Probabilistic Approach to Provisioning Guaranteed QoS for Distributed Event Detection

Yanmin Zhu†‡ and Lionel M. Ni†
†HK University of Science and Technology; ‡Imperial College London
yzhu@doc.ic.ac.uk, ni@cse.ust.hk

Abstract—It has been of significant importance to provision network-wide guaranteed QoS for a wide range of event detection applications in wireless sensor networks (WSNs). This paper investigates solutions to this QoS provision problem. For event detection applications, there are two key performance metrics, i.e., detection probability and detection latency. This paper focuses on dual-objective QoS provision, taking both metrics into account. This is very challenging due to the stringent resource constraint of sensor nodes and unpredictable randomness of physical events. We propose a novel probabilistic approach to provisioning due-objective QoS. Following a unified framework, we design a distributed algorithm that determines the active probability of every sensor node. The probability is minimized while being sufficient for QoS provision. Our approach is flexible and supports different requirements that may be posed by different applications. Theoretical analysis and comprehensive simulation experiments have been conducted, which jointly demonstrate that our approach is able to deliver guaranteed QoS for distributed event detection while prolonging the system lifetime significantly compared with other alternative schemes.

I. INTRODUCTION

Wireless sensor networks (WSNs) have been employed for various event detection applications whose central task is to detect physical events of interest, such as wild fire, gas leakage and radiation. There are two key performance metrics for event detection, detection latency and detection probability. Quality of Service (QoS) is an important concept that defines the performance constantly delivered by a system. Moreover, this performance is directly perceived by end users. Event detection applications can be diverse in the real world and different applications have varying QoS requirements. It is clear that more sensitive applications usually need higher detection QoS.

It has been of paramount importance to provision guaranteed QoS for distributed event detection over the whole application field. Such QoS requirements should be customized by end users. This allows the system to avoid over-provision of detection QoS and to optimize resource utilization of the sensor network. With the provision of guaranteed QoS, the users can safely expect the performance of event detection and make presumption about the derived data from the system. This paper focuses on dual-objective QoS provision taking both the important metrics into account for distributed event detection.

Sensor networks are distinct from traditional computer networks. Small sensor nodes are particularly resource constrained as they have small memory, limited processing capability and low bandwidth. Most importantly, each sensor node is typically powered by a battery having a short lifetime. However, real-world applications require the network to sustain monitoring operations for a long duration. This implies that designing a long-lived network with tiny short-lived sensor nodes is very difficult. In this paper, we explore energy-efficient design for provisioning guaranteed QoS of event detection in distributed sensor networks.

Many research attempts have been made to reduce energy consumption of sensor nodes and thus to prolong system lifetime. A lot of principles from them can benefit design of energy-efficient sensor networks for event detection. Many algorithms select a subset of sensor nodes to keep vigilant for event detection and put the others in power-save mode for energy conservation, such as PEAS [1], sponsored coverage [2] and differentiated scheduling [3]. They typically aim at providing full coverage in both the spatial and the temporal dimensions. In their schemes, the detection of any event is instantly finished and no latency is incurred. In addition, any event can be detected as long as the field is fully covered.

Unfortunately, these existing schemes suffer from several problems when they are applied to QoS provision. First, the algorithms fail to support flexible QoS and typically over-provision more QoS for event detection than is required by the application. This is undesirable for many realistic applications and leads to waste of precious energy. Second, the algorithms suffer from the serious problem of unbalanced energy consumption. According to the existing algorithms, a set of sensor nodes are selected to stay active for full sensing coverage while the other sensor nodes turn to power-save mode. The consequence is that the selected active sensor nodes will be depleted earlier. If an active sensor node becomes unavailable due to power depletion or physical damage, the area covered by this sensor node will become a blind spot.

To overcome the limitations of the existing schemes, we propose a novel probabilistic approach to dual-objective QoS provision for distributed event detection. The approach allows the sensor nodes to stay active probabilistically for effective energy conservation. Following a unified framework, we design a completely distributed algorithm to determine the active probability, the important configuration parameter of every sensor node. The active probability is adaptive to its neighborhood of sensor deployment. Energy consumption is minimized and well balanced across the network. At the same time, however, network-wide QoS of event detection is dynamically maintained.
To the best of our knowledge, no existing algorithms can deliver guaranteed QoS for distributed event detection. The original contributions we have made in this paper are highlighted in the following. First, we have thoroughly investigated the issue of provisioning detection QoS. We propose a novel probabilistic approach, which is energy efficient, to realize such provisioning of QoS. Second, we have developed a distributed algorithm Loya that determines the active probability. Not relying on costly time synchronization, this algorithm is lightweight and fully distributed, supporting high scalability with network scale and sensor density. Finally, we have conducted both theoretical analysis and comprehensive experiments to validate the design and study the performance of the algorithm in comparison with other schemes.

