2012 IEEE 18th International Conference on Parallel and Distributed Systems

EODS: An Energy-efficient Online Decision Scheme in Delay-sensitive Sensor Networks for Rare-event Detection

Lijie Xu*†, Jiannong Cao†, Xuefeng Liu†, Haipeng Dai* and Guihai Chen*‡
*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
†Internet and Mobile Computing Lab, The Hong Kong Polytechnic University, Hong Kong
‡Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, Shanghai, China
Email: ljxu83@gmail.com, {csjcao, csxfliu}@comp.polyu.edu.hk, dhpphd2003@gmail.com, gchen@nju.edu.cn

***Abstract*—In many applications of WSNs, the events occur infrequently but once they occur, the corresponding information needs to be sent to the sink node in a short period of time for the necessary reactions. Detection of rare events in a fast and energy-efficient manner is an important issue in WSNs. In this paper, based on the optimal solution for the Best-choice Problem with Bounded Random Observation Number, we propose an Energy-efficient Online Decision Scheme (EODS) to handle this problem. Combining with the design of nodes' duty cycle, the EODS avoids the redundant transmissions and achieves the tradeoff between delay and energy efficiency. Simulation results reveal that the EODS achieves a good balance between delay and energy efficiency.**

***Keywords*-duty-cycled WSNs; rare-event detection; energy-efficient; online decision**

I. Introduction

Wireless sensor networks (WSNs) have been widely used in many applications such as environmental monitoring, medical system and scientific exploration [1]. In most of these applications, sensor networks are expected to be operated for a long period of time (months or years). To extend the lifetime of wireless sensor nodes, a widely adopted approach is to put them in a duty-cycled mode. Sensor nodes in the duty-cycled mode can significantly decrease the energy waste caused by idle listening, which consumes 50%-100% of the energy required for receiving and is the major source of energy waste [2].

Many events, although occurring infrequently, once occur, need to be reported to the sink node in a short period of time since corresponding actions usually need to be taken promptly based on the received detection information. For example, a prompt action to extinguish the fire is needed once the detection information of forest fires received. Therefore, how to report the sensed information to the sink node as soon as possible once an urgent event occurs becomes an important issue for such applications. Detecting rare but urgent events using sensor networks working in a low-duty-cycle mode is a challenging task. Considering the sleep latency for the multihop transmissions in duty cycled WSNs, it is preferable for senor nodes which detect the events to transmit the sensed information to the sink within single hop. Moreover, [3] shows that single hop routing is normally more energy efficient in total energy consumption compared with the multi-hop routing under realistic circumstances due to the drain efficiencies of the power amplifiers of current sensor node devices are actually always less than 100%. For such applications, therefore, transmitting to the sink by single hop is preferred for the nodes that can reach the sink within their maximal transmitting powers in terms of both delay and total energy consumption. However, in a randomly deployed duty-cycled sensor network, the occurrence of an event may be sensed by multiple nodes in sequence if it is located in the intersection region of their sensing ranges. If all the sensor nodes which detect the event transmit the packet to the sink, redundant energy consumption is generated since the same sensed event information is reported to the sink by multiple times. Especially for the peripheral region of the network which becomes the 'hot spot' due to the single-hop communication is adopted, the redundant transmissions with relatively larger power level would further deteriorate the network lifetime. In many applications, it is enough for the sensed information to be delivered to the sink by only one of the sensing nodes. In order to reduce the redundant energy consumption for such kind of applications, our objective is to select one sensing node with good tradeoff between node-to-sink delay and energy efficiency to be responsible for delivering the event information to the sink node. In this paper, we come up with an Energy-efficient On-line Decision Scheme (EODS), which is combined with the design of nodes' duty cycle, to achieve the objective mentioned above for the event-driven sensor networks in which the events are rare but urgent.

The rest of the paper is organized as follows: Section II illustrates the network model and states the problem. Detailed description and analysis of our scheme is presented in Section III. Followed by the simulation results in Section IV. The paper concludes our findings in Section V.

II. Preliminaries

A. Network Model and Assumptions

We assume sensor nodes are all randomly deployed in the network and operated in a duty-cycled Low Power Listening (LPL) mode, where each node wakes up once every a fixed

1521-9097/12 $26.00 © 2012 IEEE
DOI 10.1109/ICPADS.2012.50

LPL checking interval. Further, we assume all nodes have the identical LPL checking interval L and they can set their wakeup schedules independently, also, they do not share their own schedules with their neighbors, which means no clock synchronization overhead is required in our model.

In our model, when being ready to report the event to the sink, each node should properly increase its transmitting power to surely reach the sink by one hop according to the distance to the sink. Here, we associate each node with a Performance Index (PI) value to indicate how good it is in terms of the energy efficiency. The performance index value $PI(i)$ for any node i is defined as follows:

$$PI(i) = E_i - e_{is} \tag{1}$$

where E_i denotes the current residual energy of node i, and e_{is} denotes the least required energy consumed by node i in transmitting one packet to the sink by single hop. Apparently, to choose the node with higher PI value as the event-reporter would bring higher energy efficiency of the network.

