私立東海大學

資訊工程與科學研究所

碩士論文

指導教授：呂芳懌 博士

事件觸發式無線感測網路的
局部移動路由輔助策略之研究

A Routing Assistant Scheme with Localized Movement in Event-driven Wireless Sensor Networks

研 究 生：吳 文 晉

中華民國九十七年七月

摘要

在無線感測器網路中，由於感測器皆是獨立運作，所以搭載電力的有效運用便成為重要的研究議題。但是現今使用在無線傳輸上的路由方法大多是以 hop-by-hop 的方式傳輸封包，而在處於傳輸路徑上的感測器不僅僅要收集自己周圍的環境資訊並將之傳送給基地台，且還要額外負擔其他感測器的資料封包的轉送。如此便會造成無線感測器網路的能量不均衡消耗問，即是靠近基地台附近的感測器會因為不斷協助轉送封包而快速的消耗能量而導致失效，這也是所謂的能量黑洞問題。而這個問題是同時存在於訊息均衡產生及事件觸發式無線感測器網路當中。在這篇論文當中，我們提出一個利用可移動式無線感測器的局部移動路由輔助策略(RASLM)，來減輕事件觸發式無線感測網路中流量負載過高的節點的負擔，以避免因為感測器因為負載過大而過早失效造成感測環境資料的不可靠問題。而這個方法同時可以穩定路由且讓資料可以更順暢的傳遞。實驗結果證實局部移動路由輔助策略(RASLM)可以有效地改善事件觸發式無線感測器網路的系統壽命。

關鍵字：無線感測器網路、移動、路由輔助、事件觸發、能量黑洞

Abstract

In wireless sensor networks (WSNs), energy is one of the most important resources that should be economically used. But most routing approaches deployed by WSNs are hop-by-hop relay schemes, causing that sensors along a routing path should not only collect its environmental data, and then sending the data to base station, but also relay data received from its neighbors toward the base station. This will result in an unbalance energy consumption problem for WSNs, i.e., nodes near the base station will exhaust their energy more quickly than those far away, which is so-called energy hole problem. This problem appears in both message-evenly generated environment and event-driven WSNs. In the paper, we propose a routing assistant scheme with localized movement (RASLM) by using mobile nodes to help nodes along an active routing path to prevent them from dying much earlier than others. This scheme can also stabilize the routing path to ensure that the sensed data can be sent to base station safely and smoothly. The experimented results show that the RASLM can effectively improve the system lifetime in event-driven wireless sensor networks.

Keyword: RASLM, wireless sensor networks, event-driven, movement, energy hole problem

誌謝

首先誠摯的感謝指導教授呂芳懌博士，老師悉心的教導使我得以在短暫的時間內熟悉無線感測網路領域的深奧，不時的討論並指點我正確的方向，使我在這這兩年短暫的光陰中得以獲益匪淺。老師對學問的嚴謹態度及對文字的細心琢磨更是我輩學習的典範。

本論文的完成另外亦得特別感謝極有攻讀博士班潛力的李國財學長大力協助。因為有你的意見及幫忙，才使得本論文能夠更完整而嚴謹。

而在東海九年的日子裡，亦要特別感謝對我照顧有加的朱正忠、林祝興兩位系主任。還有羅文聰、楊朝棟、林正基老師，總在我人生徬徨時給予我適當的建議及指點。系助理施春蘭、林若瑩小姐的照顧及幫忙，葉山、BABY、money、睿秋、雅婷在擔任助教時期共同生活的點點滴滴，都讓我在東海的生活得以絢麗多彩，這將是我人生最難忘的一段回憶，感謝有你們。

另外感謝泰翔、又新、志揚、瑞鈞學長們不厭其煩的指出我研究中的缺失及問題的癥結點，且總能在我迷惘時為我解惑，也感謝小葛、郁婷同學的幫忙，恭喜我們一起順利走過這兩年。而中途轉換跑道的世彥同學，也衷心期盼你的黑人事業早日開花結果。另外實驗室的耀田、永倫、小胖、宏瑋學弟們當然也不能忘記，你/妳?們的幫忙及搞笑我銘感在心，未來資料庫實驗室的歷史就靠你們傳承下去了。

女友在背後的默默支持及陪伴更是我前進的動力，沒有你的體諒、包容，相信這兩年的生活無法過得如此多采多姿。

最後，謹以此文獻給我摯愛的雙親。

List of Contents

List of Figures

1. Introduction

Due to the great advancement in wireless communication and micro electrical mechanical systems [1], Wireless Sensor Networks (WSNs) have been widely studied and developed recently. A WSN often consists of a huge number of sensor nodes, each with sensing, computation, and wireless communication capabilities. Also, a sensor is often powered by battery. Therefore, its power is limited. To overcome power constraint of sensor nodes so as to prolong a WSN's system lifetime and balance workload of its nodes, the ability of movement of sensor nodes or base station (BS) has become one of the hottest research topics nowadays.

Most routing approaches deployed by WSNs are similar to those of Mesh Networks. What a sensor along a routing path should do is collecting its environmental data, and then sending the data to the base station or relaying data received from its neighbors toward the base station. Hence, the nodes near the base station will exhaust their energy more quickly than those far away, which is so-called energy hole problem [16][18]. In addition, many studies on WSNs have tried to prolong a network's lifetime by balancing power consumption among static sinks or nodes. In LEACH [2], each sensor has the same chance to be selected as a cluster head, which is responsible for aggregating data from neighbor nodes and transmitting the data to the base station. Sensors cycle their roles as the cluster head to evenly balance their energy consumption. All nodes in a directed diffusion based sensor network are application-aware [3], which enables diffusion to achieve energy saving by selecting good paths and caching and processing data in whole network instead of sending data back to base station blindly. However, in a uniformly distributed and homogeneous-node environment, if static nodes or static sinks are used, energy hole near base station can not be avoided [4]. Many more and more studies recommend using mobile sensors or sinks to prolong the network lifetime [7][10][11][12][14].

To follow the trend of development in this paper, we propose a routing assistant scheme which deploys mobile nodes with localized movement (RASLM) in an event-driven WSN to prolong lifetime of a node *N* that has suffered from relaying too many packets. The mobile nodes located near *N* when necessary can move toward *N* and share relaying burden with *N* so as to mitigate *N's* energy consumption speed. Experimental results show that RASLM can effectively prolong lifetime of a WSN and stabilize packet routing of a busy routing path.

The rest of the paper is organized as follows. Section 2 introduces background and related work of the paper. In Section 3, we explain the preliminaries used in this study. Section 4 presents the detail of RASLM scheme. The performance of the proposed scheme is evaluated in section 5. Section 6 concludes this paper and addresses our future work.

2. Background and Related Work

In this section, we classify studies relevant to mobility of wireless sensor networks into two categories, those based on mobile sinks (base stations), and those based on mobile nodes.

2.1 Mobile Sink

Many researchers have tried to improve a WSN's lifetime by using single or multiple mobile sinks. Luo and Hubaux [4] demonstrated that using a mobile sink is helpful to enhance the WSN's lifetime. If the sink's locus follows the periphery of the WSN, its energy consumption efficiency will be better. The TTDD [6] protocol built a grid structure and divided a network into regular cells with several dissemination nodes. The dissemination nodes are responsible for relaying the query and data to and from the proper sensors. Multiple mobile sinks moved around their sensing field to continuously receive data with local cells only. Wu et al. [7] proposed a method called Dual Sink by combing mobile and static sinks to change routing destination. A node forwards packets to one of the sinks according to the distance between it and the sinks. This approach truly extends the lifetime of a wireless sensor network, but too much overhead is required to maintain the routing path. Bi et al. [14] claimed that a sink moving along fixed tracks lacks flexibility and scalability. Hence, they proposed two moving schemes for a sink to alleviate the hotspot problem and prolong network lifetime. In MAEC [8], Zhao et al. proposed a base station movement-assisted energy conserving method to prolong lifetime of an event-driven WSN by directing the base station moving close to the hotspots. This scheme is effective when the events happens rarely and centralizes. Once the number of hotspot areas increase, it would no longer effective. Wang et al. [9] proposed an optimal predefined mobility trajectory of the base station to balance the energy consumption of sensors in event-driven WSNs. The approaches above are all proposed on a static sensor environment.

2.2 Mobile Sensor Node

Unlike the sink mobility, there are a lot of studies trying to solve unbalanced energy consumption problem for WSNs. Yang et al. [10] proposed a local load balance solution by modifying the Hungarian-method-based optimal solution, and integrating with movement assisted sensors to redeploy their positions into a balanced grid. Yang and Cardei [12] partitioned a WSN into several coronas to analyze their energy consumption and proposed a redeployment scheme that moves sensors from outer coronas into inner coronas if necessary so as to lower a node's data generating rate and balance energy consumption rate among nodes. Leu et al. [13] deployed a polar coordinate system to identify a mobile node's position and route network packets. However, they all focus on evenly-message-generated WSNs rather than event-driven WSNs.

3. Preliminaries

Energy consumption in an event-driven environment and in a message-evenly-generated environment is different in that in an event-driven environment, e.g. the scene of fire, only the nodes on the routing path from an event-triggered node to the base station need to relay messages. Other nodes will work lightly, e.g., only listening to its upstream nodes if they are not triggered or requested to relay packets. In the following, we will use nodes and sensors interchangeably and analyze the energy a sensor may consume in an event-driven environment.

3.1 Event-driven Environment

As mentioned above, most of the studies that have tried to overcome the unbalanced energy consumption problem are performed under message-evenly-generated environments, like those in [11][12][14][15]. However, in some cases, e.g., a WSN that is deployed to detect scene of fire in a manufacturing factory, hackers' intrusion in an autonomous network system, or amount and type of enemy in a battlefield, sensors work only when they are triggered by particular events. Therefore, messages are not evenly and periodically generated.

In such a network, several sensors, e.g., nodes A, B, D and W shown in Fig. 1, might be triggered at the same time. They forward their data packets/messages to the base station through a hop-by-hop routing approach. Node A generates and forwards its sensing data to the base station. The packets it generates need to be relayed six times before they can arrive at the base station. Node C only relays packets for, e.g., nodes A and B. Node D, in addition to generating and transmitting its packets, also relays packets generated by nodes A and B. Node B does the same, but it only relays packets for node A. The workload and power consumption of nodes A, B, C, D and E are different.

Obviously, nodes close to base station will exhaust their energy far faster than those far away from the base station. An active routing path (an active path in short) *P* in a WSN is a routing path established when *P*'s source node, e.g., node *A* or node *W*, is triggered by its surrounding events. The source node then broadcasts a route request packet. After the events are solved or removed, and no packets are transmitted through *P*, *P* will be in an inactive state.

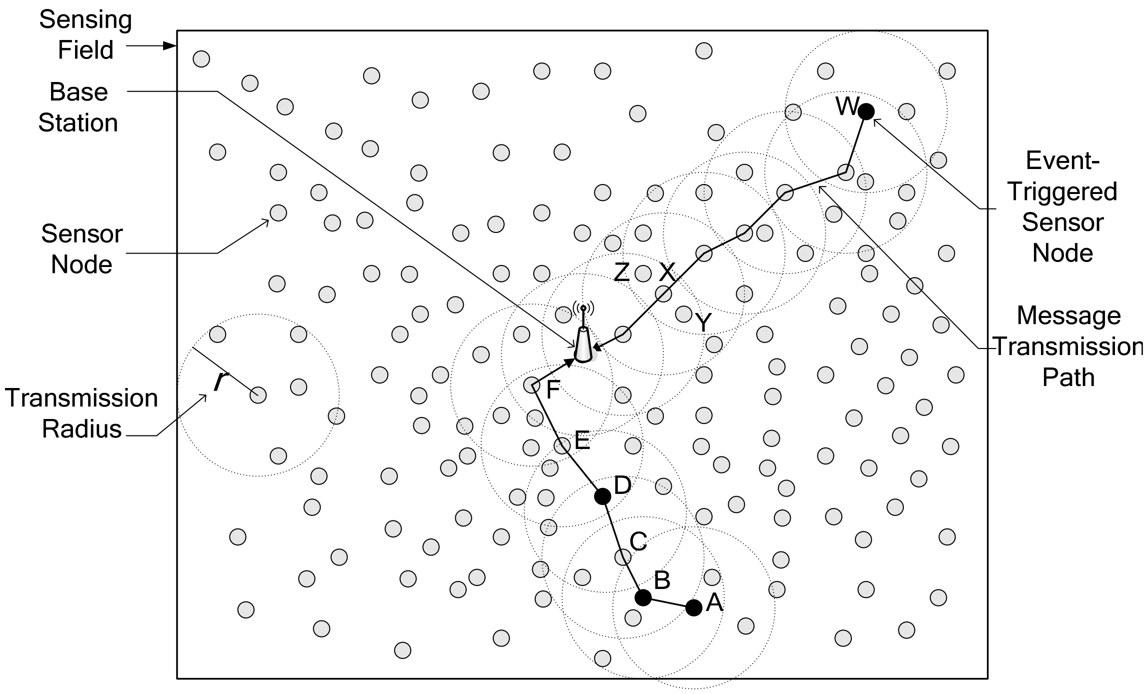


Fig. 1 An event-driven based wireless sensor network whose nodes are randomly distributed

3.2 Energy Consumption

In this paper, we assume that transmitting a data packet consumes much more energy than that of receiving a data packet, sensing environmental data, routing information maintenance, and other necessary expenditures [16][17]. We can realize that if a node *N* continuously transmits data packets to the base station through a routing path *P*, *P* will soon exhaust all their energy and die together with node *N*.