The remainder of the paper is structured as follows. The following section discusses related work. Section III introduces the provisioning of detection QoS in sensor networks and formulates the problem. In Section IV, the distributed algorithm is described and algorithm analysis is presented. Some discussions are given in Section V. Simulation results and analysis are given in Section VI. Finally, we give concluding remarks in Section VII.

II. RELATED WORK

This section discusses related work in the literature for energy conservation and event detection in sensor networks. This paper essentially proposes a power management protocol in the specific context of event detection, taking the characteristics of event detection into account. In the literature of sensor networks, there have been many contributions for power management in different kinds of applications.

A sensor node basically has two states, power-on and power-off state. The most important method to conserve energy is to power off a sensor node and hence to reduce the effective duty cycle [4]. Asynchronous wakeup [5] is advantageous since no time synchronization is required. However, additional latency is introduced into packet transmission. Kesavarzian et al [6] study the tradeoffs that exist between transmission latency and energy efficiency. In the application of object tracking, there are different tradeoffs between energy saving and tracking quality [7, 8]. In [9], probabilistic coverage has been used to track objects, which allows sensor nodes to stay active randomly.

The closest work to our work are those [10-12, 3, 1] proposed for maintaining sensing coverage over a given field. This body of work generally targets at regular monitoring. Thus, one important issue is to make sure that the whole field is fully covered by sensor nodes. To be energy efficient, it is intuitive that a minimum number of sensor nodes should be selected to stay in active mode, which are responsible for continuously monitoring the whole field. PEAS [1] proposes a probing technique. By broadcasting a probe packet, a sensor node waits for a reply from active neighbors. Upon reception of a reply packet, the node goes to sleep for power conservation. Tian and Georganas exploit geometric calculation to determine whether a sensor node is eligible for being turned off. In essence, if the sensing coverage is covered by its active neighbors, a sensor node is safe to be turned off. In [12], the set of selected active nodes can provide different degrees of sensing coverage. Meanwhile, these nodes are guaranteed to be connected. A randomized algorithm [3] is proposed to determine the active schedules such that at any time the full coverage is always maintained. Shakkottai et al. [11] extend to study the coverage that can be achieved by a sensor network when sensor nodes are unreliable. Although these works can also be applied for power management in event detection, they fail to take the characteristics of event detection into account. Furthermore, different applications may require varying QoS requirements. These works always provide the highest detection quality and cannot further conserve energy when a lower QoS is required.

There are several attempts [13, 14] made to detect events using a low duty-cycled sensor nodes for higher energy efficiency. In [14], a sensor node platform called eXtreme Scale Mote was developed which considers the detailed operations of detecting an event. However, it focuses only on an individual node but does not consider network management when there are many nodes. Cao et al. [13] focuses on the scheduling of wakeup times of sensor nodes. The objective is to minimize the detection latency. Our paper however, considers another important aspect of event detection, i.e., the provisioning of detection QoS.

Many research efforts [15-19] have been made for energy efficient communications in WSNs. MAC protocols [15, 18, 19] provide the foundation for network operations which usually manage the radio transceiver for energy conservation in consideration that wireless communication is the main source of energy consumption. Clustering [17] is an effective technique to organize the network from the viewpoint of communication where a cluster head collects data in the cluster and transfers them to a remote sink. Felemban et al. [16] propose a protocol that guarantees the timeliness and reliability of packet transmission in sensor networks. Our paper does not fix the communication protocol for event delivery and these existing works are complementary to our work.

III. DETECTION QoS PROVISION AND PROBLEM FORMULATION

In this section, we first describe the system background and then introduce some concepts and formulate the problem of QoS provision for distributed event detection.

A sensor network $S=\{1, 2, 3, \ldots, n\}$ is deployed in a square field $F$ with side length $h$ for event detection. The deployment of sensor node is assumed to be uniformly random. When the number of sensor nodes is large, a two-dimensional Poisson process with rate $n/h^2$ can approximate the node deployment.

A sensor node can be attached with several sensing devices, each able to sense a certain environmental parameter. Based on the sensing readings from the sensing device, a sensor node can detect a physical event. For example, with an acoustic sensor, a sensor node can detect the presence of a
wide animal. In practice, there is usually an effective range within which the sensing device can read meaningful sensory data, which is termed as detection range (denoted by $r$). In this paper we assume a binary detection model that a sensor node can always detect an event as long as the event falls in the detection range of the sensor node.