Before stating the problem, we make several basic assumptions used in this work as follows:

(1) We make use of disk model as the sensing model of nodes, and the communication range is initially set twice the sensing range for each node, which can ensure that the full coverage of the sensory field implies connectivity [7].

(2) Each node is aware of the location information of itself and its local neighbors, which is reasonable since in event-driven sensor networks, nodes are desirable to report their locations along with the sensed event information to the sink node so that any prompt reactions could be taken at the given location area.

(3) We assume each sensor node can communicate with the sink directly within its maximal transmitting power, in other words, each node can increase its transmitting power according to the distance to the sink so that the sink is just within its communication range, and this assumption will be relaxed in section III.

(4) An event occurs at one point in the sensory filed, and once an event occurs, it is sensed sequentially by the nearby duty-cycled sensing nodes with a random ordering in terms of relative ranks of nodes' PI values, and each of the orderings is equally likely. Also, the events in our target applications are assumed rarely happened, that is, the expected period between successive events occurred in any local area of the sensory field is relatively long.

(5) We consider the delay is dominated by the sleep latency since the transmission delay can be neglected compared with the sleep latency in terms of the order of magnitude.

B. Problem Statement

Here, we will take Fig.1 as an example to illustrate the problem. We assume node A is the first one to wake up and sense the event overall. Once an event arises inside

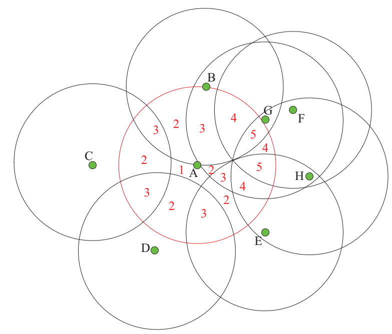


Figure 1: An Example with n=5.

the sensing range of node A, it could be sensed by N different nodes according to their wakeup sequences, in which $N = 1, 2, \ldots, 5$. As stated in Section I, it may result in redundant transmission and only one sensing node is enough as the candidate to report the event. Besides, an event can be detected as long as its lifespan intersects any of the waking periods of neighboring sensing nodes [4]. Here, we define the lifespan of any event is at least T_e, and thus the checking interval L of each node should be set at most T_e in order to make sure that all events could be detected even if in the worst case, where the event is only sensed by one sensing node (e.g. the event occurs in region 1 as Fig.1 shows). It also requires that the perceived surveillance delay, which refers to the time elapsed from the occurrence of an event to the time the event is reported to the sink [4], should be restricted within a given threshold.

Intuitively, we are inclined to choose the node with the higher PI value as the candidate in order to achieve a better load balance and prolong the network lifetime as much as possible. Meanwhile, the shorter detection delay is also desirable for delay-sensitive networks with urgent events. How to make a good tradeoff between them is a challenging problem, and thus we have the following objective.

Problem 1. *Given a delay threshold T_d, how to set an appropriate length of L and design an on-line decision scheme to select a sensing node with good tradeoff between energy efficiency and delay, namely the sensing node with higher PI value as well as shorter detection delay, as the candidate to report the event on the basis of the perceived surveillance delay of all events must be within T_d;*

Seen from the Problem 1, when choosing the sensing node, we only need to compare the PI values of neighboring nodes. Specially, if all nodes are homogeneous, the value $PI(i)$ for any node i could be approximately represented by E_i in that the e_{is} of neighboring nodes are quite similar as a result of their contiguity in locations.

III. EODS Details

The design of our Energy-efficient On-line Decision Scheme (EODS) which is proposed to address Problem 1 mainly consists of two portions: decision-maker selection and on-line decision.

A. Decision-maker Selection

A crucial issue to address Problem 1 is that once an event occurs, how to find the decision-maker, which is defined as the first sensing node that has sensed the event. To solve this issue, we propose the following distributed Decision-maker Selection Algorithm(DSA).

Each node repeats the channel detection until the channel is idle once it wakes up and has sensed a new event[1], and then broadcasts a short beacon packet by one hop, which only contains its ID and PI value. Once a node receives nothing during some short period after sending the short beacon, which means it is the first node to wake up and senses the new event, it can thus consider itself as the decision-maker and then keep listening for a period of checking interval L which is also referred to as the decision period so that it can surely hear the beacons from all the nodes that have sensed the event. The main task of the decision-maker is to make an on-line decision on receiving the beacons in sequence. Once a node is determined as the decision-maker, it should decide whether to choose itself as the candidate according to an on-line decision method which will be elaborated in next subsection, if yes, it should report the sensed information to the sink immediately by increasing its transmitting power so that the sink is just within its communication range. Likewise, once a beacon from any other node is received and the candidate has not been selected yet, the decision-maker should decide whether to accept or reject the node with the highest PI value among the nodes, that have sent the beacons so far, as the candidate according to the on-line decision method, if reject, it should at once return a short $CAND\_UNSELECTED\_REPLY$ packet attached with ID of the node with highest PI value thus far; if accept, a $CAND\_SELECTING\_REPLY$ packet, which is attached with the candidate's ID, should be broadcasted in the neighboring region and the decision-maker will report the event to the sink immediately if it is the candidate. In our method, we let the decision-maker return the $CAND\_SELECTED\_REPLY$ packets to those nodes whose beacons arrive after the candidate is selected. Therefore, for any node that senses the event, if it receives a $CAND\_UNSELECTED\_REPLY$ shortly after sending the short beacon, it will check whether it is the sensing node with the highest PI value thus far, if yes, it keeps awake until a $CAND\_SELECTING\_REPLY$ is received; if no, it goes to sleep right away and goes on working in accordance with its wakeup schedule. If it receives a $CAND\_SELECTED\_REPLY$, that means the candidate has been selected till now, the node will thus go into sleeping state immediately. And if a $CAND\_SELECTING\_REPLY$ is received, it will check whether it is the candidate, if yes, it reports the event to the sink directly; otherwise, it goes to sleep immediately. Finally, if the candidate has not yet been selected at the end of the decision period, the decision-maker will choose the node with the highest PI value overall as the candidate.