Hence, a routing path should not be permanently sustained. In fact, we can measure a node *N*'s energy consumption rate by monitoring *N*'s packet flow periodically. *N*'s workload can be analyzed under the following assumptions. (1) all nodes in underlying sensing field are homogeneous and their initial energy levels are the same, e.g., ε; (2) the communication radiuses of base station and sensor nodes are the same, e.g., r; (3) a node when sending and receiving a data packet has to respectively consume e_1 and e_2 units of energy for each data bit [18], where $e_1>e_2>0$; (4) the lifetime of a WSN is defined as the duration from when the network starts up until the first node exhausts its energy; (5) in order to simplify the complexity of event-driven scenario, we assume that an event-driven sensor keeps sends a data packet of *Data_Length* bits per round to base station, and sustains R rounds from when events occur to the events are solved or removed, and the time interval of one round is set to T_m seconds so that each time a sensor is triggered by an event or events will generate R data packets in $R \cdot T_m$ seconds.

In Fig. 1, node A only transmits its data packets without relaying packets for others. Hence, its energy consumption rate is $\frac{Data\_Length \cdot e_1 \cdot R}{R \cdot T_m} = \frac{Data\_Length \cdot e_1}{T_m}$. Nodes B's and C's energy consumption rates are $\frac{Data\_Length \cdot (2e_1 + e_2)}{T_m}$ and $\frac{Data\_Length \cdot (2e_1 + 2e_2)}{T_m}$, respectively. There is no doubt that node B will exhaust its energy before node A dies. Nodes D and E work on $\frac{Data\_Length \cdot (3e_1 + 2e_2)}{T_m}$ and $\frac{Data\_Length \cdot (3e_1 + 3e_2)}{T_m}$, respectively. Assume that along a path segment $\{x_0, x_1, x_2, \cdots x_{k-1}, x_k\}$ of an active routing path $\{x_0, x_1, x_2, \cdots x_{k-1}, x_k, x_{k+1}, x_{k+2}, \cdots, x_g\}$ of length g, there are u sensor nodes and $k+1-u$ relay nodes, where x_0 is the source node which of course is a sensor node, then, energy consumed by a node x_k, which is k nodes away from x_0, is

$$\frac{Data\_Length \cdot (u \cdot e_1 + j \cdot e_2)}{T_m} \tag{1}$$

If x_k is a sensor node, then $j=u-1$, otherwise $j=u$.

3.3 Energy Hole Problem and Non-Surveillant Zone

From the analysis above, we can realize that the path from node *A* to base station in Fig. 1 will fail on nodes *E* and *F*. Energy consumption rates of nodes *E* and *F* are the same. It is clear that burden of a node much closer to the base station will be much heavier. This is a common phenomenon, i.e., energy hole problem [16], in a message-evenly-generated WSN [18][19][20]. However, in an event-driven WSN, if those nodes triggered by events continue to transmit their sensing data to the base station through one or several routing paths, some nodes, like nodes *E* and *F* in Fig. 1, will die soon, producing a non-surveillant zone (NSZ), i.e., energy hole problem also exists in event-driven environments [17]. Situation becomes worse when nodes *E* and *F* are articulation node. An articulation node is a node that joins at least two upstream paths and at least one downstream path. This might be very dangerous for critical monitoring missions, such as fire monitoring and detection, and chemical/nuclear reaction detection. So, to prolong the lifetime of an event-driven WSN, we have to first reduce energy consumed by articulation nodes which are very often not far away from the base station.

A node can determine whether it is an articulation node or not by two ways. One is computing its energy consumption rate mentioned above which is much larger than that of anyone of its upstream nodes. The other is checking the packets it has received. The latter can be realized by two ways. The first is when base station chooses one path as the routing path, an ACK will be sent back to source node reversely through the path. An articulation node will receive at least two such ACKs with different destination addresses (i.e., the source addresses of the paths). The second is during routing it will receive data packets of different source addresses.

In this study, a node computes its energy consumption rate periodically. A node N with a very high energy consumption rate will soon exhaust its energy. However, if N's nearby "duty-free" nodes can share burden of relaying packets with it, N's lifetime and of course the WSN's lifetime can be effectively prolonged.

3.4 Mobility Models

To reduce the probability of forming NSZ or energy holes in an event-driven WSN, we assume that when events occur all its sensors are able to move to right and accurate positions by themselves. We use the mobility model of sensors similar to that defined by Chellappan et al. in [21] to migrate sensors. In this model, sensor mobility is something like a flip, and they assume that a sensor can move from its current position to a new location by using the power of propellers, fuels, coiled springs unwinding during flips, external agents launching sensors, etc, implying this sensor's movement consumes no battery energy of its own.

Several studies discuss or implement the mobility of sensor nodes. The maximum moving distance by using motor to move along a horizontal string of a sensor in [22] is 165 meters. In DARPA's self-healing minefield program, the mobility of sensors can be up to 100 hops by fuel-propeller [23].

4. Proposed Protocol

In the following, we would like to introduce how RASLM prolongs lifetime of an articulation node on an active path.

4.1 Effective Assistant Zone

In RASLM, when a node *N* on an active path realizes that its energy consumption rate is higher than a predefined threshold δ, it broadcasts a Request-Assist Packet (RAP) to request its neighbors to come to relay packets for it. The effective assistant zone (EAZ) of a node, e.g., node *y*, as shown in Fig. 2, is defined as the intersection region of communication ranges of node *y*'s immediate upstream and immediate downstream nodes i.e., nodes *z* and *x*, respectively. Node *y*'s EAZ is the only region that its neighbors coming to help it can communicate with both it immediate upstream and downstream nodes.

A node with higher energy consumption rate needs many more neighbors. A node, e.g., node *h*, on receiving a RAP sent by node *y*, may possibly move into node *y*'s EAZ in which *h* can share the burden of relaying packets with *y*. Here, those neighbors coming to help y are called *y*'s assistant nodes.

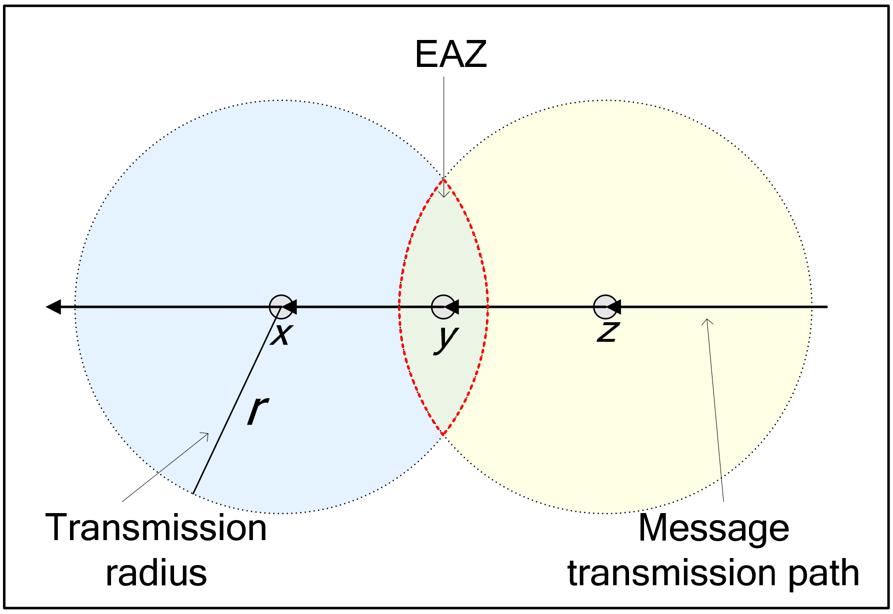


Fig. 2 Node y's EAZ is the intersection region of node x's and z's communication ranges

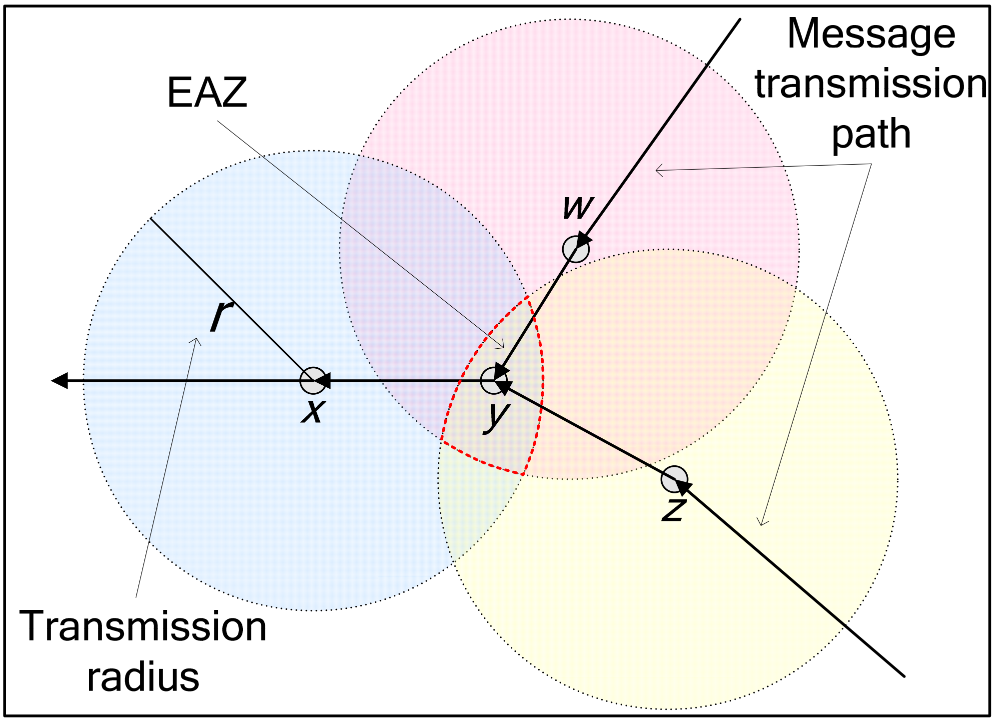


Fig. 3 Node *y*'s EAZ is the intersection region of nodes *x*'s, *w*'s and *z*'s communication ranges

The shapes of EAZ are different if number of *y*'s immediate upstream nodes varies. Node *y* in Fig. 3 has two immediate upstream nodes. In Fig. 4, *y* has three immediate upstream nodes. From Fig. 2 to Fig. 4, we can realize that shape of an EAZ is often irregular. When number of immediate upstream nodes increases, area of the EAZ shrinks. Once a neighbor node *N* on receiving a RAP packet decides to help the packet sender, it compares its current position, which is obtained by using its own positioning service, such as Global Positioning System (GPS), Ad hoc Positioning System (APS) or other positioning methods, and the sender's location, which is recorded in the RAP packet, to determine which direction it should move to. The criteria that node *N* uses to determine whether it is able to move to help the sender or not will be described later. Of course, nodes that are already in this EAZ can help the sender directly without any movement. Of course, due to some reasons, they can also refuse to help the sender.

