{err}})_0 = P_0(P_F)_0 + P_1(P_M)_0. \quad (8)$$

The retransmission probability r is a design parameter, ideally set to

$$r = \arg\min_{r \in [0,1]} (P_{\mathrm{err}})_0.$$

Fig. 2 shows the relationship between the error probability and the window length W for the simple case $P_0 = P_1 = 1/2$, $P_f = P_m = 0.1$ and $N = 5$, 10, 20, and 30. As expected, the error probability decreases as the window length W and the number of sensors N increase. In Fig. 3, we show the probability of error versus the number of LDs N for a system with $P_0 = P_1 = 1/2$, $P_f = P_m = 0.1$. From Figs. 2 and 3 [especially Fig. 3(a)], we note that unless W (window length) is increased with N (number of sensors), the performance does not continue to improve with N. This is understandable because for large N and small W, the system is unable to distinguish between LD transmission attempts on account of (a) observing H_1; and (b) observing H_0 but experiencing false alarm in multiple LDs. To avoid this situation, we need $NP_f \ll W$.

B. Dynamic Collision Resolution-Based Scheme

The performance of the decision rule based on channel states just described is expected to be limited. It does not converge

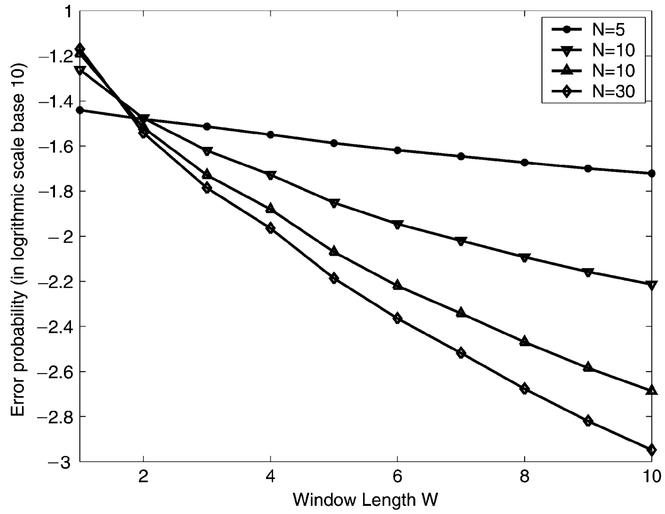

Fig. 2. Global performance of a system with no CRA and with $P_f = P_m = 0.1, P_0 = P_1 = 1/2$.

(as $W \to \infty$) to the performance of a system with dedicated LD-DFC channels (which is calculated in [16]). To improve performance, we need to employ a collision resolution algorithm (CRA), that provides the DFC with actual LD decisions, not just transmission attempts.

There are several CRAs to select from [18]. Here we follow slotted ALOHA, one of the simplest random access protocols available.[3]

As before, at the beginning of every W window, all sensors are synchronized, they sample the environment, and make their decisions, u_i^T. Those LDs making the decision of $u_i^T = 1$ at the beginning of a W window are in state $\mathrm{RTT}_i^1 = 1$ and will try to transmit their decisions to the DFC within this window. All others are in state $\mathrm{RTT}_i^1 = 0$. Once a LD was able to transmit its decision successfully, it moves to $\mathrm{RTT}_i = 0$ for the remaining duration of the window. We assume that there are three kinds of channel feedback available to all LDs and to the DFC at the end of each time slot. They are $\mathcal{S}$ for *success*, $\mathcal{I}$ for *idle*, and $\mathcal{C}$ for *collision*. All LDs with $RTT_i^k = 1$ will try to transmit at the kth time slot with probability r (the *retransmission probability*). We assume r is updated at each time slot, and its value, as well as the total number of detectors (N), the local error probabilities (P_f, P_m), and the number of successful transmissions so far $(S(k))$ are known to all LDs and the DFC. Similar to slotted ALOHA, we set the *retransmission probability* to be the reciprocal of the expected number of transmitting LDs (ideally $1/K_{\mathrm{RTT}}(k)$). Since the true value of $K_{\mathrm{RTT}}(k)$ is unknown, each LD and the DFC in our system estimate the number of LDs who wish to transmit at the beginning of the W window ($K_{\mathrm{RTT}}(1)$, the number of LDs who decided $u_i^T = 1$ at $k = 1$) and update their estimates of $K_{\mathrm{RTT}}(k)$ by subtracting the number of the successful transmissions between time slot 1 and time slot k from the estimated $K_{\mathrm{RTT}}(1)$. To esti-

[2]Can be adjusted for a general Bayesian cost.