[1]As the assumption (4) mentioned, an event which is sensed at some active state of any sensing node is defined as new if and only if there is no event is sensed at the last scheduled active state of this sensing node.

Here, we assume that the beacon and reply packets are so small that energy overhead could be neglected compared with the case of reporting the larger sensed data packets to the sink directly, where the transmitting powers are much higher especially for nodes in the peripheral region of the network which is the bottleneck of the energy efficiency. Also, DSA can make sure that once an event occurs, only one sensing node is selected as the candidate to avoid the redundant energy. And apparently seen from DSA, a sensing node that has been rejected could later be recalled.

B. On-line Decision

As stated above, the decision-maker should perform an on-line decision scheme such that the candidate could be energy-efficiently determined as quickly as possible. Before describing our proposed On-line Decision Algorithm(ODA), we first demonstrate how to appropriately set the checking interval length L of each node in order to satisfy the delay constraint. As stated in section II, L should be no more than T_e and for any given delay threshold T_d, the choice of L should also make sure the perceived surveillance delay be restricted within T_d. Seen from DSA, the decision-maker can finish the candidate selection within L, which means $2L$ should be no more than T_d in order to guarantee the delay constraint even for the worst case where the event is sensed by only one node. Normally, L is expected to be as large as possible in duty-cycled networks in terms of the energy efficiency and thus is set in this paper as follows:

$$L = \min\{T_e, T_d/2\} \quad (2)$$

Suppose that node i is the decision-maker, of which sensing range R_i would be divided into M non-overlapping unit regions with each being impartible by the borders of its neighbors' sensing ranges. Here, we label these unit regions with $R_i(m)(m = 1, 2, \ldots, M)$, and for each $R_i(m)$, we let $Num(R_i(m))$ denote the number of sensing nodes whose sensing ranges cover region $R_i(m)$. These information can be derived by node i due to the awareness of its own and neighbors' locations. Once any node i is determined as the decision-maker, it means the event must occur within R_i. We assume the event occurrence each time is uniformly distributed in the field, node i can thus figure out the random variable N_i, i.e. the number of sensing nodes that can sense the event, meets the following distribution of probability:

$$p_i(j) = P\{N_i = j\} = \sum_{Num(R_i(m))=j} \frac{S(R_i(m))}{S(R_i)} \quad j = 1, 2, \ldots, n \tag{3}$$

where $S(R_i(m))$ and $S(R_i)$ denote the area of region $R_i(m)$ and region R_i respectively, and

$$n = \max_{m=1,\ldots,M} \{Num(R_i(m))\} \tag{4}$$

Seen from Fig.1 in which $M = 16$ and $n = 5$, once being determined as the decision-maker, node A can be aware of the distribution of probability $p_A(j)(j = 1, \ldots, 5)$, which can be calculated by equation (3).

Actually, Problem 1 equals to the following On-line Decision Problem (ODP).

Problem 2 (ODP). *Based on the Decision-maker Selection Algorithm(DSA), how to design an on-line decision scheme for the decision-maker to select a sensing node with good tradeoff between energy efficiency and delay, as the candidate to be the event-reporter.*

Before addressing ODP, we first consider the Simplified On-line Decision Problem, which is abbreviated to SODP.

Problem 3 (SODP). *Based on the Decision-maker Selection Algorithm(DSA), how to design an on-line decision scheme for the decision-maker to select the sensing node with the highest PI value overall as the candidate to be the event-reporter in the presence of the following additional assumptions: (1) The short beacon packet in decision-maker selection process contains no ID information while only the PI value; (2) Once receiving the beacon from any sensing node j, the decision-maker must decide to either accept or reject node j as the candidate immediately, and a sensing node once rejected cannot later be recalled.*