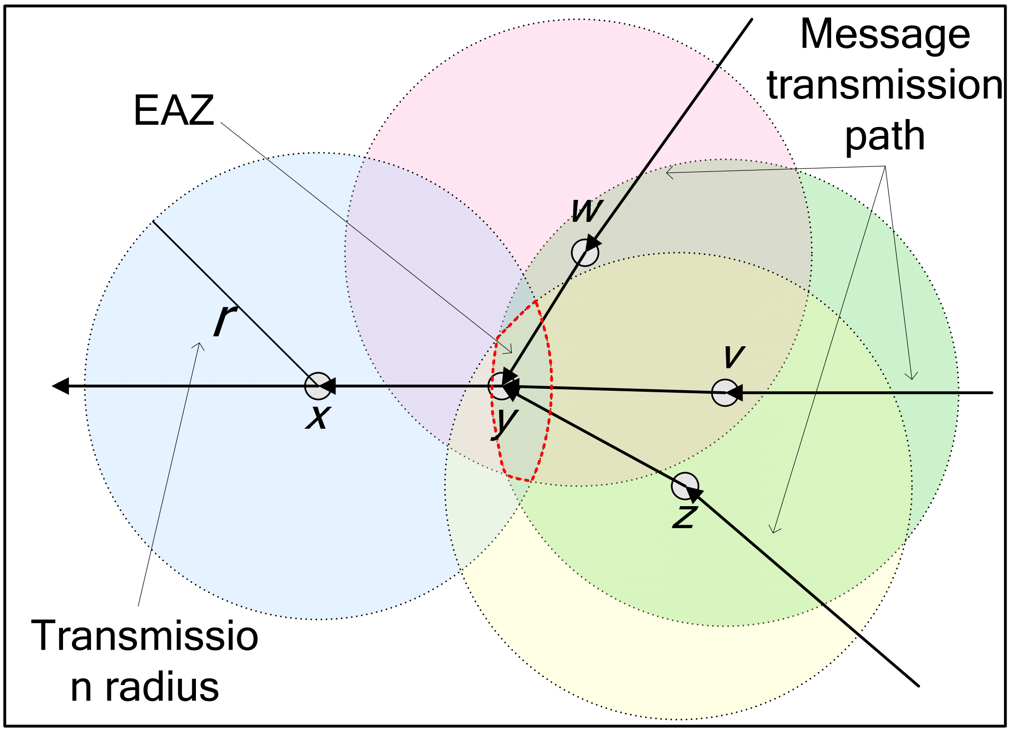


Fig. 4 The node *y*'s EAZ which is the intersection of the transmission ranges of nodes *v*, *w*, *x*, and *z*. It is an irregular region.

4.2 Broadcasting RAPs

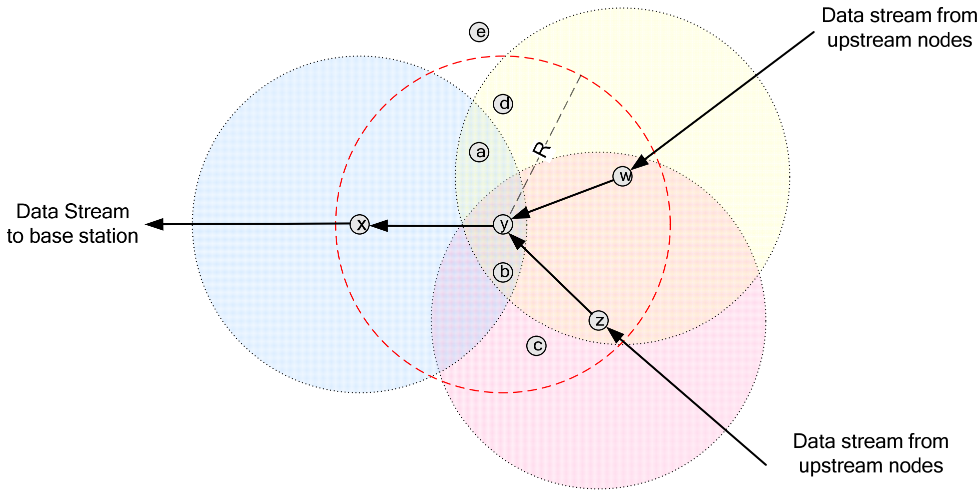
A RAP packet as shown in Fig. 5 consists of five main fields. The first is RAP-ID which comprises two sub-fields, sender ID and serial-number. The former shows who sends the packet, whereas the latter is a number used to discriminate different RAPs. Each time when a node broadcasts a new RAP, it increases its serial number by one to ensure the uniqueness of the RAP-ID. The second field is the sender's coordinates with which an assistant node can realize where it should move to. The third field is TTL (time to live) value which is used to control the broadcast range of a RAP packet. When the energy consumption rate of a node is higher, it broadcasts a RAP packet with a higher TTL so that many more nearby nodes can receive the packets and then come to help the packet sender. The fourth field is a list of direct neighbors which records the sender's immediate upstream and downstream nodes. An assistant node on deciding to help the sender of a received RAP packet must keep its antenna working while moving toward the sender. Once they have received all the radio signals sent by those nodes listed in the RAP received, it can realize that it is now in the EAZ, thus stopping its movement. This is helpful in shortening an assistant node's moving distance, consequentially saving its moving energy. The last field is timestamp T_s which is used to tell receiving nodes that this call for help will time out at timestamp T_s+T_m, where T_m, a predefined time period, as stated above is time interval of a round.

RAP-ID		Sender's Coordinates	TTL	List of direct neighbors	Timestamp
Sender ID	Serial-Number				

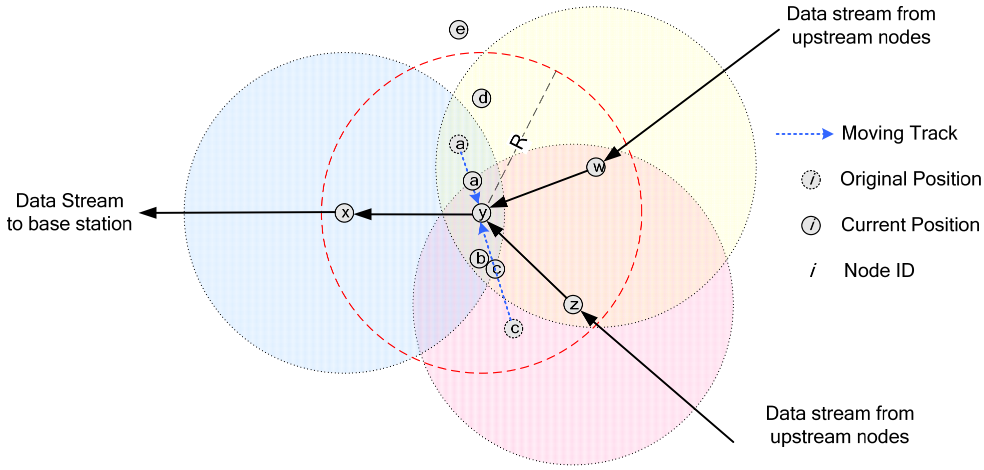
Fig. 5 Format of a Request-Assist Packet (RAP) packet

4.2.1. Call for Help

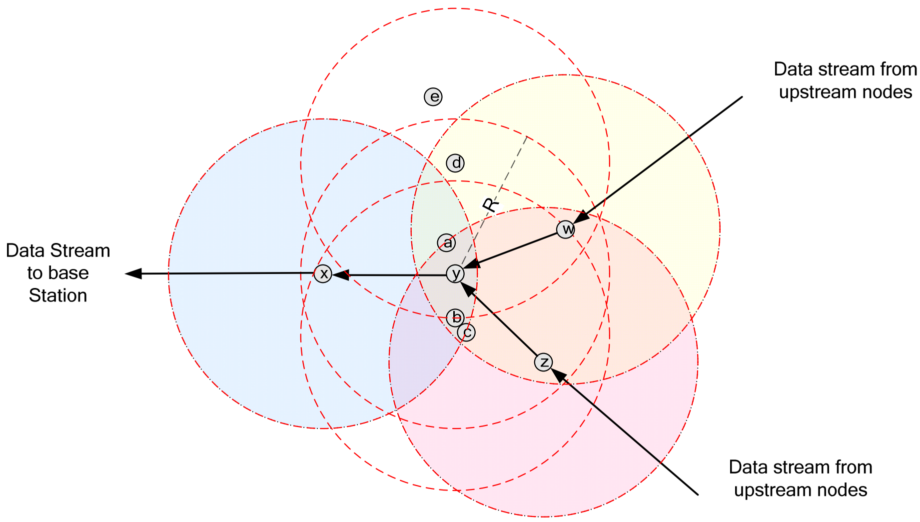
Fig. 6a gives an example of call for help. Node y is an articulation node of two routing paths. It needs to relay packets received from nodes *w* and *z*. After it broadcasts a RAP packet with TTL=1, and its direct neighbors *a* and *c* on receiving the RAP decide to help it, then a and c start moving toward *y*, and stop at the boundary of *y*'s EAZ. Node *d* does not accept the request owing to some reasons which will be described later. Node *b* is already in the EAZ. It, according to its current situation, may or may not help to relay packets. If it decides to help *y,* no movement is required. Node *e* does nothing due to receiving no RAP from *y*. After the first call for help, the network topology becomes the one shown in Fig. 6b.



(a) Step 1



(b) Step 2



(c) Step 3

Fig. 6 An example of RAP broadcast procedure

4.2.2. Handling a RAP Request

A node currently on its way to help other node will not accept other RAPs. A node on an active path (called an active node) will not act as an assistant node. A non-active node on receiving a RAP, checks T_m times out or not. If not, it further looks up its history list to see whether the RAP exists or not. If the RAP truly exists, the RAP will be discarded, otherwise, the node autonomous determines whether it can go to help the RAP-sender or not, by checking to see if it has sufficient time to move? sufficient moving energy? sufficient energy to relay packets? Once it decides to go, it starts for the destination EAZ. A node, no matter it is an active node or not, on receiving a RAP will simply check the TTL value. If $TTL-1>0$, it records the RAP-ID in its history list, decreases TTL value by one, and rebroadcast the RAP. If not, it discards the RAP.

Once the predefined time period $2{\cdot}T_m$ times out after a RAP is sent out, the sender, e.g., node *y*, arranges the relaying schedule with a first-come-first-served approach for all its assistant nodes, and broadcasts its routing table and its unique ID. Assistant nodes can then obtain their routing tables without exchanging routing information with their new surrounding nodes. Since node *y*'s assistant nodes relay packets for node *y*, contents of their routing tables should be the same. This broadcast can speed up the initial phase of call for help which is the period of time from when y sends out a RAP to the time point that all assistant nodes start to relay packets. After the initial phase, node y enters its working phase, in which it relays packets together with all assistant nodes. If no packet has been sent through underlying EAZ for a defined period of time, e.g., since events have been solved or removed or an upstream node dies, y and the assistant nodes stop relaying packets. Now, an assistant node can decide to stay at its current location, move back to its original position, or migrate to a new EAZ to help another articulation node.

Each time when node *y* finishes scheduling relaying task for a call for help, it re-calculates the energy consumption rate for itself by using the formula

$$C_y = \frac{Data\_Length \cdot (i \cdot e_1 + j \cdot e_2)}{T_m \frac{\lfloor er_0 \rfloor}{\sum_{t=0}^{m} \lfloor er_t \rfloor}} \quad (2)$$

where er_0 and er_i respectively denotes *y's* residual energy and node *i's* residual energy, i = 1 to m. If C_y is higher than its threshold value $\boldsymbol{\delta}$, node y will generate a new RAP with new serial number, and increases TTL. This time the broadcast range as shown in Fig. 6c covers node *e*, which on receiving the RAP will determine whether it is able to help node *y* or not. Node y repeats the procedure until C_y is lower than δ.

Here, we further assume that

(1) Each mobile node can move up to *M* meters with speed *V* m/sec by using fuel-propeller. The maximum value of *M* is set to *l* where $\frac{l}{V} \le T_m$, i.e., a mobile node is able to move to any place in half field in T_m seconds.

(2) Each node has the ability to move back to its original or new position.

(3) The X-Y coordinate system is used.

(4) The base station is placed at the center of the sensing field.

(5) T_m is long enough for an assistant node *P* to move to RAP-sender *S's* EAZ.

(6) The coordinates of node *i* are represented by (x_i, y_i).

Hence, the Euclidean distance between *P* and *S* can be calculated by

$$D_{p\_s} = \sqrt{(x_p - x_{s)}^2 + (y_p - y_s)^2} \quad (3)$$

when $D_{p\_s} \le \frac{M}{2}$ and $\frac{D_{p\_s}}{V} \le T_m$, *P* will insert the RAP into its candidate list for a final decision, going to help S or not and to help which *S* if *P* has received several RAPs at the same time. The algorithm for node *P* on receiving a RAP is shown in Fig. 7.

After collecting RAPs from several *S*s for a predefined period of time T_m counting from the smallest T_s among all RAPs collected in the candidate list, P calculates D_{p_s} for each S in the candidate list, picks up the nearest S whose $D_{P\_S} \leq \frac{M}{2}$, and then moves toward the S. The algorithm for P to process the candidate list is shown in Fig. 8.

Algorithm: Algorithm for a node *P* when receiving a RAP packet.

Input: Receiving a RAP with a source address and T_s sent by node y.

Output: a candidate list.

If (*RAP_ID(y)* exists in *History_List(P)* or T_s+T_m times out) **then**

{ *DiscardMsg*(RAP); Return; }

If (*ActiveStateCheck(P)* or *OutofMovableDistance*(RAP) or *TimeNeededtoMove*(RAP)>T_m or *InsuffEnergytoRelayPacket(P)*) **then**

/* Criteria to determine if *P* can go to help *y* or not: *P* is an active node, distance between *P* and $Y > \frac{M}{2}$, *P* can not move to *y*'s EAZ within time interval T_m, or *P* has insufficient energy to relay packets for node y */

{**If** ((*TTL-1*) > 0) **then**

{*ReBroadcast*(RAP); *Return*;}

else {*drop*(RAP); Return; }}

else

{ *RAP_COUNT(P)*++; /* counting number of call for help */

AppendToCandidateList(RAP); }

Fig. 7 Handling algorithm for a node *P* when receiving a RAP packet

Algorithm: Algorithm for a node P to handle Candidate List.

