A FAULT-TOLERANT APPROACH TO SENSOR DEPLOYMENT IN DISTRIBUTED SENSOR NETWORKS

Santpal Singh Dhillon* and Krishnendu Chakrabarty
Department of Electrical & Computer Engineering
Duke University
Durham, NC 27708
E-mail: {ssd5, krish}@ee.duke.edu

Distributed sensor networks (DSNs) are a key part of the surveillance and reconnaissance infrastructure in any modern military system. DSNs offer several important benefits, such as ease of *deployment*, *responsiveness* to battlefield situations, *survivability, agility* and easy *sustainability*. These benefits make DSNs a lethal weapon for the US army, providing it with high quality of surveillance, and reconnaissance data necessary for any combat operation.

The effectiveness or quality of service (QOS) provided by any DSN depends on the coverage provided by the sensor deployment strategy. In this paper, we present novel fault-tolerant strategies for sensor deployment that provide sufficient coverage under the constraint that up to K ($K \geq 0$) sensors can fail.

Sensor placement is essential for effective utilization of sensor network resources. Intelligent sensor placement facilitates the unified design and operation of sensor/exploitation systems, and decreases the need for excessive network communication for surveillance, target location and tracking. In a typical scenario, surveillance authorities have several different types of sensors available, which can be appropriately placed in the sensor field. These sensors differ from each other in their modalities, monitoring range, detection capabilities, and cost. Intelligent sensor deployment strategies are necessary to minimize cost and yet provide sufficient sensor coverage. A sensor deployment strategy should also be inherently fault-tolerant to make it effective.

It is clearly not feasible to continuously monitor sensor status and identify failing sensors over an extended period of time. Sensors often operate in hostile conditions under severe resource constraints, and it is desirable to limit communication to useful target detection, localization, and tracking data. Therefore, the initial sensor deployment strategy must ensure that the quality of service, i.e. coverage, does not degrade due to the failure of sensors over an extended period of time. This is especially important since low-cost sensors are expected to fail in harsh operational environments.

It has been shown recently by the authors that when high coverage of the sensor field is mandated, an intelligent sensor deployment strategy reduces the number of sensors significantly compared to a random deployment strategy. However, prior work does not consider the impact of sensor failures on the sensor field coverage; in fact, it relies on the rather unrealistic assumption that sensors will never fail. Thus there is a pressing need to include fault tolerance in any intelligent sensor deployment strategy.

In this paper we present two-fault tolerant sensor placement algorithms, which we refer to as *K_FT_REP* and *K_FT_NGH,* respectively. We show that that not only are these algorithms computationally efficient, but more importantly, their computational complexity is independent of K, the number of sensors that can fail without affecting coverage. These algorithms pave the way for an effective deployment of sensors, which ensures that the sensor field is adequately covered, even in the event of failure of any K sensors. Furthermore, the algorithm *K_FT_NGH* is also independent of the base deployment strategy that is used for providing a given coverage under the assumption that no sensor failure occurs.

We model the sensor field as a two dimensional grid of points. We also assume, without loss of generality, that the probability of detection of a target by a sensor varies exponentially with the distance between the target and the sensor. (The sensor placement algorithms can also take as input other sensor detection models.) A target at distance d from a sensor is detected by that sensor with probability $e^{-\alpha d}$. The parameter α can be used to model the quality of the sensor and the rate at which its detection probability diminishes with distance. First, we present an algorithm for sensor deployment for a given set of detection probabilities in a sensor field (both with and without obstacles) with the assumption that there is no sensor failure. The goal of this sensor placement algorithm, referred to as *0_MAX_MIN_COV*, is to determine the minimum number of sensors and their locations such that every grid point is covered with a minimum confidence level. Figure 1 shows simulation results for a 2-dimensional grid with 20 grid points in each dimension for a total of 400 grid points and α = 0.5. A number of random obstacles were incorporated into the model, as a result of which a significant number of

This research was supported in part by DARPA under grant number N66001-00-1-8946, in part by Office of Naval research under grant number N00014-01-1-0712, and in part by DARPA and Army Research Office under award number DAAD19-01-1-0504. Any opinions, findings, and other conclusions or recommendations expressed in this publication are those of author(s) and do not necessarily reflect the view of the DARPA and ARO agencies.

detection probabilities were either made zero or considerably reduced due to occlusion, compared to the values obtained from our detection model.

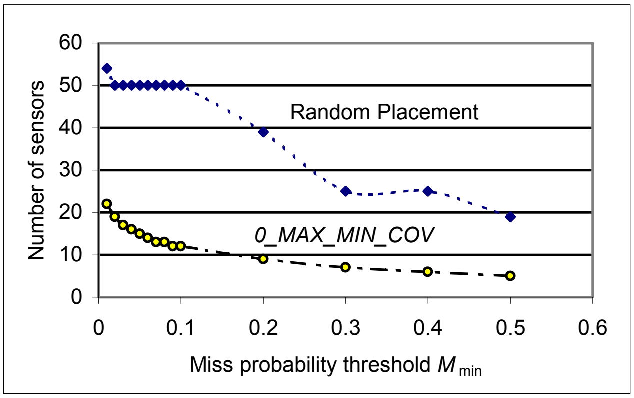

Figure 1: Comparison of *0_MAX_MIN_COV* and random placement algorithms for a 20 by 20 grid with randomly placed obstacles.

