This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

A Novel On-Demand Framework for Collaborative Object Detection in Sensor Networks

Guanqun Yang, Vinod Shukla, and Daji Qiao
Iowa State University, Ames, IA 50011

***Abstract*—Quality of object detection and network lifetime hold critical importance to many sensor network applications such as military surveillance. Unfortunately, improving one of these aspects comes at the expense of the other. In this paper, based on the probabilistic sensing model, we propose a novel framework for object detection in sensor networks, called DeCODe (on-Demand framework for Collaborative Object Detection), which provides a desired object detection performance (characterized in terms of detection probability and false detection probability), while attempting to prolong the network lifetime. The design of DeCODe is motivated by a counterintuitive observation that simple collaboration among active sensors indeed degrades the object detection performance. By contrast, each active sensor in DeCODe can trigger its neighboring inactive sensors to participate in the detection process in an on-demand fashion, so as to achieve the same low false detection probability while increasing the probability of detection. The effectiveness of the proposed DeCODe framework is supported by theoretical analysis and simulation-based validation.**

I. Introduction

Wireless sensor networks are being increasingly considered as a viable alternative in monitoring and surveillance systems. One of the promising applications is for object detection. Sensors are low-power and resource-constrained devices, thus making energy efficiency a prime concern in wireless sensor networks. Our objective is to develop a novel framework for collaborative object detection in sensor networks, which guarantees a minimum level of system performance while simultaneously attempting to prolong the network lifetime.

Many past works [1]–[3] for objection detection are based on the 0/1 disc sensing model where each sensor is assumed to have a sensing range and it can sense the environment and detect object inside (outside) its sensing range with probability one (zero). Obviously, this is not true in practice because such a model does not capture the degradation of a sensor's sensing capability as the distance between the sensor and the object increases. In this paper, we consider the probabilistic sensing model [4]–[7] where sensor measurements are affected by noise and the detection probability varies with the distance between the sensor and the object. Based on this sensing model, we revisit the object detection problem in sensor networks.

We characterize the performance of a sensor network by two metrics: the detection probability (P_D) and the false detection probability (P_{FD}). It is natural to think that collaboration among sensors should result in better detection performance as apposed to no collaboration; however, upon preliminary studies [8] we discover that simple decision fusion-based collaborations among active sensors indeed degrade the detection performance. This motivates us to develop a novel on-**De**mand framework for **C**ollaborative **O**bject **De**tection in wireless sensor networks, called DeCODe. In this framework, a sensor node which has potentially detected an object triggers its neighboring nodes (which may or may not be sensing) for decision making. An object detection is reported if the sensor node collects a certain number of positive alarms from its neighboring nodes. We provide theoretical analysis for P_D and P_{FD}, and support it with simulation-based validation.

The rest of the paper is organized as follows. We discuss the related work in Section II and give models and assumptions in Section III. Section IV describes our proposed DeCODe framework in detail, and Section V provides the theoretical analysis of the proposed framework along with simulation-based validation. Finally, we conclude the paper in Section VI.

The research reported in this paper was supported in part by the Information Infrastructure Institute (iCube) of Iowa State University, and the NSF under Grants No. CNS 0520102 and No. CNS 0716744.

II. Related Work

Detection or coverage problem based on the 0/1 disc sensing model has been well studied in the past [1]–[3]. In such a model, an object inside (outside) a sensor's sensing range is detected with probability one (zero). Despite its simplicity for analysis, many researchers consider alternative sensing models [4]–[7], [9] in order to better understand and characterize sensor measurements which are usually affected by noise and vary with the distance between the sensor and the object. In this paper, we base our studies on the probabilistic sensing model.

A lot of research efforts [6], [10]–[13] have been made on collaborative object detection in wireless sensor networks, where the local data or decisions from individual sensors are gathered by a fusion center to make the final decision. However, none of these works considers the physical proximity between sensors and the object while in practice, as the received signal emitted by the object decays fast with distance, sensed readings by sensors far away from the object are less important to decision making. In [5], the authors introduce the concept of virtual sensor resulting from neighboring sensors' collaboration, which may improve the coverage for object detection. In contrast to our work, [5] assumes that sensors' locations are known and focuses on deriving a proper coverage set with less sensors to guarantee a required coverage. Due to the greedy and heuristic nature of the algorithm in [5] to find the coverage set, it is difficult to theoretically analyze the relation between the number of sensors deployed and the coverage performance.

