私立東海大學

資訊工程研究所

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移動助理節點之控制架構

Event-Driven Mobile Assistant Control Scheme of Wireless Sensor Networks

研究生：黃宏瑋

中華民國九十九年八月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 黃　宏　瑋 所提之論文

事件觸發式無線感測網路的移動助理節點之控制協定

經本委員會審查，符合碩士學位論文標準。

學位考試委員會
召　集　人 林宜隆 簽章

委　　　員

指 導 教 授 簽章

中華民國 99 年 7 月 14 日

摘要

在無線感測器網路中，能量是一個重要的資源，必須節省使用。但是現有很多路徑佈置節點方式都利用 hop-by-hop 方式傳送封包，造成節點會沿著路徑不但要收集資料和傳送資料到基地台，而且要幫助鄰居節點傳送資料到基地台。這樣會造成能量消耗的不平衡問題。在事件觸發的環境中，假使節點更新資料頻繁，會導致基地台附近的節點能量消耗快速，如何降低能量消耗是一個重要的課題，在這篇論文中，我們提出節點呼叫助理節點的架構，命名為事件觸發式無線感測網路的移動助理節點之控制架構，當有一個節點 N 能量消耗快完畢，它會呼叫一個移動式節點來幫助它可以繼續感測環境變化和傳送資料給上游的節點。這樣可以避免能量黑洞的問題，實驗結果顯示可以有效延伸節點的使用率和網路的存活時間。

關鍵字：事件觸發無線感測網路、移動節點、節點密度、網路存活時間、控制架構

Abstract

In wireless sensor networks (WSNs), energy is one of the most important resources that should be economically used so as to prolong a powered-by-battery WSN's lifetime. Currently most routing approaches deployed by WSNs are hop-by-hop relay schemes, causing sensors along a routing path should not only collect its environmental data, and then send the data to base station, but also relay data received from its neighbors toward the base station. This will result in an unbalanced energy consumption problem for WSNs. In an event-driven environment, if events frequently occur in a node's upstream area, the node's energy will consume very quickly, particular for those nodes near the base station. Hence, how to prolong network lifetime is one of the most important issues in WSN's research topics. In this paper, we propose a node's call-for-assistant scheme, named An Event-Driven Mobile Assistant Control Scheme (EDMAC for short), with which when a node N is going to exhaust its energy, it calls for mobile nodes to help it to relay packets so the tasks of sensing environment or environment change and relay packets for upstream nodes can be proceeded. This can avoid energy-hole problem. Experimental results show that this approach can effectively prolong node utilization rate and network lifetime.

Keywords: event-driven WSN, mobile node, node density, network lifetime, control scheme

誌謝

得以順利完成研究與論文，首先感謝我的母親與家人，你們的支持是我最大的動力，感謝你們多年來的摘培與支持，讓我順利完成學業。所有學識上的研究成果榮耀，全都獻給你們。

誠摯的感謝指導教授呂芳懌博士，老師悉心的指導使我得以接觸無線感測網路，在研究上極所能的指導，引領正確的研究方向，在為人處世之道上，時時苦語建言，對學生的研究生活與成長有莫大的影響。在此，向老師至上最大的深切感謝。

此外，承蒙口試委員林宜隆教授、蕭瑞祥教授、段裘慶教授及黃其泮教授對於本論文不吝指教，並給予學生我許多寶貴意見，能獲得你們的肯定，是莫大殊榮，在此制上最深謝意。

最後，感謝在研究所的日子裡，實驗室裡共同的生活點滴，學術上的討論，並感謝學長、學姊及同學梗延、國基、炫又、佩勳、小夫等同學的幫忙，使我能完成論文，最後感謝好姐妹宛玲、采蓉、小戴和麻吉益賢、怡君在我面臨研究困頓時幫我解惑，並給予我諸多的關懷與鼓勵，由衷的感謝你們。

最後，謹以此文現給我的摯愛的母親與家人

List of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Recently, wireless sensor network (WSN) applications, such as monitoring climate and disasters and observing environmental ecology, have been extensively developed. A WSN often has a large amount of sensors distributed to the sensing field under surveillance. Many researchers have published their achievements of WSN studies [1][2][3], among which energy consumption is one of the most important issues in prolonging a WSN's lifetime since sensor nodes are often powered by battery.

In a WSN, there typically exist one or a few base stations that collect information from sensors. Each sensor behaves as a data originator as well as a forwarder to respectively generate packets and transmit packets toward the base station or one of the base stations. Due to many-to-one characteristic, some nodes consume energy very quickly, particularly for those close to the base station since they very usual relay many more packets than those nodes far away from the base station, consequently resulting an energy hole around the base station [1]. This often dramatically reduces a WSN's lifetime.

In order to reduce energy consumption for WSNs, Chaing and Byrd [1] controlled and changed sensor node states including active, sleep and discovery to control node density. When nodes do not relay packets, they enter sleep state and awake in a predefined time period. If a node's neighbor node density is lower than the default value, it can enter active state to increase node density. However, controlling and changing node states conduct a complicated computation. Wei and Chan [4] and Sivavakeesar et al. [5] solved the energy conservation problem by using distributed clustering methods. Method proposed by [5] is a (p_{ax}, t_{ax}, d_{ax})-model, where p_{ax} is the probability that a mobile node *a* within x^{th} *virtual cluster* with a distance d_{ax}

from the center of the cluster *x*, and stays inside the cluster for a given time period t_{ax}. However, this model must compute value for p, d and t, which increases the system computation complexity.

In a WSN, a node's death very often is due to relaying too many packets for others. Our opinion is that when a node is going to die, if other nodes can come to help the node to relay packets (may also sense the environment), then lifetime of the WSN can be effectively prolonged. Routing assistant scheme with localized movement (RASLM) [6] also calls for mobile nodes to help packet forwarding along an active routing path. This can effectively avoid a node from relaying too many packets to mitigate the energy hole phenomena. But when many events occur at the same time, many mobiles nodes are required. This may reversely shorten a network's lifetime.

Therefore, in this study, we propose a mobile assistant protocol, called An Event-Driven Mobile Assistant Control Scheme (EDMAC for short), with which when a static node is dying, it calls for assistant nodes to relay packets for it in order to prolong an event-driven WSN's lifetime. In other words, static and mobile nodes coexist in the WSN. The static nodes take charges of sensing environmental change and relay packets. Packets are generated only when events occur in the sensing field. The mobile nodes do not sense the environment. They only relay packets. When it comes to help a dying static node, it also senses the environment.