[3]The ALOHA network was developed in the early 70 s for radio communication between a central computer and data terminals at the University of Hawaii. Slotted ALOHA was subsequently suggested—the basic idea is that "each unbacklogged node simply transmits a newly arriving packet in the first slot after the packet arrival, thus risking occasional collisions but achieving very small delay if collisions are rare" [19, p. 277]

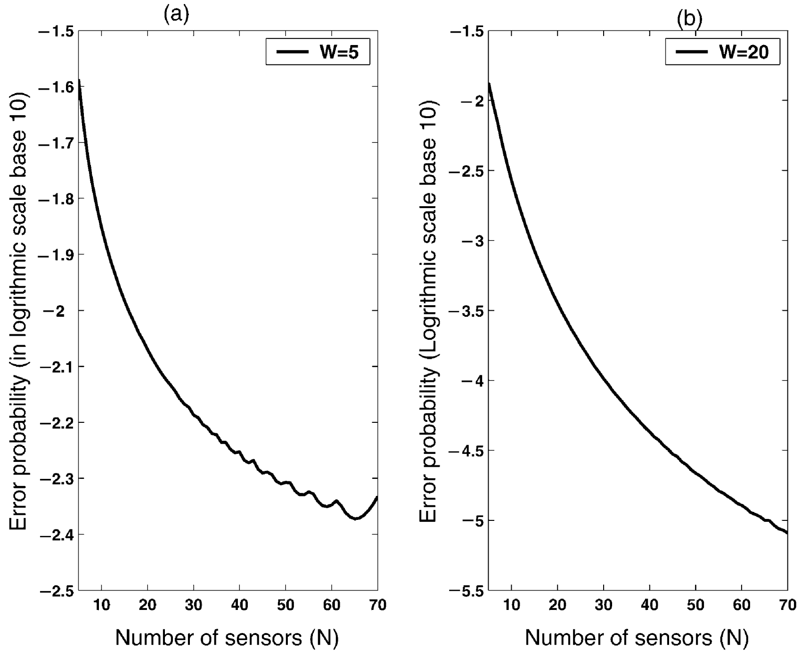

Fig. 3. Global performance of a system with no CRA and with $P_f = P_m = 0.1, P_0 = P_1 = 1/2$. (a) Short window $W = 5$. (b) Intermediate window $W = 20$.

mate $K_{RTT}(1)$, we employ an approximate Bayesian maximum *a posteriori* (MAP) estimator [20].

Let X_l, $l = 1, 2, \dots W$ denote the random state of the lth time slot in a W window with realization $x_l \in \{\mathcal{S}, \mathcal{I}, \mathcal{C}\}$. Let $x_{l:m} = (x_l, x_{l+1}, \dots, x_m)$, $1 \le l \le m \le W$ denote a realization of a sequence of states from the lth time slot to the mth time slot inside a W window. $\hat{K}^k_{\mathrm{RTT}}(1)$ is the estimate of $K_{\mathrm{RTT}}(1)$ after k time slots. Moreover, the estimate of $K_{\mathrm{RTT}}(k)$, $\hat{K}_{\mathrm{RTT}}(k) = \hat{K}^k_{\mathrm{RTT}}(1) - S(k)$, where $S(k)$ is the number of successful transmissions from time 1 up to the kth time slot. We try to find (for $k \ge 1$)

$$\begin{aligned}\hat{K}^k_{\mathrm{RTT}}(1) &= \arg\max_n P(K_{\mathrm{RTT}}(1) = n | x_{1:k}) \\ &= \arg\max_n (P(x_{1:k} | K_{\mathrm{RTT}}(1) = n) \\ &\quad * P(K_{\mathrm{RTT}}(1) = n))\end{aligned} \tag{9}$$