In [14], the authors propose and design a new power management scheme using a radio-triggered hardware component to prolong the network lifetime. Equipped with a special radio-triggered circuit, a sleeping sensor (with both radio and CPU turned off) can be waken up by a special radio signal (transmitted at a different frequency from regular data communications) from a nearby active sensor.

978-1-4244-2026-1/08/$25.00

III. Models and Assumptions

A. System and Source Models

We consider a sensor network consisting of $(N^a + N^i$ wireless sensors deployed randomly over a unit-area conve region, where N^a is the number of active sensors and N^i i the number of inactive sensors. Each active sensor senses a a certain sampling frequency. In this paper, we assume that sensors make object detection decisions based on snapshot readings without considering temporal correlation; active sensors are called *sentry nodes* and inactive sensors are called *inert nodes*. There can be different models for the functioning of inert nodes. In one model, called the *message-based model*, inert nodes are only listening but not performing any sensing task. A sentry node can trigger its nearby inert nodes to start performing the sensing tasks by flooding a message carrying the triggering command. In another model, called the *circuit-based model*, inert nodes are sleeping and doing nothing. Under such a model, the sensors are provisioned with a special circuit for being triggered when need arises [14] and the triggering signal is transmitted at a different frequency from that being used for regular data communications.

We study the objects which emit physical signals such as sound and electromagnetic waves. The strength of the signal emitted by the object decays according to power law, meaning that the signal strength measured at distance d away from the object is: [5], [6]

$$\omega = \begin{cases} \Omega, & d < d_0, \\ \frac{\Omega}{(d/d_0)^\alpha}, & d \geqslant d_0, \end{cases} \tag{1}$$

where Ω is the signal amplitude of the object, d_0 is a small constant, and α is a known decay exponent. Since our analysis below can be applied to any decay exponent, we let α = 2 in this paper without loss of generality.

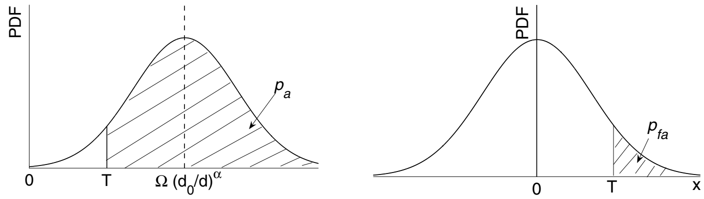
Assume that there is a single object in the region, which at any given time is either present or absent at a random location in the region according to certain probability distribution. Each active sensor collects its sensed reading of $\mathbf{x}$. Depending on the hypothesis of whether the object is present ($\mathcal{H}_1$) or not ($\mathcal{H}_0$), and the distance (d) between the object and the sensor if the object is present, sensed readings are:

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{x} = n, \\ \mathcal{H}_1 &: \mathbf{x} = \omega + n, \end{aligned} \tag{2}$$

where ω is the received signal strength given by (1) and n is the background noise. As in [5], we assume a Gaussian noise with zero mean and variance of one; indeed, our analysis can be applied to any noise model as long as its pdf is known.

B. Sensing and Alarm Models

In contrast to the 0/1 disc sensing model, we consider the probabilistic sensing model where (i) the sensor measurements are affected by noise; and (ii) based on a pre-determined decision threshold, a sensor detects an object with a probability that varies with distance between the sensor and the object. Assuming that a sensor is raising an alarm solely based on its own measurement ($\mathbf{x}$) and the decision threshold (T), the probability of genuine alarm (p_a) and false alarm (p_{fa}) raised by the sensor are shown as areas of shaded regions in Figs. 1(a) and (b), respectively.



(a) (b)

Fig. 1. (a) Probability of genuine alarm: $p_a = P(\mathbf{x} \geqslant T|\mathcal{H}_1)$ and (b) probability of false alarm: $p_{fa} = P(\mathbf{x} \geqslant T|\mathcal{H}_0)$, where $\mathbf{x}$ is the sensor measurement and T is the decision threshold.