In SODP, we use a sequence of variables, $X_i(1)$, $X_i(2)$, $\ldots$, $X_i(N_i)$, to denote the observations, i.e. relative ranks of PI values that are sequentially received by the decision-maker i, where $X_i(j)$ is the rank of the jth received PI value among the first j received PI values at the decision-maker i by the ordering from the largest to the lowest, rank 1 being the best and the 1st value received by the decision-maker i is the PI value of itself, thus $X_i(1) \equiv 1$. Further, a sequence of real-valued reward functions, $y_i(1, N_i)$, $y_i(2, N_i)$, $\ldots$, $y_i(N_i, N_i)$, is defined as follows:

$$y_i(\mathrm{j},\mathrm{N}_i) = \begin{cases} P\{\text{the largest PI overall appears among} \\ \quad \text{the first j received ones}\} & \text{if } \mathrm{X}_i(j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Where $1 \leq j \leq N_i$ and $0 \leq y_i(j, N_i) \leq 1$. The decision-maker i observes the sequence $X_i(1)$, $X_i(2)$, $\ldots$, $X_i(N_i)$ during its decision period based on the sequentially received PI values from itself and its neighbors. For each stage $j = 1, 2, \ldots, N_i$, after observing $X_i(1)$, $X_i(2)$, $\ldots$, $X_i(j)$, the decision-maker i may stop and receive the reward $y_i(j, N_i)$ which denotes the probability that the jth received PI value is the largest over all N_i values, or may continue to wait for the arrival of next PI value and observe $X_i(j + 1)$. The objective for the decision-maker in SODP is thus to choose a stage to stop as early as possible to maximize the expected reward and determine the node, whose PI value is received at the stopping stage, as the candidate. If the decision-maker never stop at the jth stage where j is from 1 to N_i, it has to choose itself as the candidate to be responsible for reporting at the end of its decision period.

As the assumption (4) mentioned in section II, we can find that SODP is actually the *Best-choice Problem with Bounded Random Observation Number* [5], which is a well-known problem in *Optimal Stopping Theory*. The probability distribution of the number of observations N_i in SODP is known as $p_i(j)$ $(j = 1, \ldots, n)$, where n is the bound of N_i.

Definition 1 ($s(r)$ rule). *For any integer $r \in [1, n]$, the decision-maker i rejects the first $r-1$ PI values and accepts the next first relatively largest PI value, namely stops at the hth stage in which $h = \min\{r \leq h \leq N_i : X_i(h) = 1\}$. If $X_i(h) > 1$ for all $h \in [r, N_i]$, accept the PI value of the decision-maker i at the end of its decision period.*

As shown in [5], $s(r)$ rule is the optimal stopping policy for the Best-choice Problem with Bounded Random Observation Number and thus can be used to solve SODP as an optimal solution. Here, we employ $\phi(r, N_i)$ to denote the probability that the chosen PI value by using $s(r)$ rule is the largest overall, and under assumption (4), it is obvious that if $r = 1$,

$$\phi(r, N_i) = \sum_{m=1}^{n} \frac{p_i(m)}{m} \tag{6}$$

if $r \geq 2$,

$$\begin{aligned} &\phi(r, N_i) \\ &= E(y_i(\min\{l \geq r : X_i(l) = 1\}, N_i)) \\ &= \sum_{m=r}^{n} p_i(m) \sum_{k=r}^{m} P\{\min\{l \geq r : X_i(l) = 1\} = k\} \\ &\quad \cdot (y_i(k, m)|X_i(k) = 1) \\ &= (r-1) \sum_{m=r}^{n} \frac{p_i(m)}{m} \sum_{k=r-1}^{m-1} \frac{1}{k} \end{aligned} \tag{7}$$

Therefore, we can see that the optimal rule is to reject $r_{opt} - 1$ PI values and then accept the next first relatively largest PI value, in which

$$r_{opt} = \arg\max_{1 \leq r \leq n} \{\phi(r, N_i)\} \tag{8}$$

However, $s(r)$ rule may not be an efficient method when being used in ODP, where node ID information are included in the short beacon packets which may result in the Information Discordance Problem. Here, we will take Fig.1 as an example to illustrate what the problem is.

In SODP, after receiving a new beacon from one of its neighbors, the decision-maker A is unaware of which node the received beacon is from since no node ID information is included in the short beacons, so the initially derived probability distribution $p_A(j)(j = 1, \ldots, 5)$ is always available, which we call the Information Accordance of the probability distributions at different stages. In ODP, however, Information Discordance Problem may come out. As shown in Fig.1, once receiving the beacon from node C, the decision-maker A will get the knowledge that the event must occur within the intersection region of node A and node C's sensing ranges due to node ID is included in the received beacon, and the probability distribution as well as the bound n of N_i is thus changed at this stage. For instance, given that the decision-maker A figures out the optimal r is 4 by using $s(r)$ rule at the beginning of the decision period, which means that it should reject the first 3 PI values and accept the next relatively largest one. But when receiving the value from node C, A finds that the bound of N_i is now refined to be 3 instead of the original value 5. Obviously, the predetermined optimal policy in which $r = 4$ is not valid any more. Even the bound of N_i is unchanged at some stage, the probability distribution of N_i has also been changed since the possible range of the event occurrence has been restricted into a more refined intersection region that determined by the sequentially received ID information until current stage, that means the probability distribution information of N_i acquired at current stage is more precise compared with the previous ones, which we call the Information Discordance Problem. Therefore, $s(r)$ rule may be inefficient when being directly used in ODP due to the Information Discordance Problem. Next, we will introduce an efficient $s(r)$ rule-based on-line decision scheme which is used in ODP.