The rest of the paper is organized as follows. Chapter 2 introduces background and related work of this study. Chapter 3 describes our system architecture and proposed algorithms. Experiments and their discussions are presented in chapter 4. Chapter 5 concludes this paper and outlines our future research.

Chapter 2. Background and Related work

Many methods have been proposed to reduce unnecessary and unbalanced energy consumption for sensors. Basically, they can be classified into several types, including density control [1][7][8], node mobility [6][9][10][11], overhearing avoidance [1][12][13][14], energy hole avoidance [9][15][16], energy-efficient design[6] [17][18] [19] and control protocol [20][21][22].

2.1 Density Control

Chiang and Byrd [1] proposed a system call Neighborhood-Aware Density Control (NADC for short), and claimed that controlling node density is helpful in reducing unnecessary energy consumption for nodes since high node density implies short node distance, which can truly lower data relaying energy. GAF [7] and OGDC [8] used location information to achieve uniform density over a network. In NADC, nodes dynamically adapt their participation in the multi-hop network topology. When there are data packets waiting to be relayed, the node enters its active state to relay the packets. When no packets can be relayed, nodes enter sleeping mode. A sleeping node wakes up in a fixed time interval and observes its neighborhood density. Such can reduce unnecessary energy consumption and prolong network lifetime.

2.2 Node Mobility

A lot of studies have tried to solve unbalanced energy consumption problem for WSNs with mobile nodes. Yang and Cardei [10] 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 when necessary to lower an inner node’s data relaying rate and balance energy consumption rate among all nodes. Leu et al. [11] employed a polar coordinate system to identify a mobile node’s position and route network packets. However, they all focused on evenly-message-generating WSNs rather than event-driven WSNs. Leu et al. [6] proposed a routing approach that deploys mobile sensors to help relaying packets for nodes along an active routing path to prevent the nodes from dying much earlier than others. This scheme can also stabilize the routing path, which ensures the sensed data to be sent to the base station safely and smoothly. Marta and Cardei [9] increased a wireless sensor network lifetime by employing mobile sinks that change location when the nearby sensors' energy becomes low. In deciding a new location, a sink searches those zones with richer sensor energy.

2.3 Overhearing Avoidance

Overhearing consumes unexpected energy. In wireless networks, packets are very usual sent by broadcasting. Various approaches developed for reducing the overhearing energy consumption on the medium-access control (MAC) layer have been proposed [1][12][13].

The rationale behind is to save energy through limiting the impact of idle listening and traffic overhearing. TMAC [12] and S-MAC [13] used RTS-CTS mechanism to avoid overhearing. Nodes in NADC [1] as stated above observing their neighborhoods and adapting their participation in the multi-hop network topology dynamically is another example.

A-MAC [14] combined the strengths of TDMA and CSMA to achieve the goal of low power transmissions for long-term surveillance and monitoring applications, where sensor nodes are typically vigilant and inactive most of the time until events are detected. It also employed an advertisement mechanism to eliminate collisions. Both can reduce overhearing.

2.4 Energy Hole Avoidance

In order to improve energy hole problem, [9] as stated above employed sink mobility, with which a sink changes its location when its nearby sensors' energy is low. In this way, the sensors located near base station change over time, thus diminishing the energy imbalance around the sink. [16] used non-uniform node distribution to provide nearly balanced energy depletion in the network. With this approach, number of nodes grows in geometric progression from outer coronas to inner ones. It devises a q-switch shortest path routing algorithm coupled with the proposed non-uniform node distribution strategy, which effectively switches the data flow among its corresponding q or (q-1) next-hop forwarding candidates to balance energy dissipation, where q is the common ratio of the above mentioned geometric progression. Wang, Srinivasan and Chua [15] used a mobile node as the sink, claimed that this can enhance the network lifetime by a factor of nearly four.

2.5 Energy-Efficient Design

Nodes in different WSNs are often organized into specific structures, such as hierarchical structures in which clustering schemes are proposed for the purpose of distributing energy depletion among all nodes. In LEACH [17], 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. In HEED [6], communication costs and remaining energy of nodes are the key factors in selecting cluster heads. [18] proposed two energy levels for nodes. Nodes that closer to the sink are given higher levels of initial energy and those further away from the sink are given lower levels of initial energy. [19] proposed a new initial energy assignment strategy with which the total initial energy may be limited or unlimited, and the levels of initial energy assigned to a node is based on the energy consumption per data collection round, so all the nodes run out of energy and die at the same time.

2.6 Control protocol

Control protocol is one of the most important techniques in wireless sensor networks to reduce energy consumption and prolong network lifetime. Hong, Choi and Kim [20] proposed the topology control protocol which focuses on determining best-effort transmission power of each sensor node to maximize the lifetime of sensor networks without violating the connectivity of them. The authors claimed that their proposed protocol increases the lifetime of network and guarantees the network connectivity among sensor nodes. Enigo and Ramachandran [21] designed a congestion control mechanism at the source node based on the sum of the node weights at each node. Each node adds its current weight to that it received from a downstream node, and passes this information toward the upstream node. After, the source node will receive the sum of all weights from its downstream nodes. With the sum of weights the source node controls data rates. Huang, Ku, Kung [22] proposed a novel efficient communication topology control protocol, called quorum-based load-sharing control protocol, which chooses appropriate communication nodes, adjusts the service loads of critical nodes and performs adaptive sleep management. This protocol is suitable for harsh environments without a central control server calculating the locations of sensors. It uses the factor of the remaining power to build the system topology.

Chapter 3. The Proposed Scheme

In this study, the sensing field as shown in Figure 1 is partitioned into many concentric circles, called coronas, which from the innermost to the outermost are numbered corona 0, corona 1, … , corona n-1, where n is number of coronas in the field. Radius of corona i is $i \cdot r$, where r is the radius of corona 0, also the communication radius of a sensor node. The center point of the field, denoted by C, is the location where the base station is placed. Nodes in the field are organized into groups. Those in corona i form group i, i=0, 1, 2, … , N-1. Some nodes in a corona are static nodes which will not change their locations during their lifetime once they are individually allocated to the field, whereas the remaining nodes are mobile nodes which can only move in the corona where it is distributed to.