IV. DeCODe: The Proposed On-Demand Framework for Collaborative Object Detection

In [8], we present and discuss several simple collaborative object detection mechanisms. Through analytical and simulation studies, we find that simple decision fusion-based collaborations among sentry nodes do not help improve the object detection performance. On the other hand, we suspect that better detection performance may be achieved if the collaboration among sensors is planned carefully. This motivates us to develop an on-demand framework for decision fusion-based collaborative object detection in wireless sensor networks. The key idea of our framework is that it no longer mandates sentry nodes to collaborate only with each other; instead it exploits the fact that, if a sentry node senses the object, its neighboring inert nodes may also be able to sense the object upon being triggered. This way, by leveraging on the inert nodes, we enjoy the same low false detection probability while increasing the probability of detection because the density of inert nodes is usually much higher than that of sentry nodes.

A. The Proposed Framework

Formally, our proposed framework is described as follows. Upon sensing a measurement higher than the decision threshold (T_K) where K is the collaboration degree, a sentry node triggers the neighboring inert nodes within its *fusion range* (a disc centered at the sentry node with radius of R_f) to collaboratively sense the environment. A collaboration degree of K means that, in order to report a detection of the object, a sentry node which initiates the detection process needs at least $(K-1)$ positive alarms from sensors within its fusion range. The decision threshold T_K varies with K, N^a_K (number of sentry nodes), and N^i_K (number of inert nodes). Our framework consists of the following three phases: *initialization*, *bounded flooding*, and *selective bouncing*.

1) Phase 1: Initialization: Upon sensing a measurement higher than the decision threshold, a sentry node initiates the collaborative detection process. If inert nodes function according to the message-based inert node model, the sentry node will enter *Phase 2* immediately after initialization. Otherwise, with the circuit-based inert node model, the sentry node will wait for an appropriate triggering time for the inert nodes within its fusion range to be triggered and then it enters *Phase 2*. The triggering time varies with the size of the fusion range (R_f) as well as the strength of the triggering signal. For example, with one of the radio-triggered circuits described in [14] to work with MICA2 motes, if the triggering signal strength is 10 dBm, it takes about 5 ms [14] to trigger the inert nodes within the fusion range of radius R_f = 30 feet. Thus, depending on the inert node model, after a certain time, the sentry node enters *Phase 2* to start the *flooding-and-bouncing* protocol to collect alarms from its neighboring inert nodes.

2) Phase 2: Bounded Flooding: The objective of this phase is to form a tree rooted at the sentry node, which will be used in *Phase 3* to collect positive alarms from sensors within the sentry node's fusion range. This is accomplished via flooding of *f_msg* and *ack_msg* messages. The flooding of *f_msg* messages is bounded within the fusion range of the sentry node by a TTL field carried in the *f_msg* message header. The initial TTL value is determined by the radius of the fusion range (R_f) and the average distance between neighboring nodes in the network. Upon reception of the first *f_msg* message with a positive TTL value, an inert node attaches itself to the tree by replying with an *ack_msg* message, then refreshes its sensed reading, if necessary, and records it in a local variable $iNode_j.Reading$.

After Algorithm 1 has been executed for sentry node x, almost all the inert nodes within its fusion range are attached to the tree. Each on-tree node ($iNode_j$) maintains the IDs of its parent node ($iNode_j.x.PID$) and children nodes ($iNode_j.x.ChildrenList$), as well as a Boolean variable ($iNode_j.x.BranchAlarm$) which indicates whether any of the nodes between itself and the sentry node has sensed a measurement higher than the decision threshold.

Fig. 2 gives an example of the proposed framework. Fig. 2(a) shows a sentry node and inert nodes within its fusion range before flooding-and-bouncing. The formed tree after execution of Algorithm 1 is shown in Fig. 2(b), where black/white dots represent inert nodes with sensed readings higher/lower than the decision threshold, and cross dots represent inert nodes whose own readings are lower than the decision threshold but lie along the branches on which at least one node has sensed a reading higher than the decision threshold.