where $n \in \{1, 2, \dots, N\}$; we note that the *a priori* estimate of $K_{\mathrm{RTT}}(1)$ is $\hat{K}^0_{\mathrm{RTT}}(1) = N(P_0 P_f + P_1 P_d)$. The term $P(x_{1:k} | K_{\mathrm{RTT}}(1) = n)$ denotes the probability that, given that n LDs accepted H_1 at the beginning of the W window, the channel state sequence is $x_{1:k}$. For $k \ge 2$ we have

$$\begin{aligned}P(x_{1:k} | K_{\mathrm{RTT}}(1) = n) &= P(x_{1:k-1}, x_k | K_{\mathrm{RTT}}(1) = n) \\ &= P(x_k | x_{1:k-1}, K_{\mathrm{RTT}}(1) = n) \\ &\quad * P(x_{1:k-1} | K_{\mathrm{RTT}}(1) = n)\end{aligned} \tag{10}$$

with

$$\begin{aligned}&P(x_{1:1} | K_{\mathrm{RTT}}(1) = n) \\ &= P(x_1 | K_{\mathrm{RTT}}(1) = n) \\ &= \begin{cases} n * r(1) * [1 - r(1)]^{n-1} & x_1 = \mathcal{S} \\ [1 - r(1)]^n & x_1 = \mathcal{I} \\ 1 - [1 - r(1)]^n - n * r(1) * [1 - r(1)]^{n-1} & x_1 = \mathcal{C} \end{cases}\end{aligned} \tag{11}$$

and

$$\begin{aligned}P(K_{\mathrm{RTT}}(1) = n) &= P(K_{\mathrm{RTT}}(1) = n | H_0) P_0 \\ &\quad + P(K_{\mathrm{RTT}}(1) = n | H_1) P_1.\end{aligned} \tag{12}$$

The value of $r(1)$ is $1/N(P_0 P_f + P_1 P_d)$, the reciprocal of the initial estimate of $K_{\mathrm{RTT}}(1)$.

We now have a recursive form of estimating $\hat{K}^k_{\mathrm{RTT}}(1)$ (see (2) for $P(K_{\mathrm{RTT}}(1) = n | H_i)$, $i = 0, 1$).

The multiplicative updating term in (10) is 0 if $n - S(k-1) < 0$. Otherwise for $k \ge 2$

$$\begin{aligned}&P(x_k | x_{1:k-1}, K_{\mathrm{RTT}}(1) = n) \\ &= \begin{cases} P(x_k = \mathcal{S} | x_{1:k-1}, K_{\mathrm{RTT}}(1) = n) = P_{\mathcal{S}|n}(k) \\ P(x_k = \mathcal{I} | x_{1:k-1}, K_{\mathrm{RTT}}(1) = n) = P_{\mathcal{I}|n}(k) \\ P(x_k = \mathcal{C} | x_{1:k-1}, K_{\mathrm{RTT}}(1) = n) = P_{\mathcal{C}|n}(k) \end{cases}\end{aligned} \tag{13}$$

with

$$\begin{aligned}P_{\mathcal{S}|n}(k) &= [n - S(k-1)] * r(k) * [1 - r(k)]^{n - S(k-1) - 1} \\ P_{\mathcal{I}|n}(k) &= [1 - r(k)]^{n - S(k-1)} \\ P_{\mathcal{C}|n}(k) &= 1 - P_{\mathcal{S}|n}(k) - P_{\mathcal{I}|n}(k).\end{aligned} \tag{14}$$

In (14) we use $r(k)$ to indicate that the retransmission probability is time-varying. Using the estimate of $\hat{K}_{\mathrm{RTT}}(k)$ from (9)–(14), every LD updates the *retransmission probability* as

$$r(k+1) = \min\left(1, \frac{1}{\hat{K}_{\mathrm{RTT}}(k)}\right), \quad k = 1, 2, \dots, W. \tag{15}$$

We note that r tends to 1 after a long string of *idle* slots.

Our estimation algorithm is summarized in Table I.