First, we replace the equation (3) as follows:

$$\begin{aligned} p_i(j,k) &= P\{N_i|k = j\} \\ &= \sum_{Num(R_i(m))=j} \frac{S(R_i(m)\cap(\bigcap_{s\in I(k)} R_s))}{S(\bigcap_{s\in I(k)} R_s)} \end{aligned} \tag{9}$$

where $j = k, \ldots, n_k$; $N_i|k$ and $p_i(j,k)$ respectively denote the number of sensing nodes that sense the event and its probability distribution derived at the kth stage by the decision-maker i; $I(k)$ denotes the set of $\{ID_1, ID_2, \ldots, ID_k\}$ in which ID_k means the ID of the sensing node which the received beacon at the kth stage is from and obviously $ID_1 = i$; n_k denotes the bound of N_i derived at the kth stage and

$$\begin{aligned} n_k = \max_{R_i(m)\subseteq \bigcap_{s\in I(k)} R_s \; for \; m=1,\ldots,M} \{Num(R_i(m))\} \\ (k \leq n_k \leq n_1 \; and \; n_k \geq n_{k+1}) \end{aligned} \tag{10}$$

In ODP, we therefore make use of the aforementioned conditional probability distribution $p_i(j,k)$ at each stage which is more precise than that used in SODP to make our decision. Similar to $s(r)$ rule, we define the following $s(r|k)$ rule at the kth stage here.

Definition 2 ($s(r|k)$ rule)**.** *When receiving the PI value at the kth stage, the decision-maker i stops and accepts the largest PI value from the 1st stage to the kth stage for r=1, and for any integer $r \in [2, n_k - k + 1]$, the decision-maker i rejects the first $r-1$ PI values from the kth stage on and then accepts the next first relatively largest PI value from the 1st stage on, namely stops at the hth stage in which $h = \min\{k + r - 1 \leqslant h \leqslant N_i|k : X_i(h) = 1\}$.*

Here, we employ $\phi(r|k, N_i|k)$ to denote the probability that the chosen PI value by using $s(r|k)$ rule at the kth stage is the largest overall, and based on the assumption (4), $\phi(r|k, N_i|k)$ can similarly be figured out as follows: if $r = 1$,

$$\phi(r|k, N_i|k) = \sum_{m=k}^{n_k} \frac{p_i(m,k)}{m} \cdot k \tag{11}$$

if $2 \leq r \leq n_k - k + 1$,

$$\begin{aligned} &\phi(r|k, N_i|k) \\ &= E(y_i(\min\{l \geqslant k + r - 1 : X_i(l) = 1\}, N_i|k)) \\ &= (k + r - 2) \sum_{m=k+r-1}^{n_k} \frac{p_i(m,k)}{m} \sum_{t=k+r-2}^{m-1} \frac{1}{t} \\ &= \sum_{m=k+r-1}^{n_k} \frac{p_i(m,k)}{m} \sum_{t=k+r-2}^{m-1} \frac{k+r-2}{t} \end{aligned} \tag{12}$$

With the existence of Information Discordance Problem, we prefer to make decision at each stage based on the up-to-date probability distribution information. Also, ODP can recall the past rejected node as the candidate, our policy for ODP is therefore to greedily choose the stage s_{opt} to stop, where

$$s_{opt} = \min\{k \geq 1 : \arg \max_{1\leq r\leq n_k-k+1} \{\phi(r|k, N_i|k)\} = 1\} \tag{13}$$

and the following On-line Decision Algorithm(ODA) shows the decision process.

At each stage k, the decision-maker i judges that whether $\phi(1|k, N_i|k)$ is larger than any other $\phi(r|k, N_i|k)$ where r is from 2 to $n_j - k + 1$ based on the currently available probability distribution $p_i(j,k)(j = k, \ldots, n_k)$. If no, it sends a $CAND\_UNSELECTED\_REPLY$ to its neighbors and continues to wait for the next stage; If yes, that means to choose the best node from nodes $\{ID_1, \ldots, ID_k\}$ as the candidate is the optimal policy at current stage and a $CAND\_SELECTING\_REPLY$ packet with the candidate ID should be sent, specially, if the candidate is the decision-maker itself, it should report the sensed event to the sink directly and immediately. Once the candidate

Algorithm 1 On-line Decision Algorithm

Input: The decision-maker N(i)
Procedure: ODA(i)

candidateSelected=FALSE; k=1;
ID_1=i; PI_1=PI(i);
while N(i).currentTime<N(i).nextWakeupTime **do**
  if k>1 **then**
    N(i) keeps listening for a short time;
    if no packet is received **then**
      continue;
    end if
    if a BEACON packet b_pkt is received **then**
      ID_k=b_pkt.getID; PI_k=b_pkt.getPI;
      if candidateSelected==TRUE **then**
        send a CAND_SELECTED_REPLY packet;
        k=k+1; continue;
      end if
    end if
  end if
  compute $\phi(r|k, N_i|k)$(r=1, ..., n_k-k+1);
  opt=arg $\max\limits_{1 \leqslant j \leqslant k} \{PI_j\}$;
  if arg $\max\limits_{1 \leqslant r \leqslant n_k - k+1} \{\phi(r|k, N_i|k)\} == 1$ **then**
    select node N(ID_{opt}) as the candidate;
    if opt==1 **then**
      send eventMSG(i) to sink directly;
    end if
    send a CAND_SELECTING_REPLY(ID_{opt});
    candidateSelected=TRUE;
  else
    send a CAND_UNSELECTED_REPLY(ID_{opt});
  end if
  k=k+1;
end while
if candidateSelected==FALSE **then**
  select node N(ID_{opt}) as the candidate;
  if opt==1 **then**
    send eventMSG(i) to sink directly;
  end if
  send a CAND_SELECTING_REPLY(ID_{opt});
end if