Input: In Candidate List, $T_s'+T_m$ times out where T_s' is the smallest T_s among RAPs that P has received.

Output: P starts for node y's EAZ.

1: **If** *RAP_COUNT*(P)≥2 **then**

2: { *SelectNearestRAP*(CandidateList);

3: *DropOthers*(other RAPs);}

4: P starts moving forward y's EAZ; Return;

Fig. 8 The algorithm for a node P to process its Candidate list when $T_s'+T_m$ times out, where T_s' is the smallest T_s among RAPs that P has so far received

4.3 A Relaying Schedule and Handling Flow of Assistant Nodes

When P enters the EAZ, it sends a RAP-ACK to S, and waits for receiving a relaying schedule from S. A RAP-ACK has at least three fields, including its Node-ID, current coordinates and its residual energy er. After S sends out a RAP, and the corresponding $2 \cdot T_m$ expires, S starts to schedule service time for each P and itself. Assuming that there are a total of q assistant nodes $\{P_1, P_2, ..., P_q\}$ already in or arriving at node S's EAZ. Node S firstly calculates the Greatest Common Divisor for their residual energy er_0, er_1, er_2, …, er_i,…, er_q. A cycle time T_c for relaying packets for S is defined as

$$T_c = \frac{\sum_{i=0}^{q} \lfloor er_i \rfloor}{GCD\left[\lfloor er_0 \rfloor, \lfloor er_1 \rfloor, \lfloor er_2 \rfloor, \cdots, \lfloor er_q \rfloor\right]} \times T_m \tag{4}$$

where T_m is a round time. Numbers of rounds that P_i and S should do, denoted by $N_R(i)$ and $N_R(S)$, respectively, in each cycle are

$$N_R(i) = \frac{\lfloor er_i \rfloor}{GCD\left[\lfloor er_0 \rfloor, \lfloor er_1 \rfloor, \lfloor er_2 \rfloor, \ldots, \lfloor er_q \rfloor\right]}, i = 1, 2, \ldots, q \quad (5)$$

and

$$N_R(S) = \frac{\lfloor er_0 \rfloor}{GCD\left[\lfloor er_0 \rfloor, \lfloor er_1 \rfloor, \lfloor er_2 \rfloor, \ldots, \lfloor er_q \rfloor\right]} \quad (6)$$

and the workload of *S*, denoted by W(S), is calculated as

$$\begin{aligned} W(S) &= \frac{Data\_Length \cdot (u \cdot e_1 + j \cdot e_2)}{T_m} \cdot \frac{N_R(S) \cdot T_m}{T_c} \\ &= \frac{Data\_Length \cdot (u \cdot e_1 + j \cdot e_2) \cdot N_R(S)}{T_c} \end{aligned} \quad (7)$$

Once, $W(S) > \delta$, *S* broadcasts an RAP with a new TTL_{new} ($=TTL_{init}+1$). The procedure repeats until $W(S) \leq \delta$.

4.3.1. TTL Values

Generally, the initial value of TTL field is one. The value can be either static or dynamic. When the network scale and event-triggered ratio are relatively small, RAPs are broadcasted with a static TTL value, i.e., TTL value is fixed to a constant, e.g., 1 or 2, since in the local area there are enough nodes will come to help an RAP-sender. This can save some levels of broadcasting overheads and then prolong underlying network's lifetime.

But, when the network scale is large, the base TTL value is dynamically increased according to the workload of the RAP-sender. Before the initial call for help, the RAP-sender *S* evaluates burden of itself by using formula (1), and broadcasts an RAP packet with TTL_{init}. If the burden calculated by using formula (7) is still higher than its threshold, *S* broadcasts a new RAP with a $TTL_{new}=TTL_{init}+1$ to call for many more assistant nodes. Values of TTL_{new} range between 1 and $\left\lceil \frac{l}{r} \right\rceil$, where *l* is the side length of underlying sensing field, and *r* is the transmission radius of a sensor node.

When a RAP with TTL=1 is broadcasted, the maximum number of assistant nodes that can come to help S is $m \cdot \frac{\pi \cdot r^2}{l^2}$ nodes, where m is number of nodes distributed to the sensing field. When the TTL increases to i, $0 > i \geq \left\lceil \frac{l}{r} \right\rceil$, maximum number of nodes that may come increases to $m \cdot \frac{\pi \cdot i^2 r^2}{l^2}$, and the initial TTL should satisfy, $TTL_{init} \geq \sqrt{\frac{W(S) \cdot l^2}{m \cdot \delta \cdot \pi \cdot r^2}}$, where *W(S)* denotes the original workload of *S*.

4.3.2. Timeline for Call-for-Help

The timeline for call for help and the corresponding activities are shown in Fig. 9. *S* firstly schedules relaying task for each P_i based on the level of energy that P_i has, and broadcasts a clock signal to synchronize all the $P_i s$. After all $P_i s$ finish calibrating their clocks, *S* broadcasts the schedule to all $P_i s$. P_i on receiving the schedule replies with a Sch-ACK and starts following the schedule to relay packets.

Each time, after *S* calculates *W(S)*, it checks to see whether its energy consumption rate and residual energy are individually higher than their corresponding thresholds. If yes, *S* and its $P_i s$ continue their relaying task. If not, *S* generates a new RAP, and broadcasts it to call for many more assistant nodes to come in order to prevent original assistant nodes and *S* from dying quickly. Meanwhile, all $P_i s$ keep working until receiving a new schedule from *S*. However, during relaying packets, due to some reasons, e.g., an upstream node dies, the corresponding route is then changed. The relaying task for the upstream path stops. Of course, source node of the path will look for another path. If one node of the new path finds that its energy consumption rate is higher than a predefined threshold, it in turn will act as *S*. The algorithms used by *S* to schedule relaying task and broadcast the schedule to assistant nodes are shown in Fig. 10, and Fig. 11, respectively.

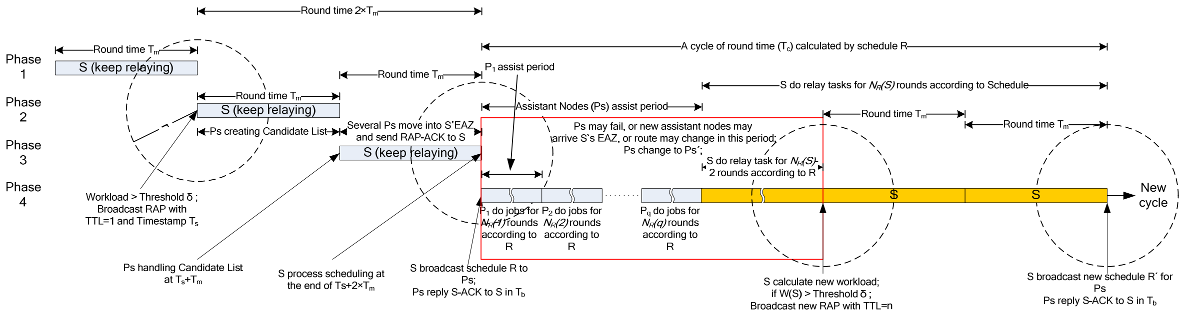


Fig. 9 Timeline expressing the RAP request and assistant nodes handling points

Algorithm: Scheduling(*Ps′*, *Ps*): Algorithm for node *S* to schedule relaying tasks when $2{\cdot}T_m$ times out after sending out a RAP.

Input: Existing assistant nodes $Ps'=\{P_1',P_2',\ldots,P_k'\}$; /* those assistant nodes already in underlying EAZ */ a set of new arriving assistant nodes $Ps=\{P_1, P_2,\ldots,P_q\}$.

Output: Broadcasting Schedule to assistant nodes and/or a new RAP.

1: **If** ($Ps \neq \phi$) **then** {CalculateEnergyLevel(*Ps′*); /*calculate residual energy er_i' for P_i' in *Ps′*, $i=1,2,\ldots,k$ since they have consumed some levels of energy for relaying packets */

2: T_c=*CalculateCycleTime*($S, Ps' \cup Ps$); /* calculate a cycle time using formula (4) */

3: *CalculateRounds*($S, Ps' \cup Ps, T_c, T_m$); /* calculate number of rounds for each P_i and S by using formulas (5) and (6), respectively */

4: Schedule the relaying tasks for all members of $Ps' \cup Ps \cup \{S\}$ based on $N_R(i)$ and $N_R(S)$ calculated;

5: *BroadcastSchedule*(*Schedule*, $Ps' \cup Ps$); /* broadcast the schedule arranged to all assistant nodes */

6: *CalculateW(*S); /* calculate energy consumption rate $W(S)$ for S using formula (7) */

7: **If** ($W(S)>\delta$) **then** { /* workload is not lower than threshold δ */

8: *GenerateRAP(TTL+1)*;

9: *Broadcast*(RAP); }}

Fig. 10 Algorithm for a node S to schedule relaying tasks when $2{\cdot}T_m$ times out after sending out a RAP

Algorithm: *BroadcastSchedule*(*R, Ps*): Algorithm for node *S* to broadcast schedule assistant nodes.

Input: The schedule arranged, e.g., *R*, and all assistant nodes *Ps*.

1: Set T_b; /* a timer for broadcasting the schedule */

2: Broadcasting *R*;

3: S-ACK=ϕ;

4: **While** (T_b does not time out)

On receiving an Sch-ACK from P_i, S-ACK=S-ACK$\cup\{P_i\}$;

5: **If** ($Ps \neq S\text{-}ACK$) /* not all assistant nodes have received the *R*, e.g., they are *Ps"* */

Scheduling(*Ps*-*Ps"*, ϕ); /* rescheduling */

Fig. 11 Pseudo code for node *S* to broadcast the schedule arranged

5. Experiment and Discussion

A total of five experiments are performed in this study. The first evaluates the relation between a square and a round sensing field, and observe how nodes and, event-triggered nodes are distributed in such an environment. The second studies what the load of a node in this environment is. The third addresses lifetime of RASLM. The fourth evaluates how nodes are distributed after RASLM is enabled in WSN. The fifth experiment measures the system lifetime on different ratios of failed nodes to further validate the effectiveness of AODV and DSDV routing protocols integrated with RASLM. In the first two experiments, simulation programs are developed by using language C++ to eliminate effects generated by the MAC and PHY layers and not considered when dealing with the relation between events and node distribution.

5.1 Randomly Distributed Sensor Environment

In the first experiment, we would like to evaluate how many nodes are triggered in each corona. Initially, 49 sensors were randomly distributed to a 1000×1000 m^2 sensing field. The transmission radius of a sensor is set to $r=250$m. The whole sensing field was partitioned into N coronas $\{c_1, c_2, ..., c_N\}$, where c_i is the region between $c_{i-1} \times r$ to $c_i \times r$. c_0 is the center of the field. Node 0 is placed at c_0. Hence, there is a total of 50 nodes on the field. 5% to 40% of sensors are triggered by events. The experiment was performed 100 times and each time event-triggered nodes are randomly selected. In Fig. 12, C*i*-Nodes represents the total number of nodes distributed to corona *i*, and C*i*-ETN means number of triggered nodes in corona *i*, *i*=1,2,3.

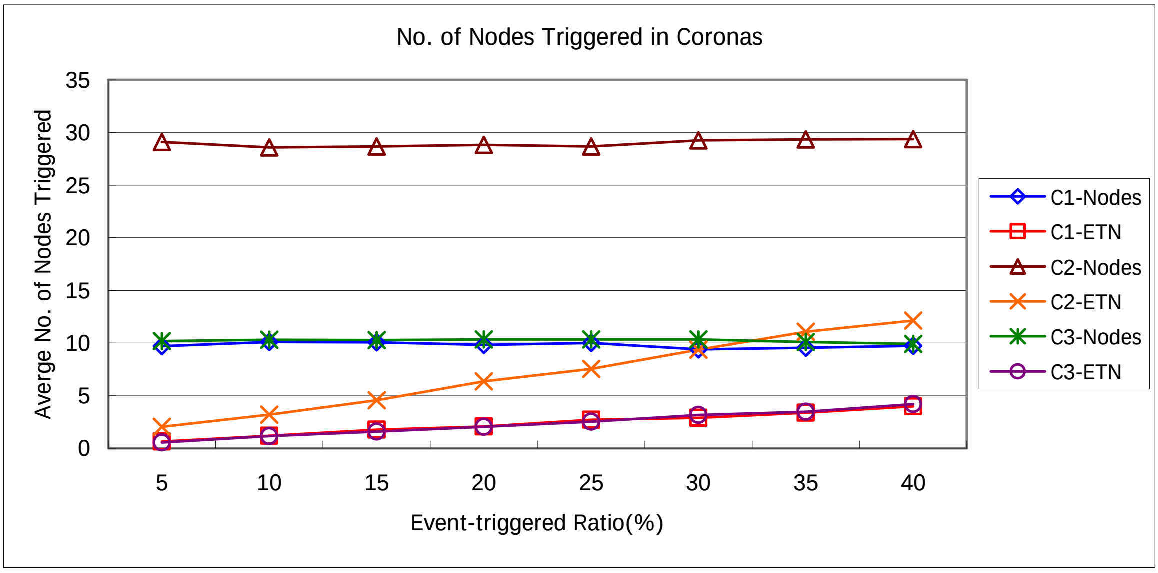


Fig. 12 A total of 49 nodes are randomly distributed to a 1000×1000 m^2 sensing field, with different event-triggered ratios ranging from 5% to 40%. C*i*-Nodes represents total number of nodes distributed to corona 1, and C*i*-ETN means number of triggered nodes in corona *i*, *i*=1,2,3.