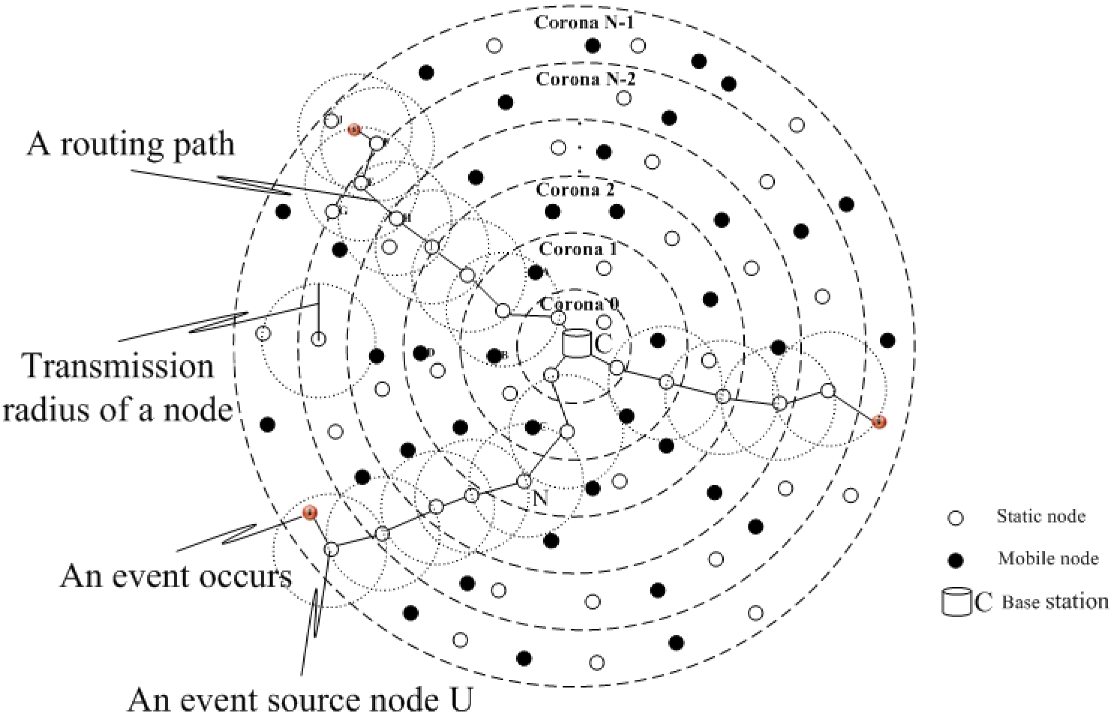


Figure 1 In the event-driven wireless sensor network that we employ, the sensing field which is partitioned into N coronas. Data/event packets are forwarded to a base station hop by hop.

Basically, node density [1][2][23] and energy initially assigned to nodes of a corona [16][19] are the two main factors that determine a network's lifetime. From the two factors, in this study, several different node deployment environments were proposed and tested, and a static node before exhausting its energy will call for a mobile node, called an assistant node, to help relay packets and sense environmental change for it.

Generally, when events occur, an inner node (an inner corona) relays many more packets than an outer node (an outer corona) does. Hence, if we want all nodes of the field die almost at the same time, there are three methods to achieve this. The first is increasing an inner corona’s node density if all nodes in the field are given the same initial energy. The second is, in a uniform node density environment, an inner node is given a higher initial energy level than that given to an outer node. The third is an inner corona is given a higher initial energy level and node density.

They are employed under the assumption that all nodes of a given WSN may die simultaneously.

3.1 Establishing a Routing Path

In this study, the energy aware routing method proposed in [24] is employed. With this method, a source node, before sending data packets to the base station, initiates a route discovery process to create a neighbor list which records all nodes able to directly receive packets from the source node. After the route request (RREQ) packet is broadcasted, a node on receiving this packet checks its residual energy to see whether it has enough energy to relay packets or not. If not, this node discards the packet. Otherwise, it replies a packet telling the source its residual energy level. The source node then chooses the one with the highest energy level and transmission quality as its downstream node, which in turn broadcasts a RREQ and chooses a downstream node. The process repeats until the RREQ arrives at the base station. When base station receives the first RREQ packet, indicating that there is a request for establishing a new route, it transmits a route reply packet along the reverse direction of the path the RREQ traveled through. The routing path is then established. The authors claimed that this method consumes less energy than AODV. That is why it is employed.

3.2 An Effective Assistant Zone

The effective assistant zone (EAZ) of a node, e.g., node *x*, as shown in Figure 2, is defined as the intersection region of communication ranges of node *x*'s immediate upstream nodes, e.g., nodes *w* and *z*, and immediate downstream nodes, e.g., node *y*. Node *x*'s EAZ is the only region that its assistant nodes can communicate with all *x*'s immediate upstream and downstream nodes.

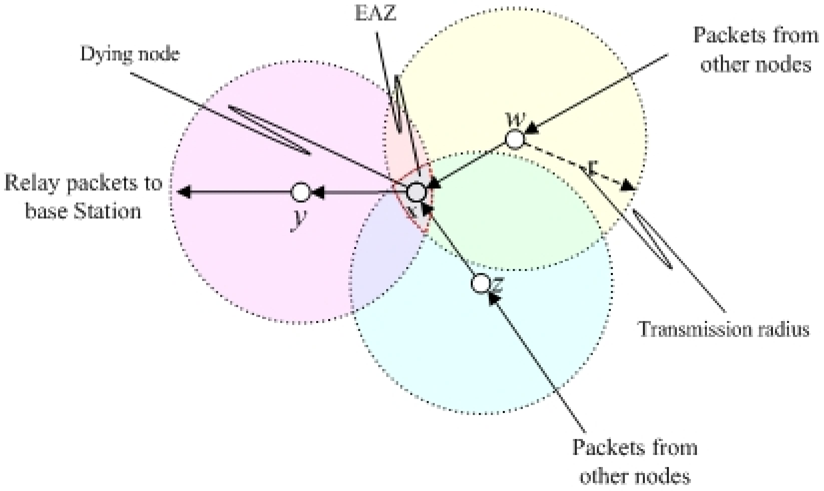


Figure 2 Node *x*'s EAZ is the intersection region of nodes *w*'s, *y*'s and *z*'s communication ranges.

3.3 Packets Involved

In the EDMAC, seven control packets are involved in this study, including request-assistance packet (RAP), RAP-reply packet, call-for-substitution packet, help packet, re-routing packet, path-release packet and dying-notification packet. When a node *N*, which is distributed to a corona k and now on an active routing path, realizes that its energy will soon be exhausted, it initiates a timer λ and broadcasts a RAP packet to request one of the mobile nodes to come to its EAZ.