is selected at some stage, the decision-maker i will send $CAND\_SELECTED\_REPLY$ packets to those nodes which send beacons after that stage so that the redundant transmission is avoided. Besides, when the next scheduled wakeup time is arriving, the decision-maker i should check whether the candidate has been selected during the decision period before executing the Decision-maker Selection Algorithm, if no, it will choose the best node overall as the candidate by sending a $CAND\_SELECTING\_REPLY$ packet with the candidate ID. The detailed process is shown in Algorithm 1.

Essentially, our On-line Decision Algorithm is a tradeoff between two intuitive strategies, that is, the decision-maker always chooses itself as the candidate at the beginning of the decision period which is called Delay-first Strategy, and it always chooses the best node overall as the candidate at the end of the decision period, which is called Energy-first Strategy. Our On-line Decision Algorithm which is a greedy-like method is actually to balance the energy efficiency with the delay.

Theorem 1. *Applying the strategy with On-line Decision Algorithm, the stopping stage s_{opt} will not exceed $\frac{1}{2}n_1$.*

Proof: According to equation (11) and (12), we can obviously find that the sufficient condition of $\arg \max\limits_{1 \leq r \leq n_k - k+1} \{\phi(r|k, N_i|k)\} = 1$ is that for any $2 \leq r \leq n_k - k + 1$,

$$k \geq n_k - k \geq n_k - k - r + 2 \geq \sum_{t=k+r-2}^{n_k-1} \frac{k+r-2}{t} \quad (14)$$

namely $k \geq \frac{1}{2}n_k$.

As we know, $n_1 \geq n_k (k > 1)$, thus $\frac{1}{2}n_1 \geq \frac{1}{2}n_k$, which means at stage $\frac{1}{2}n_1$, the decision process must have stopped. In other words, the stopping stage s_{opt} must not exceed $\frac{1}{2}n_1$. ■

Theorem 2. *If $p_i(j, 1) \leq p_i(j+1, 1)$ for $j = 1, \ldots, n_1 - 1$ and $p_i(1, 1) > \sum\limits_{m=3}^{n_1} \frac{p_i(m,1)}{m} \sum\limits_{k=2}^{m-1} \frac{1}{k}$, the strategy with On-line Decision Algorithm will turn to be Delay-first Strategy.*

Proof: First, we proof the sufficient condition that $\phi(r|1, N_i|1)$ is a unimodal function of r is $p_i(j, 1) \leq p_i(j+1, 1)$ for $j = 1, \ldots, n_1 - 1$. Suppose $\phi(r|1, N_i|1)$ is not unimodal, that means there exists a r so that $f(r) = \phi(r|1, N_i|1) - \phi((r+1)|1, N_i|1) \geq 0$ and $f(r+1) = \phi((r+1)|1, N_i|1) - \phi((r+2)|1, N_i|1) < 0$. In fact, $\phi(r|1, N_i|1)$ equals to $\phi(r, N_i)$ as shown in equation (6) and (7). Therefore,

$$\begin{aligned} &\phi((r+1)|1, N_i|1) - \phi((r+2)|1, N_i|1) \\ &= \sum_{m=r+1}^{n_1} \frac{p_i(m,1)}{m} \Big(1 - \sum_{k=r+1}^{m-1} \frac{1}{k}\Big) \end{aligned} \quad (15)$$

and $\phi((r+1)|1, N_i|1) < \phi((r+2)|1, N_i|1)$ implies that

$$1 - \sum_{k=r+1}^{n_1} \frac{1}{k} < 0 \quad (16)$$

since $p_i(j, 1) \leq p_i(j+1, 1)$ for $j = 1, \ldots, n_1 - 1$, thus,

$$\sum_{m=r+1}^{n_1} \frac{1}{m} \leq \sum_{m=r+1}^{n_1} \frac{p_i(m,1)}{m \cdot p_i(r,1)} \quad (17)$$

from equation (16) and equation (17) we can derive that

$$p_i(r,1) - \sum_{m=r+1}^{n_1} \frac{p_i(m,1)}{m} < 0 \quad (18)$$

Actually, we note that

$$f(r)-f(r+1) = (p_i(r,1)-\sum_{m=r+1}^{n_1}\frac{p_i(m,1)}{m})\cdot\frac{1}{r} > 0 \quad (19)$$

which contradicts with the equation (18), and thus if $p_i(j,1) \le p_i(j+1,1)$ for $j = 1,\ldots,n_1-1$, $\phi(r|1,N_i|1)$ is a unimodal function of r.