Location information is of great importance in the application realm of event detection. When an important event is detected, such an event should be tagged with the location information indicating where this event happened. Otherwise, the network operators have no means to locate this event. We assume that each node $i$ is aware of its physical location $(x_i, y_i)$. The position information can be discovered by GPS or localization algorithms [20].

Physical events can arise anytime anywhere in the sensing field. It is usually difficult to predict when and where the next event will happen. We consider that events arise uniformly and randomly over the field. In addition, the arrival of events follows a Poisson process in the time dimension.

The system goal is to deliver event detection service with guaranteed QoS. This paper focuses on provisioning dual-objective detection QoS, which consists of the maximum detection latency and the minimum detection probability. Any event that occurs within the field should meet the dual-objective QoS requirement. With such QoS guarantee, the network operators can safely predict the properties of the data obtained from the sensor network. In addition, for different applications with varying quality requirements, the sensor network can precisely provide the requested QoS and effectively minimize resource usage.

Throughout the network lifetime, the sensor network should guarantee the dual-objective QoS. On the one hand, the detection probability of any event should be larger than a specified minimum $\delta_0$. On the other hand, the detection latency of any event should be constrained. The detection delay of an event is a random variable dependent on the arrival time of the event, the number of nodes covering the event, and the working schedules of these nodes. It is ideal that the system provides a hard bound for detection delay, i.e., the detection latency is always less than a given value. However, this compels sensor nodes to wake up regularly, which prevents the system from conserving precious energy.

Instead, we aim at providing a soft bound, which is valuable for practical real-world applications. More specifically, the user specifies the longest delay $\mathcal{L}_0$ that is characterized by a cumulative distribution function (cdf). For example, the user can require that with 30% probability, the detection latency is within 1s, with 50% probability within 2s and with 80% probability within 3s. This actually characterizes a random variable $\mathcal{L}$. Let $F_\mathcal{L}(d)$ denote the cdf of this random variable. The system should then ensure that the detection latency of any event is constrained by $\mathcal{L}_0$ according to the following definition.

**Definition 1 (latency constrained):** Detection latency $\mathcal{L}_i$ is constrained by $\mathcal{L}_2$, denoted by $\mathcal{L}_1 \leq \mathcal{L}_2$, if

$$F_{\mathcal{L}_1}(d) \geq F_{\mathcal{L}_2}(d), \forall d > 0.$$  \hspace{1cm} (1)

where $\mathcal{L}_1$ and $\mathcal{L}_2$ share the same domain.

To specify the QoS requirement on detection latency, the user can easily set the cdf of $\mathcal{L}_0$.

To realize the provisioning of dual-objective QoS, we propose a probabilistic approach and two additional components: a unified metric and a distributed algorithm. The metric design is to be introduced shortly and the algorithm is described in the next section. Following the probabilistic approach, a sensor node probabilistically stays active. At any time, a node $i$ is active with probability $\delta_i$ and in power-save mode with probability $1-\delta_i$. Intuitively, a sensor node is more vigilant for event detection when its active probability is higher. The detection performance is therefore increased. At the same time, however, its power consumption becomes much higher. Using such a probabilistic approach makes it possible that sensor nodes are adaptive to the QoS requirement. In addition, it provides a unified framework to tune sensor nodes while detection QoS being guaranteed.

Since events may arise at any place within the field, we investigate the relation between the detection performance of this event and the aggregate activity of sensor nodes at the event point where this event arises. We find that the performance is solely determined by the aggregate probability of the active probabilities of the neighboring sensor nodes. To this end, we propose a metric called aggregate activity to facilitate this analysis.

**Definition 2 (aggregate activity):** At a given time $t$, the aggregate activity of a point $q$ in $F$, denoted by $\phi(q, t)$, is the probability that $q$ is within the detection range of at least one active sensor node.

If the aggregate activity of the point is invariant with respect to time, it is simplified to $\phi(q)$. The aggregate activity of $q$ is dependant on several factors including the number of covering nodes and the active probabilities of these nodes. By definition, the aggregate activity of $q$ is given by,

$$\phi(q) = 1 - \prod_{i \in S(q)} (1 - \delta_i) .$$  \hspace{1cm} (2)

where $S(q)$ denotes the set of sensor nodes whose detection coverage contains $q$.

The aggregate activity of a point solely characterizes the detection performance of any event occurring at this point in a unified way. We present the following analysis for this
characterization, based on the example illustrated Figure 1. In essence, the detection probability and the detection latency of an event are functions of the aggregate activity at the event point.