Theoretically, corona 3 should be the one with the largest amount of nodes. However, in Fig. 13, we can see that the most number of nodes are distributed to corona 2. This is because we deploy a square field and partition it into coronas. The outermost corona goes beyond the effective region of the square.

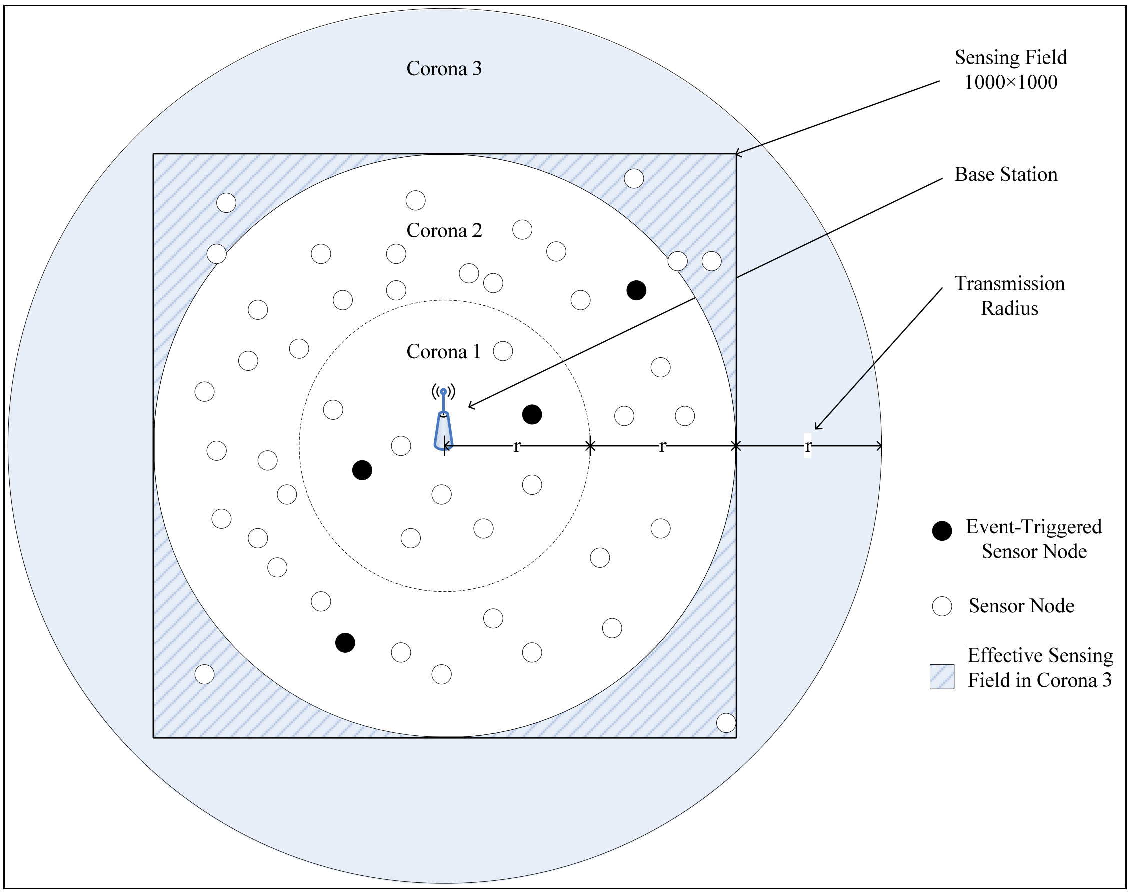


Fig. 13 49 nodes are random distributed to a 1000×1000 m^2 sensing field, and the ratio of event-triggered nodes is set to 5%, i.e., 4 sensors are triggered, 2 in corona 1, and the other 2 in corona 2

In general, number of nodes distributed to a corona is proportioned to the corona's area. The areas of coronas 1, 2, and 3 are πr^2, $(4-1)\pi r^2$, and $(9-4)\pi r^2$, respectively. But actually, corona 3 has only $4r^2(4-\pi)$. So the average ratios of nodes distributed to coronas 1, 2, and 3 are respectively *1:3:1.093*. The average numbers of nodes really triggered in coronas 1, 2, and 3 are 0.64, 2.03 and 0.54 nodes, respectively. We can conclude that number of sensors distributed to a corona is proportional to its effective area, but excluding outermost corona. However, if we further upscale the sensing field to 2000×2000m^2, the number of coronas partitioned will be $4=\frac{2000}{2\times 250}$. But this is not true.

The number of coronas as shown in Fig. 14 is upscaled to $\left\lceil \frac{1000 \times \sqrt{2}}{250} \right\rceil = 6$. Its experimental results are shown in Fig. 15, not only indicating the reason why corona 4 has many more event-triggered nodes than those in coronas 5, and 6, but also implying that nodes in inner corona of a randomly distributed sensor network have to relay packets for outer coronas. This is why in an event-driven WSN, sensors in inner coronas, like that in a message-evenly-generated WSN, will also die earlier than those in outer coronas.

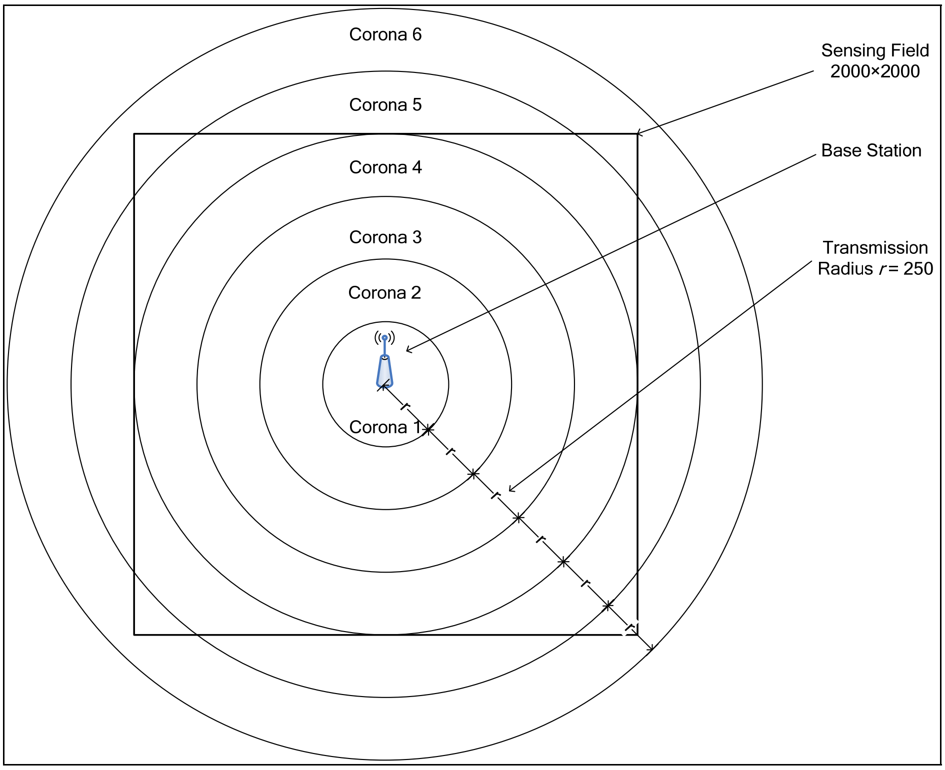


Fig. 14 A 2000×2000 m^2 sensing field is partitioned into 6 coronas with width r=250 m

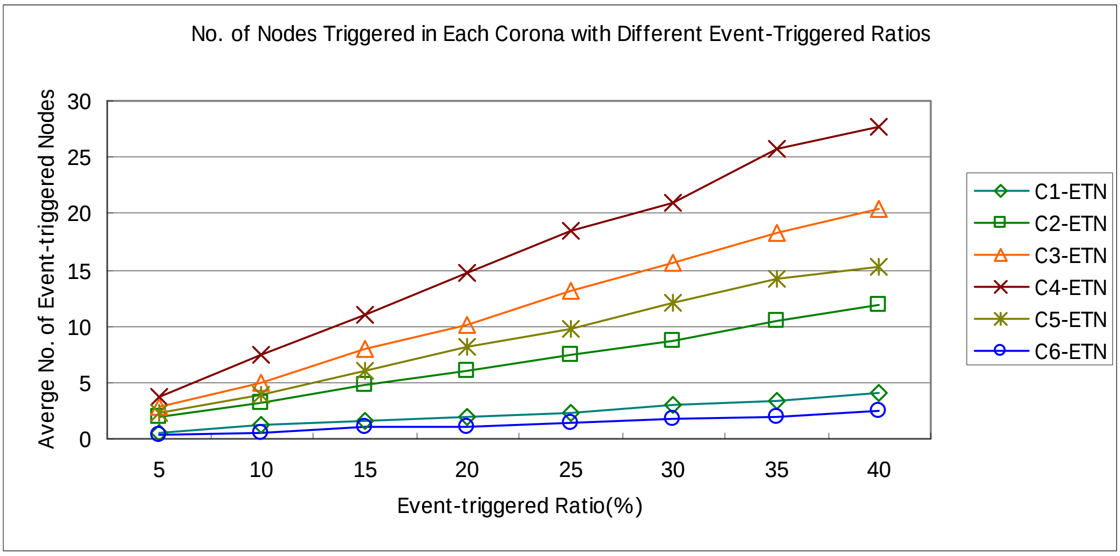


Fig. 15 200 sensors are randomly distributed to a 2000×2000 m^2 sensing field with different event-triggered ratios ranging from 5% to 40%

5.2 Nodes' Loading

In the second experiment, like that in the first portion of experiment 1, 49 nodes were randomly distributed to a sensing field of 1000×1000 m^2 and its base station is placed at the center of the field. We analyze loading of nodes on a routing path from an event-triggered node to the base station given different ratios of event-triggered nodes ranging from 5% (2.5 nodes in average) to 40% (20 nodes in average). In Fig. 16, the average ETN represents average number of nodes triggered by events in the field, and the average loading denotes the average number of nodes that a node along the routing path should relay packets for. The Max and Min Loadings are the maximum and minimum numbers of nodes that a node should relay packets for.

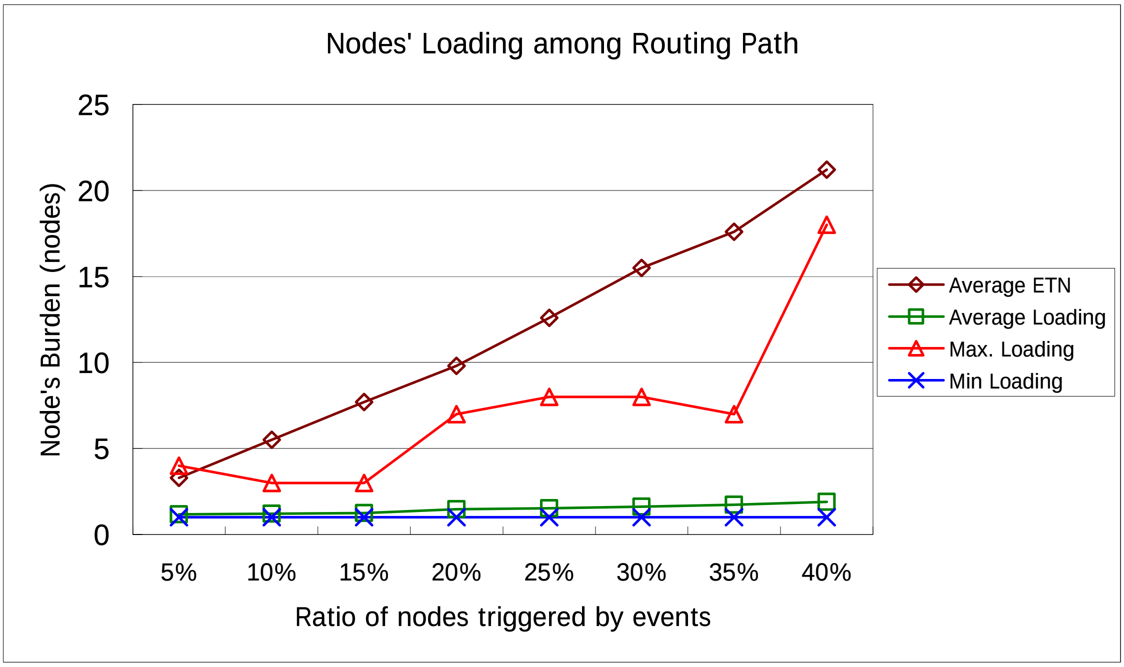


Fig. 16 Nodes' loading along routing paths from event-trigger nodes to base station

When ratio of nodes triggered by events increases from 5% to 40%, burden of a node rises from 1.167 to 1.906, which is below 2, representing that when ratio is up to 40% from 5%, the whole system can still keep working for 63.32% ($=\frac{1.906-1.167}{1.167}$) of lifetime. However, we can also find that the maximum loading of a node is not linearly increased with the event-triggered ratio, because sometimes the event-triggered nodes might be close one another, giving a routing path very heavy burden. Here, 40% also indicates that in the worst case nodes on the 20 event-triggered nodes' routing paths might exhaust their energy at very high consumption rates. So, if some other nodes can come to help them to relay packets, the system will effectively extend its lifetime.