Obviously, if $\phi(r|1,N_i|1)$ is a unimodal function of r and meanwhile $f(1) > 0$, $\phi(1|1,N_i|1)$ must be the largest overall, and $f(1) > 0$ implies

$$p_i(1,1) > \sum_{m=3}^{n_1}\frac{p_i(m,1)}{m}\sum_{k=2}^{m-1}\frac{1}{k} \quad (20)$$

thus, the proof is completed. ■

C. Sink Augmentation Scheme

In practice, assumption (3) may not always hold especially for the large-scale network and many nodes that are far away from the sink node may not be able to reach the sink by one hop. Thus, we employ a sink augmentation scheme which is similar to that proposed in [6] to relax the assumption (3). Essentially, our problem is to select a minimum subset of nodes as the sink nodes such that any node can communicate with at least one of the sinks directly within its maximal transmitting power, which is a typically Set Cover Problem and can be solved heuristically like shown in [6]. Compared with the scheme proposed in [6] which adopts a completely multi-hop mode, our sink augmentation scheme combined with EODS design requires much less number of sinks especially for tight delay requirement.

Clearly, the number of augmented sinks increases as the maximal transmitting power decreases. In practice, we can therefore properly decrease the allowable maximal transmitting power to balance the energy cost and the sink augmentation cost.

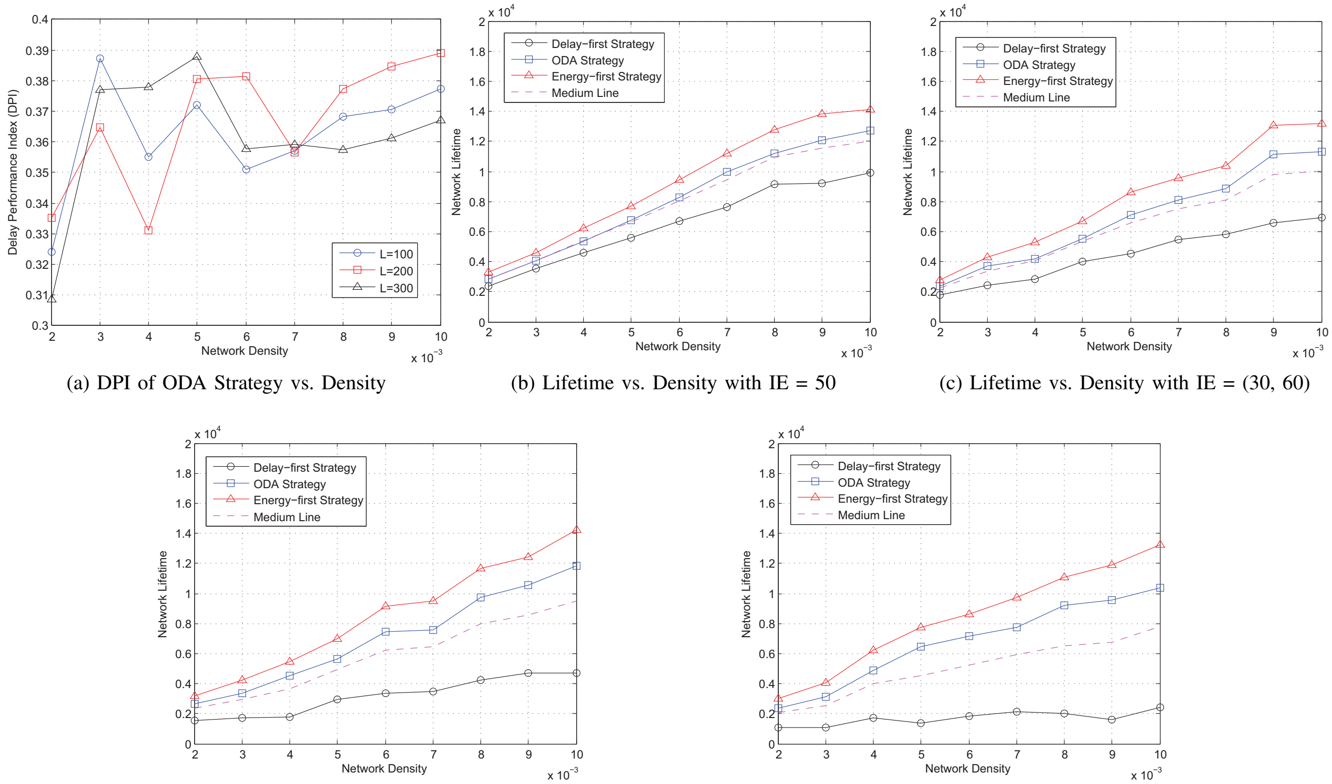
IV. Simulation

In this section, we evaluate the performance of EODS via simulation experiments. For simplicity, we consider that sensors are distributed uniformly in a 200m*200m sensory field where the sensing range and the communication range of each node are 25m and 50m respectively. In fact, EODS can still work for the nonuniform distributed case. Unless indicated explicitly, the LPL checking interval is set as 100 time units and each node independently and randomly chooses a time to wakeup once every LPL checking interval. Also, it is assumed that 1 unit energy is consumed when delivering a data packet to the sink directly for all nodes. All the results are derived from the average of results of 50 experiments.