At every time step, the DFC uses the standard Bayesian decision rule with all available LD decisions (see [17]). The sufficient statistic for the DFC is the weighted sum of the local decisions. The weights are determined by the values of P_d and P_f of the corresponding LD that makes this decision. If all LDs

TABLE I
ESTIMATING $K_{\mathrm{RTT}}(k)$

1. Initialization ($k = 1$, the first time slot)

- Set $r(1) = 1/N(P_0 P_f + P_1 P_d)$;
- Calculate $P(x_{1:1}|K_{RTT}(1) = n)$ from equation (11);
- Estimate $\hat{K}_{RTT}(1)$ from equations (2), (9), (11) and (12);
- Set $\hat{K}_{RTT}(1) = \hat{K}^1_{RTT}(1) - S(1)$;
- Calculate $r(2)$ from equation (15).

2. For $k = 2, 3, \ldots, W$

- Obtain $P(x_k|x_{1:k-1}, K_{RTT}(1) = n)$ from (13) and (14);
- Calculate $P(x_{1:k}|K_{RTT}(1) = n)$ from equation (10);
- Get $\hat{K}_{RTT}(k)$ using $P(x_{1:k}|K_{RTT}(1) = n)$, (2) and (12);
- Set $\hat{K}_{RTT}(k) = \hat{K}^k_{RTT}(1) - S(k)$;
- Update $r(k+1)$ according to equation (15).

have the same P_d and P_f, the decision rule becomes a "K out of N" rule of the form

$$\hat{K}^k_{\mathrm{RTT}}(1) \underset{H_1}{\overset{H_0}{\gtrless}} \frac{N * \left[\frac{\log(1-P_d)}{(1-P_f)}\right] + \log\left(\frac{P_1}{P_0}\right)}{\log\left(\frac{P_f(1-P_d)}{P_d(1-P_f)}\right)} \tag{16}$$

(where we assumed $P_f < P_d$). We note that rule (16) can be used at every time slot (we need not wait to the end of the window, namely to $k = W$). Performance of the DFC can be calculated using results from [16].

Table II shows simulation results of our $K_{\mathrm{RTT}}(1)$ estimator. In this simulation, $N = 10$, $P_f = P_m = 0.1$. We achieved convergence to the correct $K_{\mathrm{RTT}}(1)$, which is listed in the first column in the table, in all cases. The second and third column are the average length and standard deviation (in time slots) required for convergence. The results are based on 3000 simulation runs for each line. Fig. 4 presents two typical simulation runs for $K_{\mathrm{RTT}}(1) = 6$ and $K_{\mathrm{RTT}}(1) = 10$. The *retransmission probabilities* shown in Fig. 4 go to 1 as our estimates get better and more LDs are successful in transmitting their messages.

In Fig. 5 we show the probabilities of error against the window length W for a system with $N = 10$, $P_f = P_m = 0.1$ and $P_0 = P_1 = 1/2$. In this case $NP_f = 1$. Probabilities of error are shown for the system without CRA (calculation), with a CRA (simulation) and with dedicated LD-DFC channels (calculation based on [16]). We observe that for window length smaller than 14, the use of CRA did not improve the performance of the

TABLE II
CONVERGENCE OF THE ESTIMATION ALGORITHM FOR $K_{\mathrm{RTT}}(1)$
($N = 10, P_f = P_m = 0.1$)

$K_{RTT}(1)$	Average Time Slots	Standard Deviation
0	1.0000	0
1	2.0000	0
2	3.1733	1.8971
3	7.0793	2.9657
4	12.1090	3.8936
5	17.3867	4.5305
6	18.5600	4.3923
7	18.5410	4.8195
8	18.7030	5.2940
9	16.8377	7.9970
10	16.0647	7.8765