(1) **A RAP packet**

A RAP packet as shown in Figure 3 consists of four main fields including packet_no.1, RAP-ID, corona ID and source coordinates. Packet_no.1 is used to identify a RAP packet. RAP-ID comprises two sub-fields, source ID and serial-number. The former shows which node issues the packet (e.g., node N), whereas the latter is a number used to discriminate different RAPs. Each time when N broadcasts a new RAP, it increases its serial number by one to ensure the uniqueness of the RAP-ID. The third field is N’s corona ID, e.g., corona-k, so that a receiver can realize whether it should relay the RAP packet or discard it since in this study a node in corona k, no matter a static or a mobile node, only relays RAP packets issued by corona-k’s dying nodes. In other words, a RAP packet issued by a corona-k node can only be delivered in corona k. A node in corona j on receiving a corona-k RAP packet will discard the packet if $j \neq k$. The fourth field, source coordinates, conveys N’s coordinates with which a relay node Q on receiving the RAP packet can realize whether or not the packet has been delivered over a half circle of corona k, i.e.,$(k-\frac{1}{2})r \cdot \pi(=\frac{1}{2}(2(k-\frac{1}{2})r \cdot \pi))$. If yes, the packet will be discarded since the packet is delivered clockwise and counterclockwise simultaneously along corona k.

1 byte	1 byte	1 byte	1 byte	6 bytes
Packet_no.1	RAP-ID		Corona ID	Source coordinates
	Source ID	Serial_No.		

Figure 3 Format of a request-assistance packet (RAP).

(2) **A RAP-reply packet**

A RAP-reply packet which can only be generated by a mobile node to respond to a RAP packet as shown in Figure 4 consists of seven main fields. The first is packet_no.2 indicating this is a RAP-reply packet. The second is RAP-reply-ID which comprises two sub-fields, source ID and serial-number, showing the RAP-ID of the

RAP packet to identify the corresponding assistance request. The third illustrates which mobile node, e.g., node M, sends the RAP-reply packet. The fourth field is M’s current coordinates. The fifth and the sixth fields respectively convey M’s residual energy level E and the energy per unit of distance (EPUD for short), $EPUD=E/l_{NM}$, where l_{NM} is the geographical distance between N and M along the corona where N and M locate, i.e., corona k. By evaluating EPUDs collected from mobile nodes, N can choose a suitable mobile node to help it. We will describe it later. The last field is corona ID, of which the function is the same as that of corona ID field in a RAP packet. A corona-k node, no matter a static or mobile node, only relays RAP-reply packets issued by a corona-k mobile node. The purpose is also to keep the RAP-reply to be delivered along corona k. When N chooses M as an assistant node, it sends a Call-for-substitution packet to M.

1 byte	1 byte	1 byte	1 byte	6 bytes	2 bytes	2 bytes	1 byte
Packet_no.2	RAP-reply-ID		Mobile node ID	Mobile node coordinates	Residual energy level	EPUD value	Corona ID
	Source_ID	Serial_No.					

Figure 4 Format of a RAP-reply packet.

(3) A call-for-substitution packet

A call-for-substitution packet as shown in Figure 5 consists of five main fields. The first is packet_no.3. The second is call-for-substitution-ID which is the RAP-ID of the corresponding RAP packet. The purpose is also identifying a specific assistance request. The third field is mobile node ID indicating which mobile node, e.g., node M, is selected as the assistant node. The fourth field is a list of direct neighbors which records N’s immediate upstream and downstream nodes. M on receiving the call-for-substitution packet starts moving toward N, and keeps its antenna working. Once it can receive all the radio signals sent by those nodes listed in this field, M then realizes that it is now in N’s EAZ. This is helpful in shortening an assistant node’s moving distance, consequentially saving its moving energy. Other mobile nodes on receiving the packet realize that they are not selected. Hence, relay the packet or drop the packet when necessary. An assistant node when arriving at the dying node N’s EAZ takes over the environmental sensing and packet relaying tasks until its energy is less than the predefined threshold. It then, like a dying static node, issues a RAP packet. Node N when realizing that M arrives at N’s EAZ does nothing until M dies. After that it starts sensing environment and relaying packets again. The fifth field is corona ID of which the function is the same as that described above. A corona-k node only relays call-for-substitution packets issued by a corona-k’s dying node.

1byte	1 byte	1 byte	1 byte	6 bytes	1 byte
Packet_no.3	Call-for-substitution-ID		Mobile node ID	List of direct neighbors	Corona ID
	Source_ID	Serial_No.			

Figure 5 Format of a call-for-substitution packet.

(4) A help packet

After an assistant node M issues an RAP packet, and before λ times out, no mobile node comes, M continues its sensing and relaying tasks. M before exhausting its energy sends a help packet to the node that it helps, e.g., node N, to request N taking over for M. A help packet as shown in Figure 6 consists of three main fields. The first is packet_no.4. The second is source _ID showing which assistant node, i.e., M, sends the packet. The third field is the caller ID, i.e., N, which is the node that called for M as an assistant, and of course is the destination of this packet. N on receiving the help packet starts relaying packets and sensing the environment. Other nodes on receiving the packet drop it.

1 byte	1 byte	1 byte
Packet_no.4	Source_ID	Caller ID

Figure 6 Format of a help packet.

(5) A re-routing request packet

A dying static node N, which cannot receive any RAP-reply packets before λ

times out, or which receives a help packet from its assistant node M, sends a re-routing request packet to the source node, e.g., node U, (i.e., an event source node (see Figure 1)), of the routing path that N is now on to request U to re-route a new path between U and the base station so that event packets can be continuously delivered. A re-routing request packet as shown in Figure 7 consists of three main fields. The first is packet_no.5. The second is re-routing-request-ID in which source_ID shows which node sends this packet, i.e., N, and the serial_number is used to identify the re-routing request. The third field indicates the destination of this packet, i.e., U, telling nodes that receive the packet where the destination of this packet is.

1 byte	1 byte	1 byte	1 byte
Packet_no.5	Re-routing-request-ID		Event source node ID
	Source_ID	Serial_No.	

Figure 7 Format of a re-routing packet.