Algorithm 1 Bounded Flooding

```
For each sentry node sNode_i (with node ID i)
Initialization:
  Determine TTL based on K, N_K^a and N_K^i
Upon sensing a reading higher than decision threshold T_K:
  Start timer_c(i) with interval τ                 /* τ is a system parameter */
  sNode_i.ChildrenList ← ∅
  sNode_i.sumAlarm ← 1                   /* to be used during Selective Bouncing */
  sNode_i.ReportList ← ∅                 /* to be used during Selective Bouncing */
  Broadcast a flooding message: f_msg⟨SID = i, NID = i, TTL, bAlarm = 0⟩
Upon receiving an acknowledgment message ack_msg⟨SID, PID, NID⟩:
  if (ack_msg.SID == i) AND (ack_msg.PID == i) then
      sNode_i.ChildrenList ← sNode_i.ChildrenList ∪ ack_msg.NID
  end if
Upon timer_c(i) is fired:
  Clear sNode_i data structure

For each inert node iNode_j (with node ID j)
Initialization:
  iNode_j.SentryList ← ∅
  iNode_j.Need2Sense = 1
Upon receiving a flooding message f_msg⟨SID, NID, TTL, bAlarm⟩:
  x ← f_msg.SID
  if (x ∉ iNode_j.SentryList) then
      Start timer_b(x) with interval δ             /* δ is a system parameter */
      Start timer_c(x) with interval τ             /* τ is a system parameter */
      iNode_j.SentryList ← iNode_j.SentryList ∪ x
      iNode_j.x.ReportList ← ∅           /* to be used during Selective Bouncing */
      iNode_j.x.BounceDone ← 0           /* to be used during Selective Bouncing */
      iNode_j.x.PID ← f_msg.NID
      iNode_j.x.ChildrenList ← ∅
      iNode_j.x.LocalAlarm ← 0
      iNode_j.x.BranchAlarm ← f_msg.bAlarm
      if (iNode_j.Need2Sense == 1) then
          Sense and store its own reading at iNode_j.Reading
          iNode_j.Need2Sense = 0
      end if
      if (iNode_j.Reading ⩾ T_k) then
          iNode_j.x.LocalAlarm ← 1
          iNode_j.x.BranchAlarm ← 1
      end if
      iNode_j.x.sumAlarm ← iNode_j.x.LocalAlarm      /* to be used during Selective
                                                                       Bouncing */
      Broadcast an acknowledgment message: ack_msg⟨SID = x,
                                        PID = iNode_j.x.PID, NID = j⟩
      if (f_msg.TTL > 1) then
          Broadcast a flooding message: f_msg⟨SID = x, NID = j,
                        TTL = f_msg.TTL − 1, bAlarm = iNode_j.x.BranchAlarm⟩
      else
          Execute Algorithm 2: Selective Bouncing
      end if
  end if
Upon receiving an acknowledgment message ack_msg⟨SID, PID, NID⟩:
  x ← ack_msg.SID
  if (x ∈ iNode_j.SentryList) AND (ack_msg.PID == j) then
      iNode_j.x.ChildrenList ← iNode_j.x.ChildrenList ∪ ack_msg.NID
  end if
Upon timer_b(x) is fired:
  Execute Algorithm 2: Selective Bouncing
Upon timer_c(x) is fired:
  iNode_j.SentryList ← iNode_j.SentryList − x
  iNode_j.Need2Sense = 1
  Clear iNode_j.x data structure
```

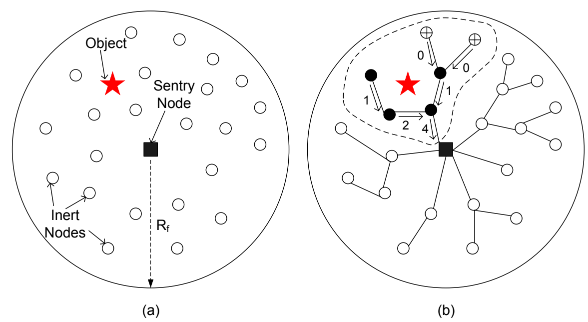


Fig. 2. The proposed DeCODe framework. The object is shown as the star, the sentry node is shown as the black square, and circles represent inert nodes. (a) Network topology before flooding-and-bouncing. (b) During *Phase 3: Selective Bouncing*, bouncing messages are initiated by leaf nodes and routed towards the sentry node along the sub-tree inside the dash-curve-bounded region; numbers along the edges are the $iNode_j.x.sumAlarm$ values reported by the corresponding inert nodes.