The detection probability of an event is determined by both the event lifetime and the aggregate activity of the event. Let denote the lifetime of event and the minimum necessary time required for a sensor node to detect and process an event. Then, the detection probability of is calculated by

\[
\zeta_e = 1 - \left[1 - \phi(p_e)\right]^{\tau_e/\tau_0}. \tag{3}
\]

Let denote the number of time slots before the event is detected by a sensor node. is a random variable. The detection latency of this event can be calculated,

\[
L_e = M \times \tau, \tag{4}
\]

The probability mass function (pdf) of is given by

\[
\Pr\{M = i\} = \phi(p_e)^i \left[1 - \phi(p_e)\right]. \tag{5}
\]

Then it is easy to find the pdf of ,

\[
\Pr\{L_e = d\} = \left[1 - \phi(p_e)\right]^k \phi(p_e), \tag{6}
\]

where \(k = d / \tau_0\).

The above analysis leads to the following theorem.

**Theorem 1:** The detection QoS of an event at a location is characterized by the aggregate activity of the location.

With Theorem 1, we guarantee the detection QoS at a location as long as the aggregate activity of this location is sufficiently large such that the corresponding detection probability is larger than the requested and the corresponding detection latency is constrained by . The key issue here is the computation of the minimum necessary aggregate activity. For detection probability, we have the constraint,

\[
\zeta_e = f(\phi(q_e)) \geq \zeta_e. \tag{7}
\]

For detection latency, we have the constraint,

\[
L_e = g(\phi(q_e)) \leq L_0. \tag{8}
\]

We can calculate the minimum aggregate activity such that constraint (7) and (8) are satisfied. Note that this calculation can be performed offline at a powerful computer and then the result is used in sensor nodes.

Now it becomes apparent that to deliver network-wide detection QoS, we need to ensure that the aggregate activity of any location point within the field is greater than the calculated minimum aggregate activity. That is,

\[
\phi(q) \geq \phi_0, \forall q \in F. \tag{9}
\]

Besides the important issue of detection QoS provisioning, we also need to deal with another critical issue, i.e., power management. It is straightforward that detection QoS can be easily provided if the active probability of every sensor node is set to a high value (e.g., all set to ) However, it is energy inefficient considering that sensor nodes can be densely deployed in most practical sensor network applications. Thus, it is crucial to minimize the active probability of sensor nodes while at the same time the system should continuously guarantee the provisioning of the requested detection QoS. Ideally, the system delivers exactly the required QoS. On the other hand, as previously discussed, the power consumption of sensor nodes should be balanced.

**IV. DESIGN OF DISTRIBUTED ALGORITHM**

We propose the distributed algorithm Loya to determine the critical parameter of each sensor node, i.e., the active probability. In this section, we first give an overview of the algorithm and then elaborate on the design details. Finally, we present algorithm analysis.

Throughout the design of the algorithm, we consider the following objectives important.

- **High scalability.** A large number of sensor nodes can be involved in a sensor network. This suggests that the sensor system should be scalable with the increasing number of sensor nodes.
- **Synchronization free.** It has been a great challenge for a sensor network to be time synchronized. A lot of efforts have been made and several time synchronization protocols have been available. However, significant overhead such as power consumption is introduced. Therefore, an algorithm free from time synchronization is very advantageous and practical.

The algorithm executes after the sensor network is deployed. After the algorithm terminates, each sensor node is assigned an active probability. Next, all sensor nodes wakes up randomly according to the active probability.

From (9), the sensor nodes should jointly make sure any point in the field has an aggregate activity not less than . It is almost impossible to check every point in the field. To solve this problem, we divide the field into virtual grids and consider the grid points. The grid granularity is an important design parameter. We shall discuss it later.

**A. Algorithm Description**

By running this algorithm, every sensor node determines its active probability cooperatively with its nearby sensor nodes. The cooperation is important and necessary since nearby sensor nodes jointly monitor a certain common area. Each sensor node only needs to cooperate with the set of nearby sensor nodes that are within the distance of , because a sensor node further than has no overlapped detection coverage with it. We call such a sensor node in the set a stakeholder.

The important steps and operations of the algorithm are outlined in Algorithm 1.

The first key issue is to start the sensor network to execute the algorithm in the absence of restrict time synchronization. To address this, the sink nodes propagate a signal packet to all sensor nodes. On receiving a signal packet, a sensor node starts timer . The length of this timer should be sufficient
for the signal packet being propagated to the farthest network edge from the sink nodes. In addition, it should accommodate the time for sensor nodes to exchange information, which enables every sensor node to discover its stakeholders.

After the timer fires, each sensor node starts to compute the active probability. This initial active probability guarantees that the aggregate activity of any grid point in the node’s detection coverage is not less than $\phi$. On the other hand, the power consumption of sensor nodes should be balanced, which means that active probabilities are better uniformly distributed. However, for each sensor node, its active probability should be adaptive to its neighborhood. It is intuitive that if there are more stakeholders, a sensor node can have a smaller active probability for energy conservation.