In our simulation, we figure out the probability distribution $p_i(j,k)$ of random variable $N_i|k$ for any node i by sampling, that is, uniformly and randomly choosing 1000 sampling points within the sensing range of node i and the area of any region can be approximately represented by the number of sampling points located in that region.

First, we compare the ODA strategy which is used in our EODS with two aforementioned strategies: delay-first strategy and energy-first strategy in terms of delay and energy efficiency. As we know, the delay-first strategy and the energy-first strategy can achieve the shortest and the longest perceived surveillance delay respectively, and herein we define the Delay Performance Index (DPI) of one strategy as the ratio of the delay difference between this strategy and the delay-first strategy to the maximal delay gap, which refers to the delay difference L between delay-first strategy and energy-first strategy. Likewise, the Energy Performance Index (EPI) of one strategy is defined as the ratio of the lifetime difference between this strategy and the delay-first strategy to the maximal lifetime gap which refers to the lifetime difference between delay-first strategy and energy-first strategy. Obviously, we are expected to find a strategy with low DPI and high EPI as a tradeoff between delay-first and energy-first strategies. Fig.2(a) shows the relationship between the network density and the DPI of ODA strategy. As shown in Fig.2(a), the DPI of ODA strategy is only about 30%-40% no matter what the network density is and how the checking interval L changes. We find that the DPI of ODA strategy is relatively stable as the network density increases, this is because the increase of the network density which prolongs the stage to stop will also incur the reduction of the expected time between successive stages.

Here, we define the network lifetime as the number of events that have occurred in the sensory field when the first node dies for simplicity consideration and actually this definition can represent the energy efficiency to a certain extent. Next, we compare the network lifetime when applying ODA strategy with those when applying other two strategies for the cases where the initial energies of nodes are homogeneous and heterogeneous respectively. In our setting, we set the Initial Energy(IE) as 50 units for all nodes at the homogeneous case, and for the heterogeneous case, we make each node set its Initial Energy (IE) as a random value in the range of (30, 60), (20, 70) and (10, 80) respectively. Seen from Fig.2(b) to Fig.2(e), the EPI of ODA strategy is always over 50% no matter in the homogeneous or the heterogeneous case, and it increases as the difference between nodes' initial energies or the network density increases. Furthermore, compared with the delay-first strategy, ODA strategy can acquire better advantage on lifetime as the difference between nodes' initial energies increases (i.e. the range of each node's initial energy increases). For the network with high density, the lifetime of ODA strategy achieves at least around 1.5, 2, and 4 times that of delay-first strategy for the heterogeneous cases with the range of (30, 60), (20, 70) and (10, 80) respectively as shown in our simulation results. Therefore, we can conclude that our



(a) DPI of ODA Strategy vs. Density

(b) Lifetime vs. Density with IE = 50

(c) Lifetime vs. Density with IE = (30, 60)

(d) Lifetime vs. Density with IE = (20, 70)

(e) Lifetime vs. Density with IE = (10, 80)

Figure 2: Performance of Delay and Energy Efficiency.

proposed ODA strategy, whose DPI is always only 30%-40% while EPI is always over 50%, achieves a good tradeoff between delay and energy efficiency especially for the cases with high network density and large difference between initial energies of nodes.

V. Conclusion

In this paper, we propose an Energy-efficient On-line Decision Scheme (EODS) which is combined with the design of nodes' duty cycle for the applications of the rare but urgent events detection. We first introduce the Decision-maker Selection Algorithm to find the decision-maker and avoid the redundant transmissions once an event occurs, and then the On-line Decision Algorithm is presented to achieve the tradeoff between delay and energy efficiency. Our simulation results demonstrate that EODS achieves a good balance between delay and energy efficiency.

Acknowledgment

The work is partly supported by China NSF grants (60825205, 61021062).

References

[1] Wei Ye, John S. Heidemann, and Deborah Estrin. *An Energy-Efficient MAC Protocol for Wireless Sensor Networks*. In Proceedings of INFOCOM 2002, pages 1567-1576, June 2002.

[2] M. Stemm and R. H. Katz. *Measuring and reducing energy consumption of network interfaces in hand-held devices*. IEICE Trans. Commun., vol. E80-B, no. 8, pp. 1125-1131, 1997.

[3] Q. Wang, M. Hempstead and W. Yang. *A Realistic Power Consumption Model for Wireless Sensor Network Devices*. In IEEE SECON'06, 2006.

[4] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic. *Towards Optimal Sleep Scheduling in Sensor Networks for Rare Event Detection*. in IPSN05, 2005.

[5] W. T. Rasmussen and H. Robbins. *The Candidate Problem with Unknown Population Size*. Journal of Applied Probability, Vol. 12, No. 4, pages 692-701, 1975.

[6] Y. Gu, T. He, M. Lin, and J. Xu. *Spatiotemporal Delay Control for Low-Duty-Cycle Sensor Networks*. In RTSS09, 2009.

[7] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. *Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks*. In SenSys'03, 2003.