The above *Zero_FT* (*0_MAX_MIN_COV*) algorithm can be made *K_FT* by exhaustively searching for the failure of any *K* sensors and then placing additional sensors such that every grid point is covered with a minimum confidence level. The computational complexity for *Zero_FT* which is made *K_FT* by exhaustively searching for failure of *K* sensors at a time for a *n* by *n* grid is $O(n^{2} * {}^{S}C_{K})$, where *S* is the number of sensors required to obtain a given coverage threshold for every grid point according to algorithm *Zero_FT.*

The pseudocode steps of the sensor placement algorithm *K_FT_REP* are shown in Figure 2. The computational complexity for the *K_FT_NGH* and *K_FT_REP* algorithms for an *n* by *n* grid is $O(n^2)$. An important aspect of the *K_FT NGH* and *K_FT_REP* algorithms is that their computational complexity is independent of the value of *K,* which makes these algorithms a viable choice when *K* is large.

Procedure *K_FT_REP* (M, M*, M_{min})

begin

Place *S* sensors using *0_MAX_MIN_COV* or randomly (or another algorithm) such that

$M_i < M_{min}$ for all i, $1 \le i \le N$

for i: = 1 to N **do**

Choose the *K* nearest sensors, such that

$$E1\,(i) = \prod_{j=1}^{K} m_{xj,\,i}$$

$E2\,(i)$ = miss $[i]$ / $E1\,(i)$

If $E2\,(i) > M_{min}$

Place sensors at grid points according to *0_MAX_MIN_COV* (or another algorithm) such that

$E2\,(i) < M_{min}$

end

Figure 2: Pseudocode for the *K_FT_REP* algorithm

We next present simulation results for the *K_FT_NGH* and the *K_FT_REP* algorithms. In Figure 3 we present the simulation results for *K_FT_NGH* algorithm for an 8 by 8 grid with a miss probability threshold M_{min} = 0.1when the failure of *K* sensors is to be tolerated. We plot the number of sensors as a function of *K*. In the first step, we use placement strategies *0_MAX_MIN_COV* and random placement to determine the required number of sensors for covering the 8 by 8 grid with miss probability threshold M_{min} = 0.1. We then use the *K_FT_NGH* algorithm to find out the exact number of sensors required for the same threshold with *K* sensors allowed to fail.

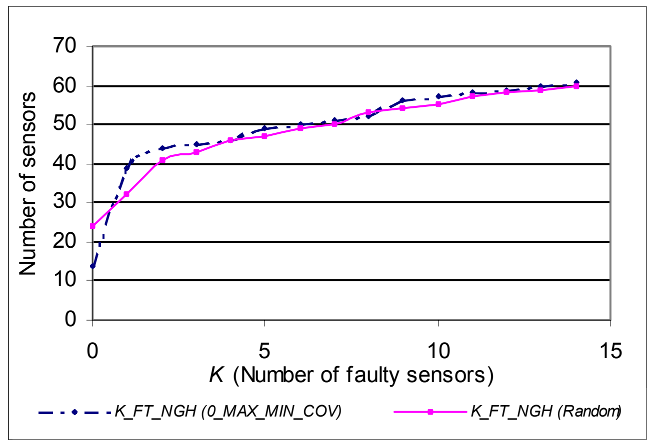

Figure 3: Number of sensors required as a function of number of faulty sensors (*K*) for the *K_FT_NGH* algorithm.

In Figure 4, we present the results for the *K_FT_REP* algorithm when the base algorithms used are *0_MAX_MIN_COV* and random placement, respectively. We again consider an 8 by 8 grid with miss probability threshold M_{min} = 0.1. We plot the results for algorithm *K_FT_REP* based on both the *0_MAX_MIN_COV* and random placement

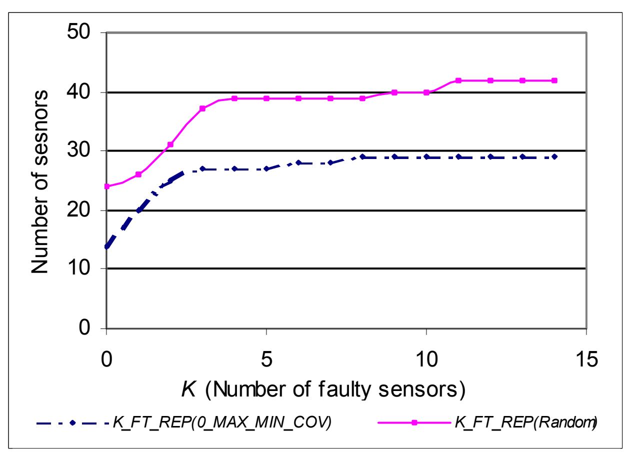

Figure 4: Number of sensors required as a function of number of faulty sensors (*K*) for the *K_FT_REP* algorithm.