Each sensor node computes the necessary probability for every grid point $q$ within its detection range, denoted by $\delta(q)$, and then chooses the maximum as its active probability, i.e., $\delta_i$. The maximum one ensures that every grid point meets the constraint on its aggregate activity.

Consider point $q$, its aggregate activity is

$$\phi(q) = 1 - \prod_{q \in S(q)} (1 - \delta_i).$$  \hfill (10)

This aggregate activity is greater than or equal to $\phi$. For the purpose of energy balance, we let the sensor nodes in $S(q)$ take equal responsibility. Thus, they have an equal active probability for a grid point. Thus, $\delta(q)$ can be computed,

$$\delta(q) = 1 - (1 - \phi)^{1/k}, \text{ where } k = |S(q)| \geq 1.$$  \hfill (11)

Let $U(i)$ denote the set of all the grid points within the detection range of node $i$. Then, the active probability of $i$ is

$$\delta_i = \max \{ \delta(q), \forall q \in U(i) \}. \hfill (12)$$

The max operation introduces another key issue that the active probability is not minimized. Consequently, the power consumption of each node is not optimized. To solve this issue, the algorithm proceeds with a refinement procedure. The procedure is iterative, being performed until the active probability can no longer be reduced. Then, it terminates. In essence, each sensor node computes a new feasible active probability based on the knowledge of the active probabilities of its stakeholders. If this new probability is smaller than the previous one, the sensor node will update its active probability to this new one.

After the initial active probability is calculated, the sensor node starts timer $tm_2$. The length of this timer should be sufficient for sensor nodes to exchange their computed probabilities. After timer $tm_2$ fires, the refinement procedure commences. Firstly, the sensor node computes a new feasible active probability that takes into account of the contributions from its stakeholders. The new feasible active probability of $i$ for point $q$ is given by

$$\delta_i^{(k+1)}(q) = \max \left\{ 1 - (1 - \phi) / |\theta, 0|, \prod_{j \in S(q) - \{i\}} (1 - \delta_j^{(k)}) \right\}, \hfill (13)$$

where $\theta = \prod_{j \in S(q) - \{i\}} (1 - \delta_j^{(k)})$. In (13), $(k)$ denotes the number of generation of this computed active probability. The sensor node takes the maximum out of all grid points,

$$\delta_i^{(k+1)} = \max \left\{ \delta_i^{(k+1)}(q), \forall q \in U(i) \right\}. \hfill (14)$$

When it derives a smaller active probability, the sensor node tries to update its active probability to this new one. However, there is a critical issue. The sensor node cannot immediately execute this update, since the computation of the new probability is dependent on its stakeholders. If it updated it immediately while other sensor nodes were also updating, an error would be introduced because each sensor node assumed that its stakeholders were not updating.

To address this issue, we let a sensor node that needs to update broadcast an update message before it can really execute such an update. This update message serves for two purposes. First, it is a prohibitive signal stopping its stakeholders from simultaneously updating their probabilities. Second, it conveys the new probability of the sensor node to the stakeholders. By this means, the stakeholders can use this new one to re-compute their new feasible probabilities.

Sensor nodes that want to update compete for the broadcast transmission. We do not explicitly specify the communication protocol for our algorithm. The simplest one is to use CSMA/CA as the MAC without power management. However, the algorithm does allow power management in MAC. The mere requirement for the protocol is that it implements a competition based media access. Many state-of-the-art protocols can be used [15, 19]. For example, we can exploit SMAC [19] with slight modification as the communication
protocol. SMAC is a widely used MAC in sensor networks, which provides flexible power management on the radio transceiver. Sensor nodes wake up periodically. At the beginning of each period, a synchronization message is broadcast for local time synchronization. Next, sensor nodes compete for the channel access by exchanging RTS and CTS. The pair of nodes that successfully exchange RTS/CTS is granted channel access. To exploit SMAC, we slightly modify it. Since we need broadcast-based transmission other than point-to-point transmission, we remove the use of CTS. A node that successfully broadcasts a RTS is assigned the channel and hence is allowed to broadcast the update message. After broadcasting the update message and other messages, the radio transceiver can go to sleep mode for power conservation. Note that the power scheduling of the radio can be different from that of the sensing device.

If it fails to broadcast its update message, a sensor node will receive one from its stakeholders. On reception of such an update message, it suppresses its own broadcast attempt and updates the active probability of this corresponding stakeholder. Next, it re-computes a new feasible probability based on the updated table of active probabilities.