5.3 System Lifetime

To verify our system, in the third experiment, we compare the system lifetime of a network which is a system with AODV routing and RASLM, called AODV-RASLM.

This experiment is simulated by using NS-2 [24]. We assume that all 50 nodes are randomly distributed to a 1000×1000 m^2 field and node 0 acting as the base station is placed at the center of the filed with coordinates (500, 500) (see Fig. 17). We assume that the number of nodes given is sufficiently and strongly connect the whole network, and each sensor knows its position (through GPS). The moving speed and maximum moving distance of a sensor are set to 25m/s, and 1000m, respectively.

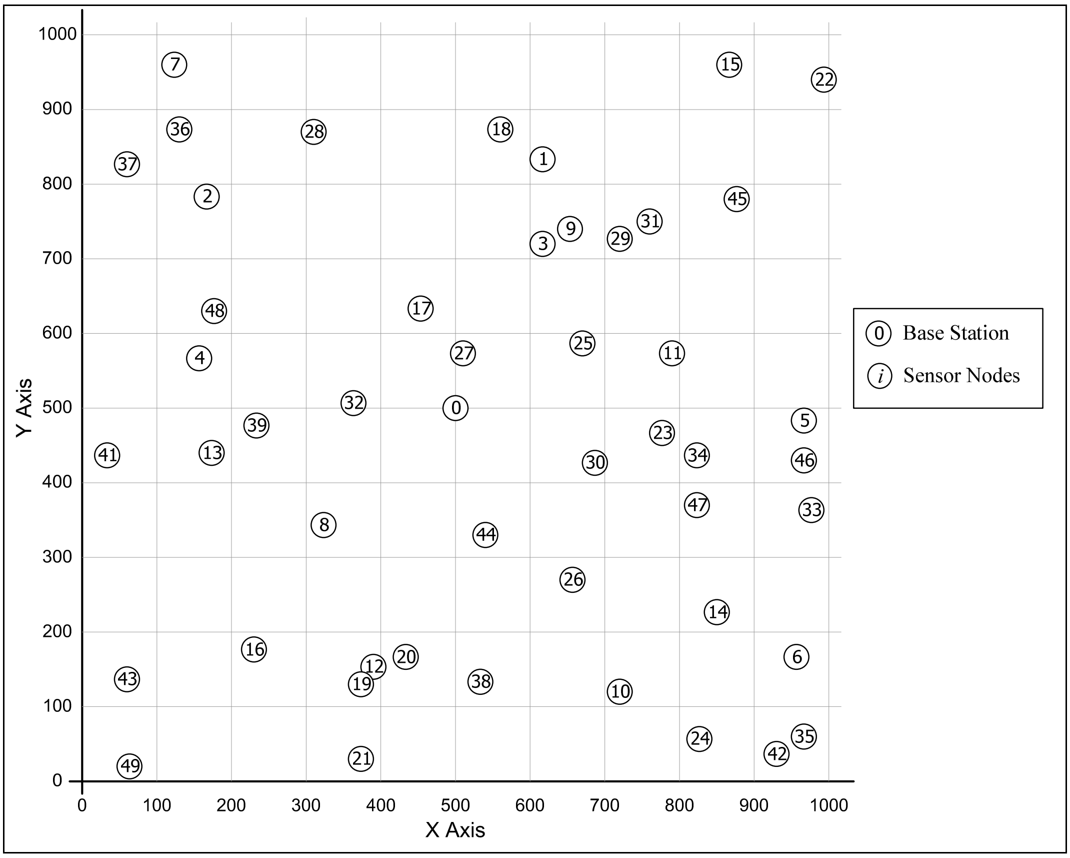


Fig. 17 Topology of an event driven wireless sensor network

Transmission and receiving ranges of sensors including ordinary nodes and base station are set to 250m. Each sensor's initial energy is 5 joules, and the energy consumptions for transmission and receiving are 1.2 μJ and 0.6 μJ, respectively. Packet size is 512bytes and a node's data packet generating time interval is 30 seconds. The AODV's active route timeout parameter is set to 40 seconds to ensure that a routing path will not be switched to another path within a packet generating time interval. The threshold δ of a node's workload is set to 1, the ratio of nodes triggered by events is set to the values ranging from 5% to 50%, and the static TTL of a RAP packet is set to the values from 1 to 3, and the maximum value of dynamic TTL is set to 4. To simplify this experiment, without losing its generality, we assume that ETN-nodes are triggered at the start time of the whole system, and they will keep sending data packets to the base station until exhausting their energy. The system lifetime is measured when the first node dies. Fig. 18 shows the experimental results in which each value is the average of 100-fold result values.

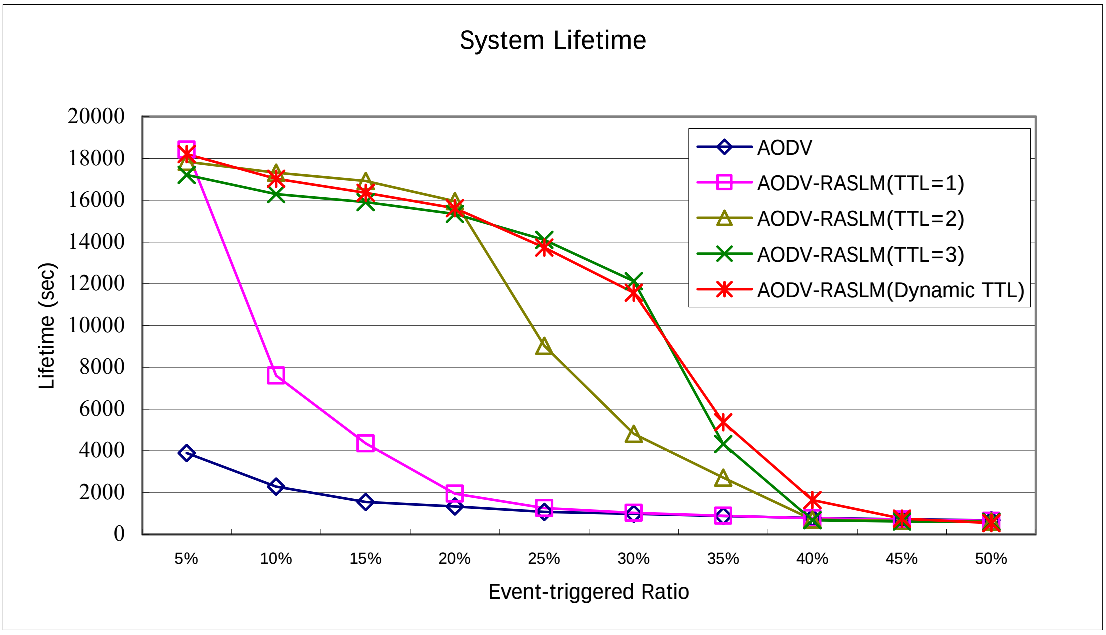


Fig. 18 System lifetime by comparing AODV and AODV-RASLM with different TTL values

From Fig. 18, we can see that the RASLM can effectively improve the lifetime of pure AODV routing, no matter what value TTL is. Generally, the AODV-RASLM with TTL=2 and TTL=3 outperform that with TTL=1, because broadcasting a RAP with bigger TTL value can call for many more assistant nodes. But, when the event-triggered ratio is 5%, the AODV-RASLM with TTL=1 performs better than that TTL=2. The reason is when the event-trigger ratio is low, few assistant nodes are required. TTL=1 is able to call for sufficient assistant nodes, and TTL=2 and 3 will call for unnecessary assistant nodes and force the event-triggered nodes, their relaying nodes and neighbors to die first due to processing the RAP-ACKs, and rebroadcasting RAP packets and/or relaying RAP-ACKs. The lifetime of AODV-RASLM with TTL=1 declines very quickly when the event-triggered ratio rises to 10% and more. This is because a higher event-triggered ratio implies that many more assistant nodes are required. But TTL=1 calls for insufficient assistant nodes. Therefore, we can conclude that RASLM with TTL=1 is only suitable for a sensing field whose event-triggered ratio is lower. In this example, it is lower than 5%.

When event-triggered ratio ≤20%, system lifetime of the AODV-RASLM with TTL=2 and TTL=3 not only is longer than that with TTL=1, but also declines slightly. When event-triggered ratio ≥ 25%, implying high burden of a relaying node, TTL=3 performs better than TTL=2 because TTL=3 can call for many more assistant nodes to share relaying burden, and TTL=2 starts calling for insufficient assistant nodes. Now, we can conclude that RASLM mechanism is able to effectively prolong system lifetime of an AODV routing. From event-triggered ratio=5% to 25%, there is a race condition. Therefore, we can predict that if TTL=4 were included, when event-triggered ratio is low, its lifetime will not be better than those when TTL=2 and TTL=3. But at some higher ratio point, e.g., 30% or 35%, it will outperform those when TTL=2 and TTL=3. The result of AODV-RASLM with dynamic TTL shows that it can perform well as static TTL=1 when the event-triggered ratio is low, and when the event-triggered ratio rise up, some heavy burden nodes can call for sufficient nodes to come for help if they needed by extending the TTL value.

We extend the sensing field to 2000×2000 m^2, and increase the total number of sensors to 199 to keep its node density the same as that when 49 sensors are distributed to a 1000×1000 m^2 field. The initial energy and, energy consumption for transmitting and receiving a packet are all the same as before. We evaluate the system lifetime by using static TTL value from 1 to 5, and the maximum TTL value of dynamic TTL is set to 8, to call for many more assistant nodes. The experimental results are shown in Fig. 19, from which we can find that the system lifetime of pure AODV routing is quickly decreased to 1200 seconds from 4000 seconds compared with that shown in Fig. 18. The reason is the length of a routing path increases with the extension of the sensing field.

The trends of experimental results shown in Fig. 19 are similar to those illustrated in Fig. 18. However in Fig. 19, the lifetime of AODV-RASLM with TTL=4 keeps slightly downgraded when the event-triggered ratio rises to 30%. Once the event-triggered ratio goes over 35%, no matter what value the TTL is, the system lifetime could not be further improved. The reason is that most non-event-triggered sensors move to help their neighbors. They were not able to accept other RAP requests. This is also true in Fig. 18 when the event-triggered ratio goes beyond 35%. We call the ratio value life point. Life point of the 2000×2000 m^2 sensing field is a little smaller at 35% because its routing paths are longer, indicating that when a path consists of more relaying nodes, number of non-active nodes will decrease, resulting in the fact that insufficient nodes can move to help others. Therefore, the AODV-RASLM with TTL=5 does not prolong the system lifetime compared to that of TTL=4, particularly when the event-triggered ratio is high, since number of nodes TTL=5 can call for is not significantly higher than that when TTL=4, and a non-active node on receiving RAP, however, in our scheme, will pick the nearest one as the selected candidate. The AODV-RASLM with dynamic TTL can improve the system lifetime furthermore when the event-triggered ratio is 35%, and it also can save energy when event-trigger ratio is low because of avoiding from unnecessary RAPs broadcasting.