(6) A path-release packet

When the event that an event source node U detects disappears, or U receives a re-routing request packet from one of its downstream nodes, e.g., node N, U sends a path-release packet along the routing path. Release all assistant nodes along the path. Static nodes along the path may continue maintaining information of the path or release the path info depending on the content of the packet. We assume that U only establishes a path to connect itself and the base station. A path-release packet as shown in Figure 8 consists of three main fields. The first is packet_no.6. The second is path-release-ID in which source_ID conveys the ID of the event source node U, and serial_number is used to discriminate a path-release request. The type of the last field release is boolean. If release=0, indicating the corresponding event has been removed. All static nodes along the routing path continue maintaining their routing information. If release=1, showing that U has received a re-routing request packet from one of its downstream nodes, all static nodes release their routing information, and wait for a RREQ packet issued by U to re-establish a new path. An active assistant node which is an assistant node that is now relaying packets and sensing the environment (rather than a dying assistant node) on receiving a path-release packet with release = 0 changes its state from assistant to mobile since the event disappears. We do not know when U will be an event source node again. If the packet is one with release = 1, it changes its state from assistant to static and sets a timer ψ to wait for an RREQ packet since the event is still there. Once ψ times out and it has not been selected as a node on the newly established routing path that connects U and the base station, the node changes its state from static to mobile to wait for a RAP packet.

1 byte	1 byte	1 byte	1 byte
Packet_no.6	Path-release-ID		Release
	Source_ID	Serial_No.	

Figure 8 Format of a path-release packet.

(7) **A dying-notification packet**

A dying static node N after sending out an RAP packet cannot receive any RAP-reply packets, before λ times out or when N receives a help packet from its assistant node M which implies that M is dying and cannot call for a mobile node to assistant it, N besides sending a re-routing request packet also transmits a dying-notification packet to the base station, notifying the base station to warn the system administrator that N's battery should be changed or recharged immediately to avoid the case that a later routing request or re-routing request due to some reasons cannot succeed, resulting the facts that the area near N is unmonitored and/or the event source node U's event packets cannot arrive at the base station. A dying-notification packet as shown in Figure 9 consists of three main fields. The first is packet_no.7. The source_ID = N, and serial_number is used to discriminate different dying-notification packets. The third field is the ID of the base station, telling a relay node where the packet should be sent to.

1 byte	1 byte	1 byte	1 byte
Packet_ no.7	Dying packet-ID		Base station ID
	Source_ID	Serial_No.	

Figure 9 Format of a dying-notification packet.

3.4 A Mobile Node Substitution

A dying static node N in corona k on finding that its residual energy level is lower than a predefined threshold sets a timer $\lambda = \frac{X}{Y}$, and sends out a RAP packet to call for one of the mobile nodes in corona k to come and act as an assistant node where X is the length of a half circle of corona k, i.e., $(k-\frac{1}{2})r \cdot \pi$, and Y is a mobile node's moving speed. We assume that all mobile nodes' moving speeds are the same. The following process is as follows. First, a mobile node on receiving a RAP packet sent by node N subtracts its consumption energy as if it moves from its current location to N's EAZ from its current residual energy as M's virtual residual energy E, and calculates $EPUD = E/l_{NM}$, where $l_{NM} = 2(k-\frac{1}{2})r \cdot \pi \cdot \frac{\alpha}{360°}$, here, α is the angle $\angle MCN$ and C is center of the field. If M's virtual residual energy is higher than or equal to 3Joules which is the predefined energy threshold, it replies a RAP-reply packet. An active or a dying assistant node on receiving the RAP packet will discard the packet if the packet has travelled through half circle of corona k. If the packet has not, M relays the packet. We assume a sensor needs energy of 8.27 Joules [29] to move per meter.

N continues collecting RAP-reply packets until λ times out. Then, the node, e.g., node Q, of which the EPUD is the highest and the virtual residual energy is higher than or equal to the predefined value (i.e., 3 Joules), will be selected as the assistant node by N. When there are several nodes that can be chosen, for saving energy, the geographically nearest one will be selected. Node N then issues a call-for-substitution packet to invite Q. The nodes other than Q, no matter they are static, mobile or assistant nodes, on receiving the packet will relay the packet. Q on receiving the packet checks the call-for-substitution ID and corona ID to see whether the packet is valid. If yes, implying that Q has sent the corresponding RAP-reply packet with the same call-for-substitute ID as its RAP-reply ID Q then continues computing if the line $\overline{NQ}$ is on the left or right hand side of $\overline{QC}$ where C is the center of the field. If $\overline{NQ}$ is on the right, Q moves along counterclockwise direction toward N. Otherwise, it moves clockwise. Q also turns on its antenna. As stated above, when it can receive signals from N’s all upstream and downstream nodes, it stops since Q is now in N’s EAZ. From now on, Q starts sensing the environment and relaying packets for N, and N enters its sleep mode until it receives a help packet from Q.

Q before its energy is lower than the predefined threshold sends a RAP packet to call for an assistant node. If it can successfully call for an assistant node, e.g., node M’, it enters its sleep model and M’ starts relaying packets and sensing the environment. When M’ is dying, it repeats the process that Q has done. Otherwise, i.e., no mobile nodes come to help Q, Q sends a help packet to node N. N on receiving the packet starts sensing its surrounding environment and relaying packets until it dies. But before it dies, it sends a re-routing request packet to the event source node U which will issue a path-release packet and a dying-notification packet.

However, if N sends out a RAP packet and cannot receive any RAP-reply packets after λ times out, like receiving a help packet from its assistant node, it sends a re-routing packet to U and a dying-notification packet to the base station.

When the event sensed by U disappears or the event source node U receives a re-routing request packet, U sends a path-release packet along the routing path to notify static nodes and assistant node respond properly as what has been described above. An assistant node M that changes it state from assistant to mobile will at as a mobile node. When another static node needs an assistant node, Q can move and turn itself into an assistant node again.

3.5 Algorithms performed by a Node

Now we would like to stand on a node's viewpoint to propose the algorithms that nodes perform. A static node Q on receiving a RAP, RAP-reply or a call-for-substitution packet, denoted by P, checks to see whether P's corona ID is the same as the one it locates. If not, it discards P. Otherwise, it further checks to see whether P has been received or not by comparing the P's source ID and serial member with those recorded in Q's P History tables which are tables recording all P-type packets that it has ever received. If yes, Q discards P. Otherwise, Q records P in its P History table, and relays P to the next nodes. If Q is a mobile node, it further checks to see whether or not its virtual residual energy exceeds a predefined threshold, e.g., 3 Joules. If yes, it computes EPUD value and returns a RAP-reply packet to the source node N. Otherwise it does nothing.

Further, if the P has traveled through a half circle of corona k, it will be discarded since P is sent clockwise and counterclockwise simultaneously along corona k.

When Q receives a path-release, re-routing request, dying-notification or data packet, denoted by P, it checks to see whether P has been received or not by checking its P History table. If yes, Q discards P. Otherwise, it broadcast P.

A mobile node M on receiving a RAP, RAP-reply or call-for-substitution packet, denoted by P, performs the activities the same as those performed by a static node. The difference is as follows.