In practice, the new feasible probability can be close to, or much smaller than the previous one. Either case is undesirable. If the difference is great, it suggests that there will be great reduction on the active probability in one adjustment. This limits the power balance among sensor nodes. Thus, we introduce a maximum threshold $\Delta_1$ by which the active probability of a sensor node can be reduced each time. This design parameter controls the tradeoff between algorithm convergence time and energy balance. If the difference is insignificant, it is waste of energy to spend a broadcast transmission to implement this update. We use another minimum threshold $\Delta_2$. Only when the probability reduction is over the minimum threshold, a sensor node is allowed to update to a new probability. This design parameter controls the tradeoff convergence time and granularity of power balance.

As time elapses, some sensor nodes may be depleted. To be adaptive, sensor nodes periodically sends out beacon. By this means, a sensor node can be aware of the leave of a stakeholder. On receiving the leave, a sensor node removes the stakeholder from its table and re-computes a new feasible probability. Next, it broadcasts an update message to inform its stakeholders of the new probability.

The granularity of grid points, characterized by grid size $d$, is a big concern in the algorithm design. The algorithm actually provides guaranteed QoS for all grid points. However, this does not necessarily imply that the QoS at any point in the field is also satisfied. To address this problem, we adopt a similar technique as in [3]. We propose a nominal detection range $r^\dagger$ that is smaller than the real detection range. For each small grid, if a sensor node covers any grid point of this grid with the nominal detection range, the node completely covers the whole grid with its real detection range. By this means, the system can deliver the guaranteed QoS at any point in the field if it ensures that the QoS at every grid point is guaranteed when using the nominal detection range. It is not difficult to compute $r^\dagger \leq \sqrt{2}d$. As shown in Figure 2, grid point $q$ is within the nominal detection range of $i$, and then the shadowed grid which $q$ is attached to is completely covered by $i$. In contrast, since $q$ is out of the nominal detection range of $i$ (although it is within the real detection range), the shadowed grid is not completely covered by $i$.

Such a solution comes at the cost of reduced energy efficiency, since each sensor node loses some area that is actually within its detection vicinity. The grid granularity controls the tradeoff between energy efficiency and computational complexity. A finer granularity can lead to better energy efficiency but causes a higher computational complexity. In our implementation, $d$ is set to one tenth of the detection range. Under this configuration, each sensor node is expected to contain $\lfloor 100\pi \rfloor$ grid points.

B. Algorithm Analysis

We analyze the distributed algorithm, including correctness, convergence and algorithm complexity.

Theorem 2 (correctness): The Loya algorithm is able to guarantee detection QoS for any event in the field.

Proof. In Section III, we have shown that the detection QoS of any event is guaranteed if the aggregate activity of any point in the field is greater than $\phi$. The initial active probability is so sufficiently large that any point has a larger aggregate activity than $\phi$. In the refinement procedure, a sensor node computes its necessary probability for each grid point in its detection range. It takes the maximum among all the grid points as its new active probability. Parallel updates are prevented using the effective broadcast competition technique. An update is a local operation in the sense that it only involves the region within the detection coverage of the updating node and does not affect other regions. By introducing the nominal detection range, the algorithm can successfully ensure that the detection QoS of any point in the field is satisfied.

Theorem 3 (convergence): The Loya algorithm terminates in a finite number of steps.
Proof. The active probability of a sensor node must be between one and zero. Each successful update will reduce the active probability by at least \( \Delta_2 \). No operation will cause the increasing of the active probability. Thus, there is no fluctuation. Since the minimum of the active probability is greater than zero, the number of updates that a sensor node could have is at most \( 1/\Delta_2 \). Thus, the algorithm terminates in a finite number of steps.

A tiny sensor node has limited computational capability. Thus, it is important that the computation complexity is affordable for such tiny nodes. We analyze the number of steps needed to determine its active probability. Each sensor node covers \( s=\pi r^2/d^2 \) grid points. Suppose a node has \( m \) stakeholders. For each grid point, this node needs \( s \) steps to determine the set of covering nodes. In computing the first active probability, the node spends constant time to compute the probability for a point. Then, it takes \( s \) steps to compute its active probability. Thus, it needs \( ms \) steps to compute the first probability. For instance, when \( d=r/10 \) and \( m=20 \), it takes less than 10 thousand steps. Later, in the refinement procedure, a sensor node performs similar operations. However, we emphasize that only those sensor nodes that feasibly further reduce probability need to perform such operations.