3) Phase 3: Selective Bouncing: The objective of this phase is to collect positive alarms from relevant sensor nodes and propagate them to the sentry node. This is accomplished via *b_msg* messages. As shown in Algorithm 2, only leaf nodes who either have sensed a measurement higher than the decision threshold or belong to a branch on which at least one node has sensed a measurement higher than the decision threshold can initiate the bouncing process. For example, the bouncing process in Fig. 2(b) is initiated by the leaf nodes inside the dash-curve-bounded region. An inert node relays the bouncing message after hearing from all of its children, and indicates in the message the total number of positive alarms raised by nodes belonging to the subtree rooted at itself: $b\_msg.nAlarm = iNode_j.x.sumAlarm$. Such bouncing process is expedited when an inert node has collected adequate (i.e., $\geqslant K-1$) positive alarms for the sentry node to report a detection. Each inert node also maintains the latest *sumAlarm* values reported by its children. This enables the node to update its *sumAlarm* values in case one of its children nodes reports

Algorithm 2 Selective Bouncing

```
For each inert node iNode_j (with node ID j)
Initialization:
for (Each sentry node x ∈ iNode_j.SentryList) do
    if (iNode_j.x.ChildrenList == ∅) AND (iNode_j.x.BranchAlarm == 1) then
        Broadcast a bouncing message: b_msg⟨SID = x, PID = iNode_j.x.PID,
                                      NID = j, nAlarm = iNode_j.x.sumAlarm⟩
        if (iNode_j.x.sumAlarm ⩾ K) then
            iNode_j.x.BounceDone ← 1
        end if
    end if
end for

Upon receiving a bouncing message b_msg⟨SID, PID, NID, nAlarm⟩:
x ← b_msg.SID
y ← b_msg.NID
if (x ∈ iNode_j.SentryList) AND (iNode_j.x.BounceDone == 0)
   AND (b_msg.PID == j) then
    if (y ∉ iNode_j.x.ReportList) then
        iNode_j.x.ReportList ← iNode_j.x.ReportList ∪ y
        iNode_j.x.sumAlarm ← iNode_j.x.sumAlarm + b_msg.nAlarm
        iNode_j.x.nAlarm_y ← b_msg.nAlarm
    else
        if (iNode_j.x.nAlarm_y < b_msg.nAlarm) then
            iNode_j.x.sumAlarm ← iNode_j.x.sumAlarm − iNode_j.x.nAlarm_y
                                 +b_msg.nAlarm
            iNode_j.x.nAlarm_y ← b_msg.nAlarm
        end if
    end if
    if (iNode_j.x.ReportList == iNode_j.x.ChildrenList)
       OR (iNode_j.x.sumAlarm ⩾ K) then
        Broadcast a bouncing message: b_msg⟨SID = x,
              PID = iNode_j.x.PID, NID = j, nAlarm = iNode_j.x.sumAlarm⟩
        if (iNode_j.x.sumAlarm ⩾ K) then
            iNode_j.x.BounceDone ← 1
        end if
    end if
end if

For each sentry node sNode_i (with node ID i)
Upon receiving a bouncing message b_msg⟨SID, PID, NID, nAlarm⟩:
y ← b_msg.NID
if (b_msg.SID == i) AND (b_msg.PID == i) then
    if (y ∉ sNode_i.ReportList) then
        sNode_i.ReportList ← sNode_i.ReportList ∪ y
        sNode_i.sumAlarm ← sNode_i.sumAlarm + b_msg.nAlarm
        sNode_i.nAlarm_y ← b_msg.nAlarm
    else
        if (sNode_i.nAlarm_y < b_msg.nAlarm) then
            sNode_i.sumAlarm ← sNode_i.sumAlarm − sNode_i.nAlarm_y + b_msg.nAlarm
            sNode_i.nAlarm_y ← b_msg.nAlarm
        end if
    end if
    if (sNode_i.sumAlarm ⩾ K) then
        Report a detection
        return true
    end if
end if
```

multiple times. This situation may occur because, if the tree is unbalanced, it is possible (though not likely) that the leaf nodes along the shorter branches have already started bouncing while the formation of the longer branches has not yet completed.