If P is a RAP packet and M's virtual energy is higher than 3 Joules, M further replies N with a RAP-reply packet. If P is a call-for-substitution packet and P is for M, M moves clockwise or counterclockwise toward N. When P is a help packet issued by an assistant node A, Q substitutes for A and starts sensing the environment and relaying packets again.

A dying node, maybe a static or a mobile node, further checks to see whether a received RAP-reply is for it or not. If yes, it collects the mobile node's EPUD and checks to see whether the timer λ times out or not. If λ has been timed out, it selects a mobile node M as an assistant node, and then sends a call-for-substitution packet to M.

Also, should we stop the WSN when a static node dies and no mobiles node can come to help it? Our opinion is that, even several static nodes have died, the remaining sensors should keep sensing environmental changes. So at least a partial area of the field is under monitoring rather than totally unmonitored.

```
Algorithm 1: /*performed by a static node Q in corona k*/
Input: A received packet P with ID = (source ID, serial_number) = (N,X)
Output: P            /* relaying P*/
While (1)
{If (Q's energy is lower than predefined threshold)
        { broadcast an RAP packet; initiate a timer $\lambda$;}
Receive a packet P
switch (packet P)
{
case "a RREQ packet":
If (Q is not the base station and Q has sufficient energy and high signal quality
    of the link between Q and Q's upstream node and the RREQ has not been
    received)
{Q's ID is added to the RREQ packet ; broadcast the RREQ packet to
```

```
else    drop P; break;
case " a RAP packet":
If ((P. corona ID = corona ID (Q)) and ((N,X) does not exist in Q's P History
    table) and (P has not traveled through a half circle of corona k))
    {broadcast P ; record P in Q's P History table;}
    else    drop P; break;
case "a RAP-reply packet":
If (the RAP-reply packet P issued by a mobile node M is not for Q, i.e., P.
RAP-reply-ID. Source_ID $\neq$ Q)
If ((P. corona ID = corona ID (Q)) and ((N,X) does not exist in Q's P History
table) and(P has not traveled through a half circle of corona k))
    { send P toward N ; record P in Q's P History table;}
else    drop P;
else if ( RAP-reply ID, i.e., (N,X), issued by M is the RAP-ID of a RAP packet
that Q has sent out) /*a RAP-reply packet for Q*/
    If ( the corresponding  $\lambda$  has not timed out)
    { If ((N,X) does not exist in Q's RAP-reply History table and P has not
    traveled through a half circle of corona k)
    {append M and EPUD(M) to Q's Candidate table;
     record P in Q's P History table;} break; }
    else /* $\lambda$ times out*/
    { If (Q's Candidate table is empty) send a Re-routing request packet to U;
     /* U is the event source node of the oath that U is on*/
    else
```

```
        {select the mobile node, e.g., node M, with the highest EPUD; send a call-for-substitution packet to M; /* if several can be selected, M is the nearest node*/}}
        break;
    case " a call-for-substitution packet":
    If ((P. corona ID = corona ID (Q)) and ((N,X) does not exist in Q's P History table) and (P has not traveled through a half circle of corona k))
      {send P toward the mobile node whose node ID ==P. mobile node ID, i.e., mobile node M; record P in Q's call-for-substitution History table;}
        else    drop P;
    break;
    case "a path-release packet":
    If (P. Path-release-ID. Source_ID == U and (N,X) does not exist in Q's Path-release History table)) /* U: an event source node of the path, e.g., R, that Q is now on, and P is for the nodes on the path R */
    {If (P. Release==1) /* N is dying and the original routing path R is no more needed*/
    {delete P's routing information from Q's routing table; send P to the next node on the routing path between U and the base station; wait for a RREQ packet to re-establish a new path} /* If P. Release==0, Q does nothing*/}
    else drop P; break;
    case "a re-routing request packet":
    If ((P. Event source node ID ==U and $U \neq Q$ and Q is on the routing path between U and the base station and ((N,X) does not exist in Q's re-routing request History table))    /* Q is not the destination of P*/
```

```
{ send P to the next node along the reverse direction of the routing path;
  record P in Q's re-routing request History table;}
else If (( P. Event source node ID== Q) and ((N,X) does not exist in Q's
  re-routing History table)) /* Q is an event source node whose downstream
  node N issues a re-routing request packet to Q*/
{ send a path-release packet along the routing path; wait for x second;
  broadcast an RREQ packet;}
  /* to re-route a new path between Q and the base station*/
break;
case "a help packet":
If ( Caller ID == Q)
  { substitute for N and start sensing the environment and relaying packets}
else drop P; break;
case "a dying-notification packet":
If ((N,X) does not exist in Q's dying-notification History table )
    { If (Q is base station)
        {send a message to warn the system administrator that N's battery should
        be changed or recharged;}
    else    /*Q is not the base station*/    send P toward the base station;
    record P in Q's dying-notification History table;}
break;
case "a data packet":
If (P is issued by Q and P is not in Q's P History table and Q is on the routing
  path between U and the base station)
    send P toward the base station;
```

```
    else drop P; break;}
```

Figure 10 Algorithm for a static node on receiving a RREQ, RAP, RAP-reply, call-for-substitution, help, path-release, re-routing request or dying-notification or data packet.

```
Algorithm 2: /*performed by a mobile node M in corona k*/
 Input: A received packet P with (source ID, serial_number) = (N,X)
 Output: P and/or a RAP-reply packet, or M moves toward N's EAZ

switch ( packet P)
{
case "a RREQ packet":
    drop P; /* a mobile node does not act as a routing node*/
break;
case "a RAP packet":
    If ((P. corona ID = corona ID (M)) and ((N,X) does not exist in M 's P
        History table) and (P has not traveled through a half circle of corona k ))
    { broadcast P;
    compute M's EPUD value;
    If ( M's virtual residual energy $\geq$ 3J )
    { send a RAP-reply packet toward N ;
    record P in M's RAP History table;}
    else   drop P; break;
case "a RAP-reply packet":
    If ((P. corona ID = corona ID (M)) and ((N,X) does not exist in M's
    RAP-reply History table) and (P has not traveled though a half circle of
    corona k))
    {send P toward N; record P in M's RAP-reply History table;} /* P is sent by
```

```
        else    drop P; break;
    case "a call-for-substitution packet":
        If ((P. corona ID = corona ID (M)) and ((N,X) does not exist in M's
        call-for-substitution History table) and (P has not traveled though a half
        circle of corona k))
        {If ( P. mobile node ID==M)    /* P is for M */
            {If ( $\overline{NM}$ is on $\overline{MC}$'s right hand side ) /* C is the center of the sensing
            field*/
                M moves counterclockwise toward N's EAZ;
            else    M moves clockwise toward N's EAZ;
            while ( M is not in N's EAZ)
                keep moving along corona k;
            claim itself is now N's assistant node; Stop;}
        else /*P in not for M*/ {send P toward the node whose ID==P. Mobile node
            ID; record P in    M's call-for-substitution History table;}
        else    drop P; break;
    default: /*receiving all other packets, including a path-release, a re-routing
    request, a dying-notification, a help or a data packet*/
        drop P
    }
```