A tiny sensor node has only small memory. For example, a typical Mica2 sensor [21] has 4K Bytes RAM. Memory usage in the algorithm needs to be carefully investigated. The memory usage is mainly for storing the active probabilities computed for the grid points, which are \( s \) bytes. In addition, the sensor node needs \( 4m \) bytes to store the information of the stakeholders. For instance, when \( d=r/10 \) and \( m=20 \), it takes less than 1K bytes. By implementing the algorithm using TinyOS codes on a Mica2 node, we find that such computation and space cost are affordable.

It is of importance to study the communication complexity as it reflects energy overhead introduced by the algorithm. We analyze the number of transmitted messages. Both the neighbor discovery and the first probability exchange need each node to broadcast a message. Thus, each sensor node needs two broadcast transmissions. Later, a sensor node can have at most \( m \) updates and therefore it can broadcast for at most \( m+2 \) times. As a result, a node can have at most \( 2+m \) broadcast transmissions. In the implementation, we find that the \( \Delta_2 \) of 1/10 can provide a good tradeoff between convergence and communication cost.

V. SIMULATION RESULTS

We have conducted extensive simulation experiments to validate the design and to evaluate the performance of the distributed algorithm. Simulations have been performed using a discrete event driven simulator developed with emphasis on event detection and energy efficiency investigation.

We focus on the performance of lifetime extension which directly reflects energy efficiency. We define \( \alpha \)-lifetime as the amount of time until the time instant when only \( \alpha \% \) of the field can deliver the guaranteed QoS. We compare our approach (i.e., Loya) with other alternative schemes.

- **Naive.** In this scheme, every sensor node has the identical active probability of \( \phi_0 \). Note that this scheme is not adaptive to the deployment density of sensor nodes.
- **Sponsor.** This scheme extends the sponsor algorithm [2] where a set of sensor nodes are selected to provide full sensing coverage and other sensor nodes sleep. The set of nodes are assigned active probability \( \phi_0 \). The sensor network is rescheduled from time to time in order to balanced power consumption. Note that this scheme assumes time synchronization.
- **Bound.** This is a virtual scheme that provides the upper bound of the 100-lifetime. This cannot really be implemented in reality.

The tight bound of the system lifetime is very difficult to be derived. Thus, we instead try to obtain an optimistic upper bound. The major power consumption of a sensor node is attributed to three main units: processor, sensing device and radio transceiver. Ideally, each unit has separate power control [22]. Let \( \rho_s \), \( \rho_p \), and \( \rho_r \) denote the power consumption rates of the sensing device, the processor, and the transceiver, respectively. As discussed, the duty cycle of the radio transceiver is determined by the communication protocol. Here, we assume it is given. A sensor node may have multiple sensing devices. However, we assume that a sensor node has only a single sensing device for analysis simplification. Our approach can be extended, with slight modification, to support multiple sensing devices on a sensor node.

A point is covered, on average, by \( v=\pi r^2/h^2 \) sensor nodes. Ideally, these sensor nodes share the same active probability, which is \( 1-(1-\phi_0)^v \). Thus, the actual power consumption rate of the sensing device is \( 1-(1-\phi_0)^v \rho_s \). The upper bound of the 100-lifetime can be computed accordingly,

\[
\epsilon \left[ (\rho_s + \rho_r) \phi_0 + (\rho_s + \rho_p) (1 - \sqrt{1 - \phi_0}) \right],
\]

where \( \epsilon \) is the initial energy of the sensor node.

We adopted the following configuration for simulation experiments. The field is a 3000m\( \times \)300m square. There are 4000 sensor nodes, if not stated elsewhere. The detection range is 10m. Power consumption rates \( \rho_s \), \( \rho_p \), and \( \rho_r \) are set to 19mW, 20mW and 24mW, respectively. The initial energy is set to 10J. We use SMAC as the underlying communication protocol, which has a duty cycle of 0.1. The event life is set to five times of \( t_0 \). Thresholds \( \Delta_1 \) and \( \Delta_2 \) are 0.1 and 0.01, respectively. According to the QoS requirement, we derive that the minimum aggregate activity is 0.684.
Figure 3 compares different schemes in terms of 100-lifetime when the number of sensor nodes increases. We can see that our approach outperforms the Sponsor scheme and the Naive scheme. Furthermore, as the deployment density increases, the performance gain becomes more significant. This suggests that our approach scales well with the increasing number of deployed sensor nodes. In contrast, the Naive scheme has a fixed lifetime and fails to scale with deployment density. Although the Sponsor scheme is scalable to some extent, it is much worse than our approach. It is also important to mention that in order to balance energy consumption of sensor nodes in the Sponsor scheme we have to reschedule the whole network. This relies on time synchronization. In addition, such repeated rescheduling incurs high cost of wireless communication. It is also difficult in practice to determine how often the network should be rescheduled. When the number of sensor nodes is small (i.e., 2000), the three schemes almost have the same 100-lifetime. This results from the fact that at this density the sensor nodes can merely cover the field. Many spots are covered by only one sensor node. When such a critical sensor node is depleted, the 100-lifetime is determined.