Not shown in Algorithms 1 and 2 are how a sentry node responds to the *f_msg* messages from other sentry nodes and how it participates in selective bouncing. In such situations, the sentry node acts exactly like an inert node except that it uses its most recently-sensed reading to participate in the decision process, instead of performing an additional sensing upon reception of the *f_msg* message.

V. Theoretical Analysis and Simulation-based Validation of the Proposed Framework

In this section, we give details for calculating the false detection probability (P_{FD}) and the detection probability (P_D). Before proceeding to that, we first introduce the concepts of *detection zone* and *fusion range* for the proposed framework.

A. Detection Zone and Fusion Range

We define detection zone (D.Z.) to be a disc centered at the object and with a radius R_d. If the distance between a sensor and the object is larger than R_d, the probability of the sensor's measurement exceeding the decision threshold is less than a very small number (we use 1% in this paper) hence negligible. On the other hand, if a sensor is within R_d from the object, the probability of its measurement exceeding the decision threshold cannot be neglected and varies with the distance between them. Given the definition of D.Z., we define fusion range to be a disc centered at a sentry node and with a radius of $R_f = 2R_d$. Such definition of fusion range guarantees that, whenever a sentry node is within D.Z. of the object and senses a measurement above the decision threshold (which itself is a high-probability event), all inert nodes within D.Z. will be triggered, as shown in Fig. 3.

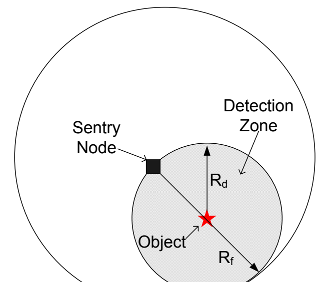


Fig. 3. Illustration of detection zone and fusion range.

B. The Calculation of P_{FD} and P_D

When there is no object in the network, a sensor's reading is only affected by noise. The false detection probability P_{FD} is the probability that at least one sentry node in the network reports a false detection:

$$P_{FD}(K, N_K^a, N_K^i) = P(\text{detection}|\mathcal{H}_0) = 1 - (1 - \mathbb{P}_{FA})^{N_K^a}, \tag{3}$$

where $\mathbb{P}_{FA}$ is the probability that a sentry node reports a false detection for a collaboration degree K. N_K^a is the number of sentry nodes. Clearly, when $K = 1$, $\mathbb{P}_{FA} = p_{fa} = \mathrm{erfc}(T_1)$, where T_1 is the corresponding decision threshold, as shown in Fig. 1(b). For $K > 1$, sensors collaborate and $\mathbb{P}_{FA}$ is given by:

$$\begin{aligned}\mathbb{P}_{FA} &= \mathrm{erfc}(T_K) \cdot P(\text{at least } K-1 \text{ sensors within fusion range raise alarms}) \\ &= \mathrm{erfc}(T_K) \cdot (1 - P_b),\end{aligned} \tag{4}$$

where

$$\begin{aligned}P_b &= P(\text{at most } K-2 \text{ sensors raise alarms}) \\ &= \sum_{m=0}^{N_K^i + N_K^a - 1} \frac{(\lambda\|\text{D.Z.}\|)^m e^{(-\lambda\|\text{D.Z.}\|)}}{m!} \times \\ &\quad \left(\sum_{n=0}^{\min(K-2,m)} \binom{m}{n} (\mathrm{erfc}(T_K))^n (1 - \mathrm{erfc}(T_K))^{m-n} \right),\end{aligned} \tag{5}$$

where $\lambda = N_K^i + N_K^a$ is the node density in the network. Note that we treat distinct sentry nodes reporting false detections as independent events. This is reasonable because, due to the low P_{FD} usually required by the system, sentry nodes that report false detection are likely far away from each other, hence their fusion ranges seldom overlap.