Figure 11 Algorithm for a mobile node on receiving a RREQ, RAP, RAP-reply, call-for-substitution, help, path-release, re-routing request, dying-notification or data packet.

```
Algorithm 3: /*performed by a assistant node M in corona k*/
  Input: A received packet P with (source ID, serial_number) = (N,X)
  Output: P, a RAP packet, a RAP-reply packet, a help message or M moves
```

```
            toward N's EAZ

If ( M's energy is lower than the pre-defined threshold)
    {broadcast an RAP packet; issue a timer $\lambda$ ;}
If (M is an assistant node helping node N and is going to exhaust its energy)
    {Send a help message to N;}
switch ( packet P)
{
case "a RREQ packet":
The activities are the same at those when a static node receives a RREQ packet.
case "a RAP packet":
The activities are the same at those when a static node receives a RAP packet.
case "a RAP-reply packet":
The activities are the same at those when a static node receives a RAP-reply
packet.
case "a call-for-substitution packet":
The activities are the same at those when a static node receives a
call-for-substitution packet.
case " a path-release packet":
    If (P. Path-release-ID. Source_ID == U and (N,X) does not exist in M's
    Path-release History table)) /* U: assistant node M's event source node of the
    path*/
    If (P. Release==0)
    change its state from assistant to mobile;
    else
    { change its state from assistant to static;
```

```
        set a timer $\psi$;
    If ( before $\psi$ times out and M does not receive a RREQ packet from U)
    change its state from static to mobile;}
    else    drop P;   /*Source node ID $\neq$ U*/
case " a re-routing request packet":
    If ( P does not exist M's P History table)
    send P toward U;
    else    drop P; break;
case " a dying-notification packet":
    If (P does not exist in M's P History table)
    send P toward the base station;
    else    drop P; break;
case " a help packet":
    If (P. caller ID==M and P does not exist in M's P History table)
    { substitute for N, which is the assistant node that comes to help M; start
    sensing the environment and relaying packets for N;}
    drop P; break;
case " a data packet":
   send P toward the base station;
break;}
```

Figure 12 Algorithm for assistant node on receiving a RREQ, RAP, RAP-reply, call-for-substitution, help, path-release, re-routing request, dying-notification or data packet.

```
Algorithm 4: /*performed by a base station B*/
Input: A received packet P with ID = (source ID, serial_number) = (N,X)
Output: P            /* relaying P*/
```

```
switch (packet P)
{
case "a RRQ packet":
If (the RREQ has not been received)
    broadcast a route reply packet along the reverse direction of the routing path ;
break;
case "a dying-notification packet":
If ((N,X) does not exist in B's dying-notification History table )
{send a message to warn the system administrator that N's battery should be changed or recharged;}
break;
case "a data packet":
    {collect P; store P}; break;
default: drop p
}
```

Figure 13 Algorithm for base station on receiving a RREQ, RAP, RAP-reply, call-for-substitution, help, path-release, re-routing request, dying-notification or data packet.

Chapter 4. Experiments and Discussion

4.1 Experiment Environment

A network's lifetime is affected by several factors, such as node density, a node's initial energy, and events' occurrence frequency. The latter is involved because a node in a higher-event-frequency environment consumes more energy than that consumed in a lower-event-frequency environment.

In this study, eight experiments were performed, all on a 250×250 m^2 sensing field, given 5% to 70% of sensors triggered once per ten seconds by events, and triggered nodes are randomly selected. The base station is located at the center of the field, i.e., x = 125, y = 125. The compared schemes include Routing Assistant Scheme with Localized Movement (RASLM) [6], Ad-hoc On-Demand Distance Vector (AODV) [25], Dynamic Source Routing (DSR) [26] and Destination-Sequenced Distance-Vector (DSDV) [27]. A WSN's lifetime is defined as the working time from system start-up to the time point when the first static node died. The transmission radius r of a sensor is 75m. We assume a sensor consumes energy of 8.27Joules [29] to move per meter. The energy required to transmit (receive) a packet is 0.221 (0.205) Joules [28], and the initial energy of a sensor is 3000 Joules (generally, a battery of 2500mAh is 10800 Joules). Generally, density is 4 and the total energy given to all nodes distributed to the field is 300K Joules.

The first experiment compared the lifetime of the five schemes given an environment with uniform node density and uniform initial node energy. A total of 100 nodes, 75 static and 25 mobile, are used. During the node distribution, nodes were randomly selected from the 100 nodes and uniformly distribution to the field. After that nodes were randomly selected and periodically triggered. The second redid the first environment, but we fixed the number of static and dynamic nodes distributed to non-uniform node density but uniform initial node energy. The fourth redid the second experiment, but a uniform node density and non-uniform non-uniform initial node energy environment was given. The fifth redid the second experiment, but a non-uniform node density and initial node energy environment was given. The sixth, the seventh and the eighth respectively redid the third, the fourth and the fifth experiments, but the lifetime is defined as the time period from when the system starts to the time point when x% of nodes die instead of when the first static node dies, where x ranges from 10% to 100%. Each experiment was performed 30 times.