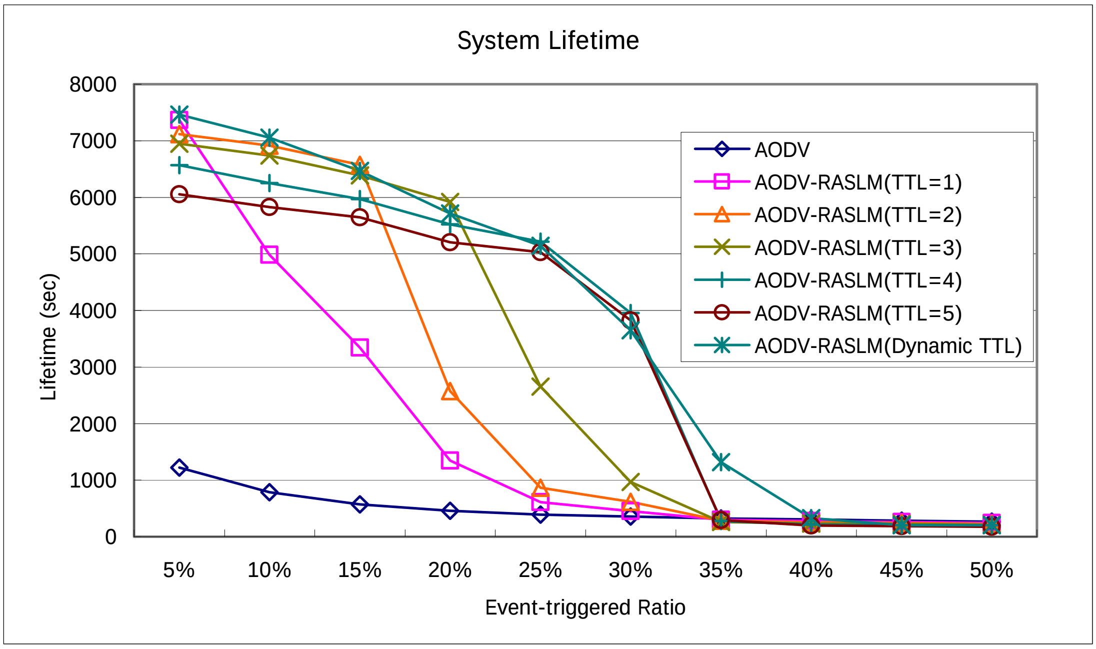
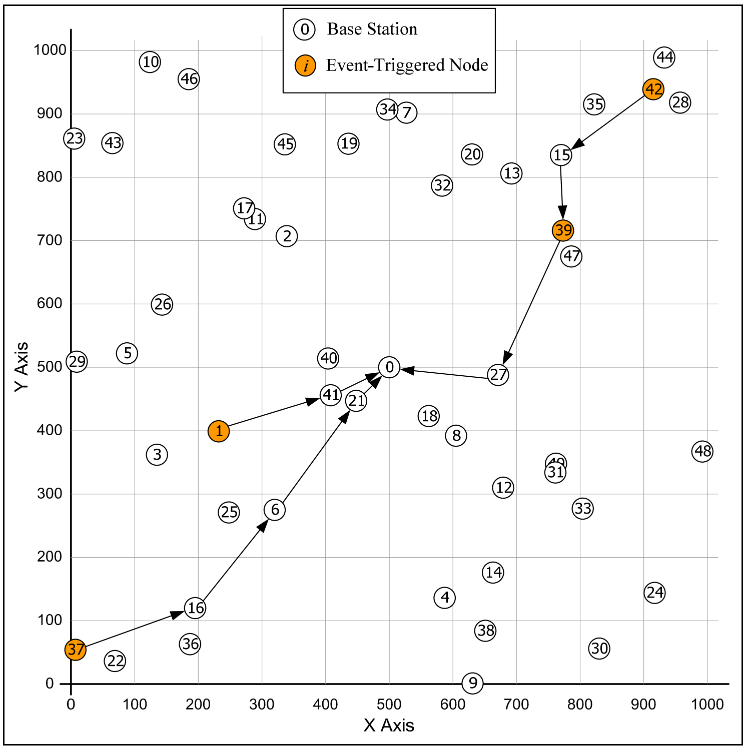


Fig. 19 System lifetime by comparing AODV and AODV-RASLM with different TTL values in the sensing field of 2000×2000 m^2 to which 199 sensors are randomly distributed

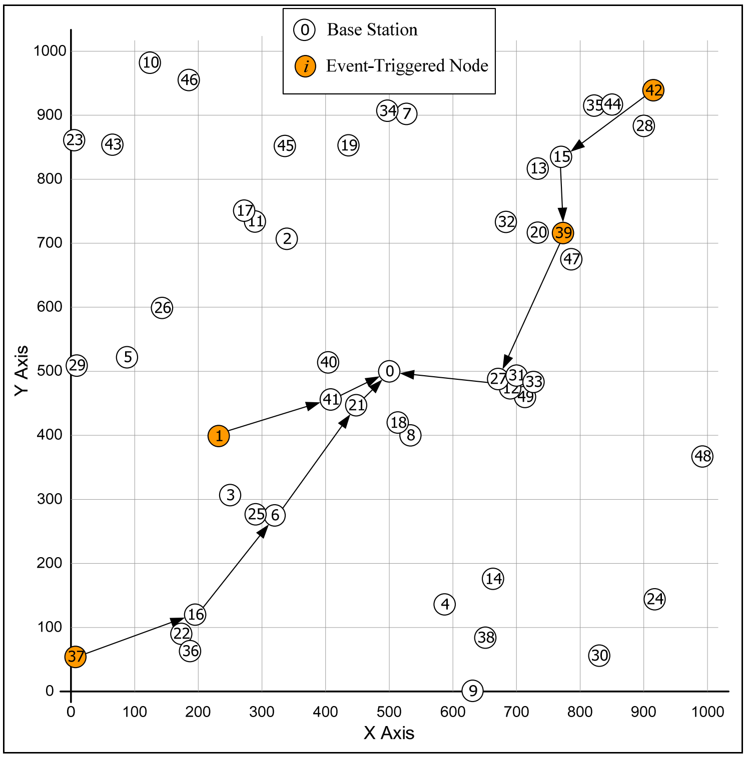
5.4 Sensor Nodes Distributed in RASLM System

The fourth experiment was also performed in the 1000×1000 m^2 sensing field on which 49 nodes are randomly distributed and event-triggered ratios given range between 5% and 40%. We draw network topologies for the experimental results and compare positions of all assistant nodes before and after they moved to help RAP senders.

In Fig. 20a, there are four event-triggered nodes (i.e., event-triggered ratio=5%) in the sensing field, e.g., node 1, 37, 39, and 42. The four nodes established four routing paths to the base station when the system is started up. Node 0 is denoted as the base station. The topology generated after active nodes sent out RAPs with TTL=1, and non-active nodes moved to the EAZs they chose is shown in Fig. 20b. We can see that the RASLM scheme can call for sufficient assistant nodes to help nodes along active routing paths. However, in Fig. 20b, we can also find that even the TTL=1, some senders, e.g., nodes 27 and 16, have called for too many assistant nodes. This causes unnecessary energy consumption for node movement.



(a) before



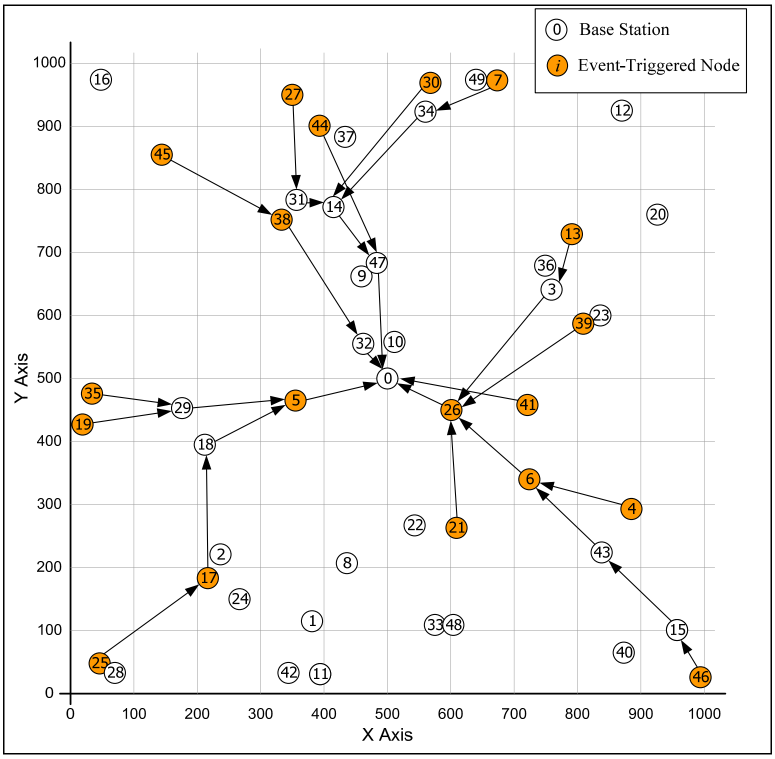
(b) after

Fig. 20 The network topology and position change of assistant nodes before and after the experiments. The experiment is performed in the 1000×1000 m^2 sensing field in which 49 nodes are randomly distributed and event-triggered ratio is 5%

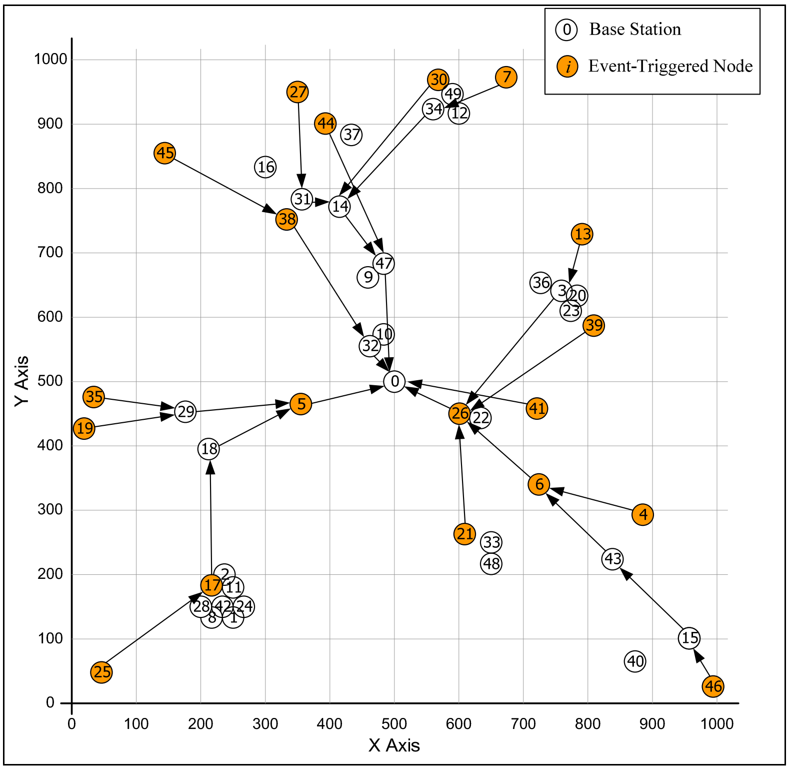
Next, we raised event-triggered ratio to 40% and redid the experiment. Fig. 21a and Fig. 21b show initial and result topologies, respectively. In this experiment, 19 nodes are triggered by events, and these nodes together with 10 non-triggered nodes form 19 partially overlapped active routing paths. Also, 20 nodes go to their nearby EAZs, acting as assistant nodes. However, by observing the position change from Fig. 21a to Fig. 21b, we can find that although some active nodes and event-triggered nodes can mitigate their relaying burden with assistance of assistant nodes, some heavy-burden nodes, e.g., node 26, did not call for sufficient assistant nodes. This node has to transmit data packets it generates and relay packets issued by six nodes, i.e., nodes 4, 6, 13, 21, 39, and 46. Energy consumption rate of node 26 can be calculated by using formula (1):

$$\frac{Data\_Length \cdot (u \cdot e_1 + j \cdot e_2)}{T_m} = \frac{512 \times 8 \times (7 \times 1.2 \times 10^{-6} + 7 \times 0.6 \times 10^{-6})}{30} = 1720.32 \times 10^{-6} \, Joule / \sec$$

But by collaborating with assistant node 22, its energy consumption rate can be reduced to one half averagely. However, still node 26 will die very soon compared with life time of other nodes. In addition, node 17 only relays packets for node 25, and transmits packets it generates to base station. But seven assistant nodes come to help it. This is because assistant nodes chose the nearest EAZ as the one it goes to help. Distance is the only consideration, ignoring the RAP-senders' energy consumption rates. This problem can be solved by adding RAP-sender's energy consumption rate into the RAP, and modifying the candidate list handling algorithm for assistant nodes.