The calculations of P_D and P_{FD} are related since both vary with K, T_K, N_K^a and N_K^i. Next we describe the calculation details for $P_D(K, N_K^a, N_K^i)$ when the target P_{FD}^* is given. The probability of detection is the conditional probability that given the object is present, at least one sentry node reports a detection. Recall that the probability that a sentry node outside D.Z. recording a detection is very low. Therefore, we have:

$$\begin{aligned} &P_D(K, N_K^a, N_K^i) = P(\text{detection}|\mathcal{H}_1) \\ &\gtrapprox \sum_{n=K}^{N_K^a+N_K^i} P(\text{n sensors inside D.Z.}) \times \\ &\sum_{m=K}^{n} P(\text{m sensors raising alarms upon triggered by a sentry node inside D.Z.}) \\ &= \sum_{n=K}^{N_K^a+N_K^i} \frac{(\lambda\|\text{D.Z.}\|)^n e^{(-\lambda\|\text{D.Z.}\|)}}{n!} \times \\ &\sum_{m=K}^{n} \left(\binom{n}{m} (\mathbb{P}_A)^m (1-\mathbb{P}_A)^{n-m} \right) \left(1 - \left(\frac{N_K^i}{N_K^a + N_K^i} \right)^m \right), \end{aligned} \tag{6}$$

where $\lambda = N_K^a + N_K^i$ is the node density in the network. $\mathbb{P}_A$ is the probability that a sensor within D.Z. has a measurement higher than T_K:

$$\mathbb{P}_A = \int_0^{\frac{R_f}{2}} \frac{2\pi r}{\pi \left(\frac{R_f}{2}\right)^2} \times \operatorname{erfc}\left(T_K - \frac{\Omega d_0^2}{r^2}\right) dr. \tag{7}$$

C. Simulation-based Validation

In this section, we conduct numerical and simulation studies to support our previous theoretical analysis. First, we study the performance of our proposed framework in terms of detection probability with respect to N_K^a, N_K^i, T_K for various K. For a fixed $N_K^a + N_K^i$ (i.e., total number of sensors deployed) and a target false detection probability $P_{FD}^* = 0.001$, Fig. 4 shows P_D for different K with respect to varying N_K^a. We observe that for a fixed N_K^a, P_D increases with K. P_D also increases with increase in N_K^a. However, for a fixed N_K^a, the performance improvement is not significant for higher degree of collaboration, e.g., K increasing from 3 to 4.

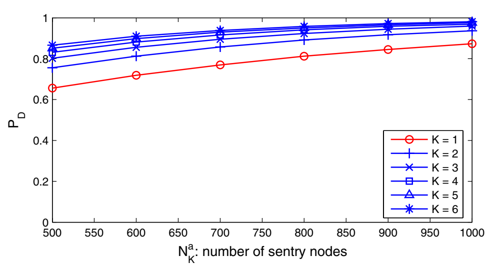


Fig. 4. Detection probability (P_d) vs. number of sentry nodes (N_K^a) when $P_{FD}^* = 0.001$ and $N_K^a + N_K^i = 4000$.

Fig. 5(a) shows the variation of decision threshold T_K (normalized with respect to Ω, signal amplitude of the object) with K for fixed P_{FD}^*, N_K^a and N_K^i. Corroborating our analysis, we observe that T_K decreases with increase in K. This means that, with a higher collaboration degree, each sentry node will have a larger D.Z., and therefore would trigger more inert nodes for collaboration.