We also investigate the performance of the schemes in terms of different lifetimes. Figure 4 shows the comparison. We can see that compared to Sponsor, Loya has a much longer 100-lifetime and a similar 70-lifetime. Sponsor however has a longer 50-lifetime. This exactly proves that Sponsor cannot well balance energy consumption and hence fails to obtain better performance in terms of the critical performance metric, i.e., 100-lifetime.

B. Effects of Thresholds

We explore the effects of the two important thresholds on the two aspects of the algorithm – convergence and communication overhead. Table 1 shows the convergence time of the algorithm under different settings of the two thresholds. The convergence takes much longer time when \( \Delta_1 \) is 0.05 than it is 0.1. When \( \Delta_1 \) is smaller, it takes a sensor node more iterations to reach its final probability. When \( \Delta_2 \) is 0.02, the algorithm takes a shorter time to converge than when \( \Delta_2 \) is 0.01. If \( \Delta_2 \) is larger, more update attempts are suppressed, leading to a shorter convergence time. In Table 2, we show the average number of transmitted packets. The number of packets per node is less than three for all configurations. This demonstrates that the communication cost is modest. Among the number, one packet is used for neighbor discovery and one is for exchanging the first active probability. Therefore, it is scalable with the node density. The number of packets per node in the configuration of \( \Delta_1=0.05 \) is considerably larger than in the configuration of \( \Delta_1=0.1 \), because a greater \( \Delta_1 \) makes the algorithm converge more rapidly. Moreover, the configuration of \( \Delta_2=0.01 \) generates slightly more packets than the configuration of \( \Delta_2=0.02 \), as a smaller \( \Delta_2 \) encourages more updates on active probability.

<table>
<thead>
<tr>
<th>( \Delta_1 )</th>
<th>12.78</th>
<th>11.96</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>8.77</td>
<td>8.63</td>
</tr>
</tbody>
</table>

<p>|</p>
<table>
<thead>
<tr>
<th>( \Delta_2 )</th>
<th>2.98</th>
<th>2.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>2.58</td>
<td>2.54</td>
</tr>
</tbody>
</table>

VI. DISCUSSIONS

When a sensing device switches from power-save mode to active mode, there is a warm-up time during which it consumes power but it is not functional. This warm-up time essentially introduces a switching overhead of power consumption. However, the recent study [14] shows that the warm-up time for most sensing devices is only several microseconds. This means that the switching overhead of power consumption is modest.

We divide the time into equal time slots. In every time slot, a sensor node is randomly in active mode or in sleep mode according to the active probability. Consecutive time slots are
independent. There is a concern that a sensor node may switch between sleep and active mode very frequently in a short time. The consequence is that each switch incurs an overhead of energy. To address this issue, we can combine several time slots into a bigger slot. Then, each sensor node uses this new slot to arrange its working schedule. By this means, the switching overhead is reduced. However, this will slightly change the granularity of the time-invariant property of the aggregate activity.

This paper has focused on the scenario where events have a typical fixed event lifetime. However, it is possible that events may have different lifetimes in some applications. To address this issue, we need to have the probability distribution knowledge of event lifetimes. Then, we can compute the expectations of detection probability and detection latency, based on which QoS can be also defined. Our approach can also be applied.

VII. CONCLUSIONS

As sensor networks are widely used in various event detection applications, it has become an important yet challenging problem to provision detection QoS. This paper has proposed a novel probabilistic approach to this problem, which support dual-objective QoS considering two important detection metrics including detection latency and detection probability for distributed event detection. A unified framework based on the concept of aggregate activity is proposed. The fully distributed algorithm determines the critical configuration parameter of every sensor node, i.e., the active probability. The algorithm minimizes the active probability and hence reduces the power consumption. In addition, it ensures that the aggregate activity of any point in the field is larger than the application-specific minimum. As a result, the dual-objective field-wide QoS is successfully delivered. Performance results demonstrate that our approach outperforms other alternative schemes in energy efficiency.

ACKNOWLEDGMENT

This research was supported in part by the National Basic Research Program of China (973 Program) under Grant No. 2006CB303000, by the Key Project of China NSFC Grant 60533110, by HKUST Nansha Research Fund, by Hong Kong RGC Grants HKUST6183/05E and HKBU 1/05C and by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant EP/C547586/1 (Biosensornt Project). The authors would also like to thank valuable discussions with Zheng Zhang and Yu Chen in MSRA.

REFERENCES