(a) before



(b) after

Fig. 21 The network topology and position change of assistant nodes before and after the experiment. The experiment is performed in the 1000×1000 m^2 sensing field in which 49 nodes are randomly distributed and event-triggered ratio is 40%

5.5 System Lifetime on Different Ratios of Nodes Failure

In the fifth experiment, we measure the system lifetime on different ratios of nodes failure to further validate the effectiveness of RASLM. In Fig. 22, we can see the system lifetime increases when the measured criterion changed from the first node dies to different ratios of node failure by using pure AODV routing protocol. When event-triggered ratio (ETR) is 10%, the system lifetime is only marked at ratios of 10%, 20% and 30%, because only a small portion of nodes actually participate in packet generation and relaying packets in the whole system. Other nodes did not act as assistant nodes or active nodes. Hence, it is not possible to have more than 30% of failed nodes on ETR=10%. We can also see that system lifetime increases as ratios of failed nodes increase.

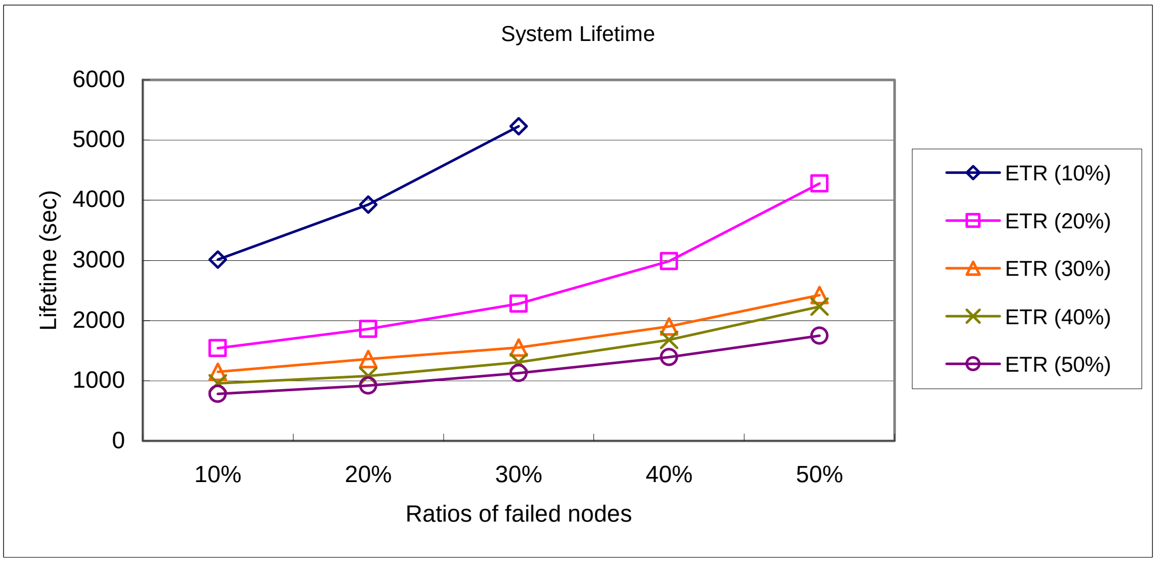


Fig. 22 System lifetime measured at different ratios of nodes failed with different event-triggered ratio (ETR) by using AODV routing protocol

Fig. 23 compared system lifetime measured on different ratios of failed nodes from 10% to 50% when the event-triggered ratios are 20% and 30% and routing protocols involved are AODV, AODV-RASLM, DSDV and DSDV-RASLM. We can find that the RASLM actually improves AODV's lifetime and DSDV's lifetime. But, the system lifetimes of AODV-RASLM and DSDV-RASLM are not significantly improved when ratios of failed nodes increase from 10% to 50%. This is because the RASLM scheme not only calls for help, but also evenly forces the RAP-sender to share the burden of relaying packets based on the residual energy levels. That means that RASLM can effectively balance the energy consumption along the routing path, even there exist some heavy burden nodes.

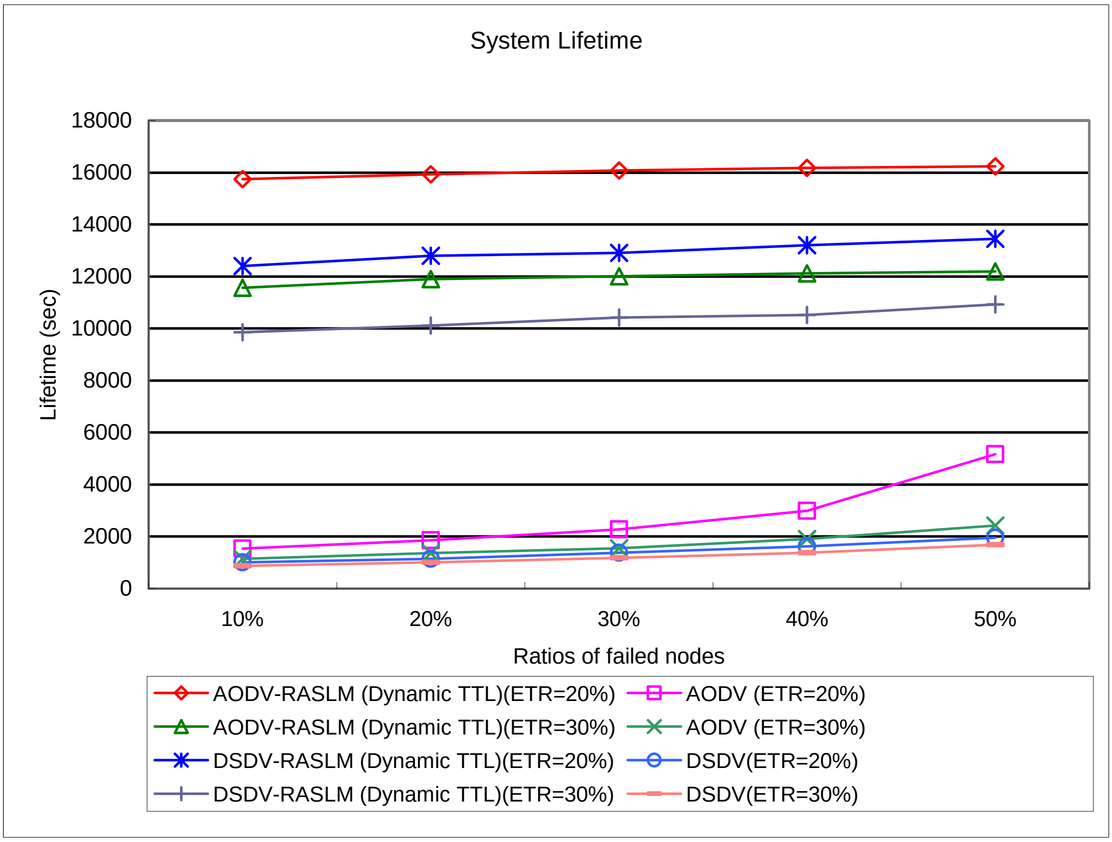


Fig. 23 System lifetimes against different ratios of failed nodes when AODV, AODV-RASLM (Dynamic TTL), DSDV and DSDV-RASLM (Dynamic TTL) are involved on event-triggered ratio=20% and 30%

6. Conclusion and Future Work

In the paper, we analyze the energy hole problem which may occur not only in message-evenly-generated sensing environment, but also in event-driven wireless sensor networks. This problem may harm to particular environments, like the scene of fire, or other environments that start to collect data once their sensors are triggered. Therefore, we propose a routing support scheme named RASLM by using mobile nodes to help nodes on an active routing path not only to prevent these nodes from exhausting energy quickly, but also stabilize the routing path. The latter can ensure that the sensed data to be sent to base station safely and smoothly. The algorithms for an assistant node and a RAP requesting node are proposed. The experimental results show that the RASLM can effectively prolong system lifetime of an event-driven wireless sensor network when number of nodes triggered increases, and the analysis of position change shows why the RASLM can improve the system lifetime when the event-triggered ratio is low, and also, why it can not improve the system life time when the event-triggered ratio is high.

Our future work includes deriving and analyzing the cost and reliability models of RASLM, and integrating the RASLM with different routing schemes. Furthermore, we will try to integrate MAC layer protocols, like sleeping mode scheduling, with the RASLM to further improve its system efficiency so as to save as much energy as possible, especially when event-triggered ratio of a sensor network is high.

7. References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “ A survey on sensor networks,” IEEE Communications Magazine, vol. 40, issue 9, pp. 102-114, Aug. 2002.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An Application-Specific Protocol Architecture for Wireless Microsensor Networks,” IEEE Transactions on Wireless Communications, vol. 1, issue 4, pp. 660-670, Oct. 2002.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” In Proc. of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM’00), pp.56-67, Aug. 2000.

[4] X. Wu, G. Chen, and S.K. Das, “On the Energy Hole Problem of Nonuniform Node Distribution in Wireless Sensor Networks,” in Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems, pp. 180-187, Oct. 2006.

[5] J. Luo and J. P. Hubaux, “Joint mobility and routing for lifetime elongation in wireless sensor networks,” in Proc. of IEEE Conference on Computer Communications, vol. 3, pp. 1735-1746, Mar. 2005.

[6] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, “TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks,” ACM Wireless Networks, vol. 11, issue 1-2, pp. 161-175, Jan. 2005.

[7] X. Wu and G. Chen, “Dual-Sink: Using Mobile and Static Sinks for Lifetime Improvement in Wireless Sensor Networks,” in Proc. of International Conference on Computer Communications and Networks, pp. 1297-1302, Aug. 2007.

[8] M. Zhao, Z. G. Chen, X. H. Deng, L. M. Zhang, A. F. Liu, and G. S. Huang, “MAEC: A Movement-Assisted Energy Conserving Method in Event Driven Wireless Sensor Networks,” In Proc. of International Conference on Wireless Communications, Networking and Mobile Computing, vol. 2, pp. 915-920, Sept, 2005.

[9] B. Wang, D. Xie, C. Chen, C. Chen, J. Ma, S. Cheng, "Employing Mobile Sink in Event-Driven Wireless Sensor Networks," in Proc. of IEEE International Conference on Vehicular Technology, pp. 188-192, May 2008.

[10] S. Yang, J. Wu and F. Dai, "Localized Movement-Assisted Sensor Deployment in Wireless Sensor Networks," in Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems, pp. 753-758, Oct. 2006.

[11] S. Chellappan, Wenjun Gu, X. Bai, D. Xuan, B. Ma, and K. Zhang, "Deploying Wireless Sensor Networks under Limited Mobility Constraints," IEEE Transactions on Mobile Computing, vol. 6, no. 10, pp. 1142-1157, Oct. 2007.

[12] Yang and M. Cardei, "Movement-Assisted Sensor Redeployment Scheme for Network Lifetime Increase," in Proc. of the ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, pp. 13-20, Oct. 2007.

[13] F.Y. Leu, and G.C. Li, "A Scalable Sensor Network Using a Polar Coordinate System," Signal Processing, vol. 87, issue 12, pp. 2978-2990, Dec. 2007.

[14] Y. Bi, J. Niu, L. Sun, W. Huangfu, and Y. Sun, "Moving Schemes for Mobile Sinks in Wireless Sensor Networks," in Proc. of International Conference on Performance, Computing, and Communications (IPCCC 2007), pp. 101-108, Apr. 2007.

[15] Y.C. Wang, C.C. Hu, and Y.C. Tseng, "Efficient Placement and Dispatch of Sensors in a Wireless Sensor Network," IEEE Transactions on Mobile Computing, vol. 7, issue 2, Feb. 2008.

[16] J. Li and P. Mohapatra, "An Analytical Model For The Energy Hole Problem In Many-To-One Sensor Networks," in Proc. of the IEEE 62nd Vehicular Technology Conference (VTC-2005), Sept. 2005.

[17] F. Y. Leu, G. C. Li and W. C. Wu, "An Autonomous Energy-Aware Routing Scheme: a Supplementary Routing Approach for Path-Preserving Wireless Sensor Networks," in Proc. of the IFIP Annual Mediterranean Ad Hoc Networking Workshop, pp. 49-60. June 2008.

[18] X. Wu and G. Chen, "Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution," IEEE Transaction on Parallel and Dirtributed

Systems, vol. 19, no. 5, pp. 710-720, May 2008.

[19] R. Jurdak, P. Baldi, C.V. Lopes, "Adaptive Low Power Listening for Wireless Sensor Networks," IEEE Transactions on Mobile Computing, vol. 6, issue 8, pp. 988-1004, Aug. 2008.

[20] M. Tacca, P. Monti, A. Fumagalli, "Cooperative and Reliable ARQ Protocols for Energy Harvesting Wireless Sensor Nodes," IEEE Transactions on Wireless Communications, vol. 6, issue 7, pp. 2519-2529, July 2007.

[21] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility limited flip-based sensor networks deployment," IEEE Transactions on Parallel and Distributed Systems, vol. 18, issue 2, pp. 199-211, Feb. 2007.

[22] D. Lymberopoulos and A. Savvides, "XYZ: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Applications," in Proc. of International Symposium on Information Processing in Sensor Networks, pp. 449-454, Apr. 2005.

[23] http://www.darpa.mil/sto/smallunitops/shm/index.htm

[24] http://www.isi.edu/nsnam/ns/