To verify the validity of our numerical analysis, we simulate our proposed framework with the following setup. We deploy 4000 nodes randomly in a unit area, out of which 1000 sentry nodes regularly sense the environment. We test 10 different deployments for evaluating the detection probability and the

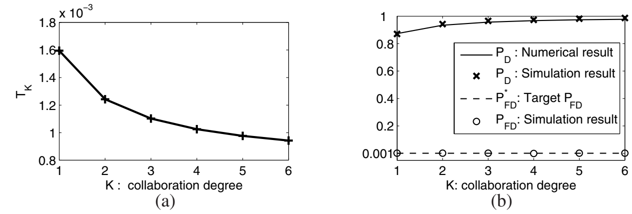


Fig. 5. (a) Normalized decision threshold (T_K) vs. degree of collaboration (K) when $P_{FD}^* = 0.001$, $N_K^a = 1000$, and $N_K^i = 3000$. (b) Comparison of numerical and simulation results when $P_{FD}^* = 0.001$, $N_K^a = 1000$, $N_K^i = 3000$, $\Omega = 2100$ mW, and $d_0 = 0.001$ units.

false detection probability. For simulating the detection probability, we randomly choose 40 different locations for the object and simulation is repeated 100 times for each object location. We evaluate the false alarm probability based on 10000 trials. Results plotted in Fig. 5(b) show a close correspondence between numerical and simulation results. Moreover, since decision fusion incurs extra energy consumption in triggering neighboring inert nodes and aggregating collaborative messages, we further investigate the energy-efficiency performance of the proposed framework and please refer to [8] for details.

VI. Conclusions and Future Work

In this paper, we propose a novel on-demand framework for collaborative object detection in wireless sensor networks based on the probabilistic sensing model. We leverage on the collaboration between sentry nodes and inert nodes to improve the system performance in terms of detection probability and false detection probability. We provide the theoretical analysis of our proposed framework, and support it with simulation-based validation. In the future, we would extend our framework to operate in the presence of multiple objects or events involving diffusion phenomenon. It would also be interesting to explore the advantages of temporal correlation and investigate our framework for mobile sensors and objects.

References

[1] X. Bai, S. Kuma, D. Xua, Z. Yun, and T. H. Lai, "Deploying Wireless Sensors to Achieve Both Coverage and Connectivity," *in Proc. ACM MobiHoc*, May 2006.

[2] Q. Cao, T. Yan, J. Stankovic, and T. Abdelzaher, "Analysis of Target Detection Performance for Wireless Sensor Networks," *in Proc. IEEE DCOSS*, June 2005.

[3] O. Dousse, C. Tavoularis, and P. Thiran, "Delay of Intrusion Detection in Wireless Sensor Networks," *in Proc. ACM Mobihoc*, May 2006.

[4] B. Liu and D. Towsley, "A Study of the Coverage of Large-scale Sensor Networks," *in Proc. IEEE MASS*, Oct. 2004.

[5] W. Wang, V. Srinivasan, K. C. Chua, and B. Wang, "Energy-efficient Coverage for Target Detection in Wireless Sensor Networks," *in Proc. IPSN*, Apr. 2007.

[6] T. Clouqueur, K. K. Saluja, and P. Ramanathan, "Fault Tolerance in Collaborative Sensor Networks for Target Detection," *IEEE Transactions on Computers*, vol. 53, no. 3, Mar. 2004.

[7] T.-L. Chin, P. Ramanathan, and K. K. Saluja, "Analytical Modeling of Detection Lantency in Mobile Sensor Networks," *in Proc. IPSN*, 2006.

[8] Tech. Rep., http://www.public.iastate.edu/~gqyang/decode-analysis.pdf.

[9] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks," *in Proc. ACM MobiCom*, 2001.

[10] P. Varshney, *Distributed Detection and Data Fusion*. Springer, 1997.

[11] R. Viswanathan and P. Varshney, "Distributed Detection with Multiple Sensors I. Fundamentals," *in Proc. the IEEE*, vol. 85, no. 1, Jan. 1997.

[12] R. Blum, S. Kassam, and H. Poor, "Distributed Detection with Multiple Sensors I. Advanced Topics," *in Proc. the IEEE*, vol. 85, no. 1, Jan. 1997.

[13] X. Luo, M. Dong, and Y. Huang, "On Distributed Fault-tolerant Detection in Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 55, no. 1, Jan. 2006.

[14] L. Gu and J. Stankovic, "Radio-Triggered Wake-Up Capability for Sensor Networks," *in Proc. IEEE RTAS*, Apr. 2004.