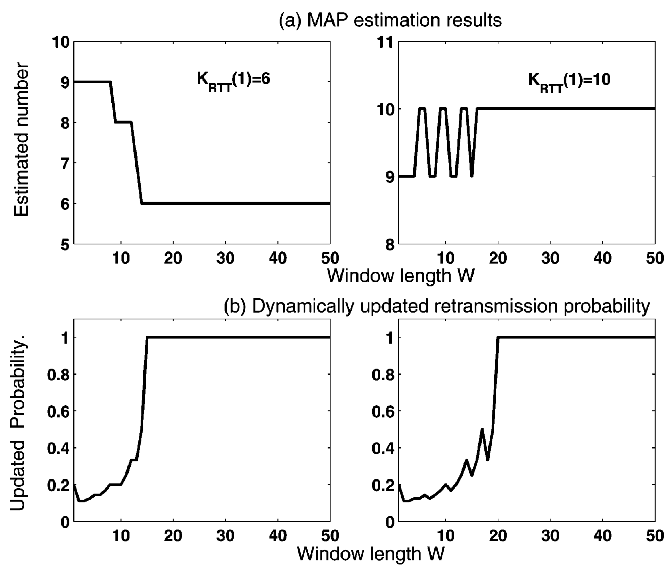

Fig. 4. Estimates for the number of LDs who need to retransmit and the optimal retransmission probability versus time with $N = 10, P_f = P_m = 0.1$.

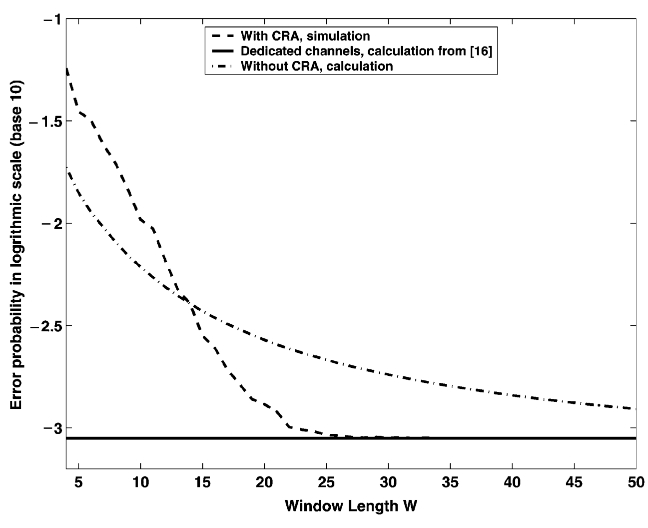

Fig. 5. Global performance comparison for three decision fusion schemes ($N = 10, P_f = P_m = 0.1$).

system, and it is better to simply count *success*, *idle*, and *collision* slots without a CRA. However as the time window becomes wider, the system with the CRA starts showing an advantage. Moreover, the system with CRA eventually converges in performance to that of a system with dedicated LD-DFC channels (since eventually all LDs with H_1 preference are able to communicate this preference to the DFC). Without a CRA, the system does not converge to the optimum.

IV. Conclusion

We have presented two decision fusion schemes for binary distributed detection architectures with a common LD-DFC channel. These schemes are geared toward sensor field applications where alerts are sent over the shared channel only when a threat is detected, and a centralized communication controller is not practical. The LDs in our architecture use a random access protocol to transmit their decisions to the DFC. The first scheme does not employ any collision resolution algorithm; the DFC makes its global decision on the basis of the statistics of the channel states within a time window. This scheme requires that the window is long enough with respect to the number of detectors and their false alarm probability ($W \gg NP_f$). The second scheme uses a simple CRA with dynamic retransmission probability. The DFC makes its decision based on the messages of those LDs that were able to transmit messages successfully during a time window. For short windows (which are larger than NP_f), the first scheme (no CRA) performs better than the second (with CRA). However the performance of the system with a CRA becomes better than the non-CRA scheme as the window length grows, and further it converges to the optimal performance as $W \to \infty$. In general, the performance of the system without CRA does not converge to the optimal performance (*viz.* the performance of a system with dedicated LD-DFC channels).

References

[1] L. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, pp. 102–114, Aug. 2002.

[2] J. Nemeroff, L. Garcia, D. Hampel, and S. Dipierro, "Application of sensor network communications," in *IEEE Proc. Military Commun. Conf.*, vol. 1, Oct. 2002, pp. 336–341.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in *Proc. HICSS 2000*, Jan. 2000, pp. 3005–3014.

[4] A. Woo and D. Culler, "A transmission control scheme for media access in sensor networks," in *Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking*, July 2002, pp. 221–235.

[5] C. Lu, B. M. Blum, T. F. Abedelzaher, J. A. Stankovic, and T. He, "RAP: A real-time communication architecture for large-scale wireless sensor networks," in *Proc. RTAS'02*, Sept. 2002, pp. 55–66.

[6] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and A. Sastry, "Distributed control applications within sensor networks," *Proc. IEEE*, vol. 91, pp. 1235–1246, Aug. 2003.

[7] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcast," in *Proc. 5th Symp. Operating System Design and Implementation (OSDI)*, Boston, MA, 2002, pp. 1–17.

[8] J. Elson, "Time Synchronization of Wireless Sensor Networks," Ph.D. dissertation, Dept. Computer Science, Univ. California, Los Angeles, CA, 2003.

[9] A. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, 2002, pp. 3418–3423.

[10] J. Chamberland and V. V. Veeravalli, "Decentralized detection in sensor networks," *IEEE Trans. Signal Processing*, vol. 52, pp. 407–416, Feb. 2003.

[11] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part I-fundamentals," *Proc. IEEE*, vol. 85, pp. 54–63, Jan. 1997.

[12] R. S. Blum, S. A. Kassam, and H. V. Poor, "Distributed detection with multiple sensors: Part II-advanced topics," *Proc. IEEE*, vol. 85, pp. 64–79, Jan. 1997.

[13] F. Gini, F. Lombardini, and L. Verrazzani, "Decentralized detection strategies under communication constraints," *Inst. Elec. Eng. Proc.-Radar, Sonar, NavigProc*, vol. 145, no. 4, pp. 199–208, Aug. 1998.

[14] C. Rago, P. Willett, and Y. Bar-Shalom, "Censoring sensors: A low-communication-rate scheme for distributed detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, pp. 554–568, Apr. 1996.

[15] T. M. Duman and M. Salehi, "Decentralized detection over multiple-access channels," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, pp. 469–476, Apr. 1998.

[16] M. Kam, W. Chang, and Q. Zhu, "Hardware complexity of binary distributed detection system with isolated local Bayesian detectors," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 563–571, May/June 1991.

[17] Z. Chair and P. K. Varshney, "Optimum data fusion in multiple sensor detection systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 22, pp. 98–101, Jan. 1986.

[18] R. Rom and M. Sidi, *Multiple Access Protocols:Performance and Analysis*. New York: Springer-Verlag, 1980.

[19] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1992.

[20] H. Van Trees, *Detection, Estimation and Modulation, Part I*. New York: Wiley, 1968.

Yingqin Yuan (S'03) received the B.S. and M.S. degrees in power engineering in 1990 and 1993, respectively, from North China Electric Power University.

He joined the Data Fusion Laboratory, Drexel University, Philadelphia, PA, in 2000, where he is currently a Ph.D. degree candidate. From 1993 to 1999, he was with the Power Plant Automation Center, North China Electric Power University, conducting research on automatic control and fault diagnosis of power generation processes. From 1999 to 2000, he was a teaching assistant with the Robotics Laboratory, Drexel University. His research interests include data fusion, sensor networks, machine learning, bioinformatics, and industrial process automation.

Moshe Kam (S'75–M'77–SM'92–F'01) received the B.S. degree in electrical and electronic engineering from Tel Aviv University in 1977, and the M.S. and Ph.D. degrees from Drexel University, Philadelphia, PA, in 1985 and 1987, respectively.

Currently, he is the Robert Quinn Professor of Electrical and Computer Engineering at Drexel University, Director of Drexel's Data Fusion Laboratory, and Technical Director of the DoD-funded Applied Communications and Information Networking (ACIN) project. He also directs the Drexel NSA Center of Excellence in Information Assurance Education. His professional interests include data and decision fusion, navigation, network security, information assurance, forensic pattern recognition, and engineering education.

Dr. Kam is Director of the IEEE Region 2 (Eastern USA) and a member of the IEEE Board of Directors.