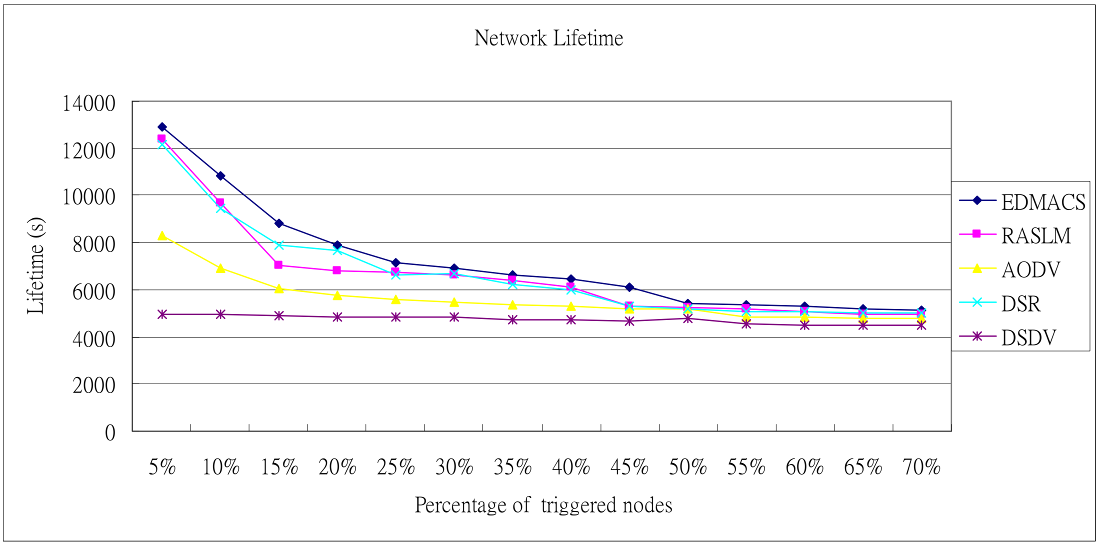


Figure 14 5% to 70% of sensor nodes are periodically triggered by events, and static nodes and mobile nodes are randomly distributed to the sensing field.

4.2 Uniform Node Density and Uniform Initial Node Energy

(1). Nodes were Randomly Distributed and Triggered Nodes were Randomly Selected

Figure 14 shows the results of the first experimental in which the EDMAC outperformed the other tested schemes since it called for mobile nodes to assist dying nodes. This can truly prolong a network’s lifetime. But when percentage of triggered nodes is over 50%, network lifetime is similar, because many more events quickly used up static and mobile nodes’ energy, and a dying node cannot call for sufficient or any assistant nodes.

With the AODV and DSR, when a source node desires to deliver a packet, it broadcasts a route RREQ packet across the network to establish a routing path. However, maintaining routing tables and establishing a routing path (when an event occurs) consume very much energy [11]. DSR employs the source-routing mechanism with which when a RREQ packet arrives at a node P, P records its node ID in the route request table. So when a routing packet is routed to the base station, the base station can choose an optimal path. DSR performed well in static and low-mobility environments [26]. Basically, it is beacon-less so it does not deliver hello packets periodically. On the other hand, the AODV as a routing protocol that floods control packets, records the source node ID and the next node ID in a node once the node receives a RREQ packet during the path establishing stage. When it finishes delivering data, in order to relay packets when the event occurs at the same location again, it must maintain the routing information. Also to correctly maintain the information, nodes in the system that employs the AODV must mutually exchange messages periodically, but the DSR only checks the information in the route request table. So AODV nodes consume very much energy. That is why it is not better than the DSR.

With the DSDV, each node must record all of links that connect to it and the distance of the links. In order to maintain a route, nodes must update their neighbor’s node information periodically. So nodes must always keep their states on active mode, thus consuming lots of energy in maintaining routing tables, which is called the overhead problem. In Figure 14, no matter what the percentage of triggered nodes was, due to the overhead problem, network lifetimes were always about the same.

With the RASLM, a dying node calls for k mobile nodes, $k \geq 1$, to assist relaying packets for it so as to prevent the node from exhausting energy quickly. The EDMAC has the same function, but it calls for only one mobile node instead of several nodes, for a dying node N, thus consuming less moving energy. This is the key reason why the RASLM's lifetime was shorter than the EDMAC's.

(2). Uniform Environments on Different Node Densities

In the second experiment, the uniform distribution environment was tested given different node densities, including 3, 4 and 5. Numbers of mobile nodes and static nodes distributed to different coronas are listed in Table 1. The number of nodes distributed to the sensing field were respectively 75, 100 and 125 instead of 50 only. Initial energy of each node was 3K Joules, and the total energy given to all nodes distributed to the field was respectively 225K Joules, 300K Joules and 375K Joules. Note that a corona-k mobile node only moves in corona k. The purpose is to keep each corona's energy consumption rate being the same so that all nodes run out of the energy and die almost at the same time.

Table 1 The numbers of nodes distributed to coronas of the sensing field for the second experiment.

Density / Kind of Node / Corona #	3		4		5	
	Static	Mobile	Static	Mobile	Static	Mobile
0	2	1	3	1	3	2
1	6	3	9	3	11	4
2	11	4	15	5	18	7
3	15	6	21	7	26	9
4	20	7	27	9	33	12
Total	75		100		125	

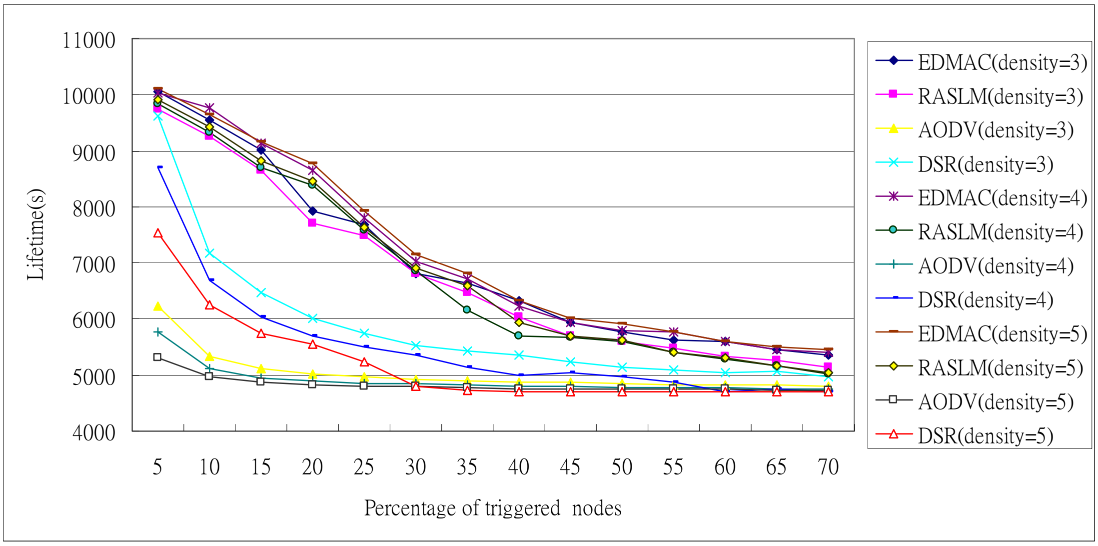


Figure 15 5% to 70% of nodes were triggered randomly given the same node density to all coronas, and node densities of the sensing field range from 3 to 5.

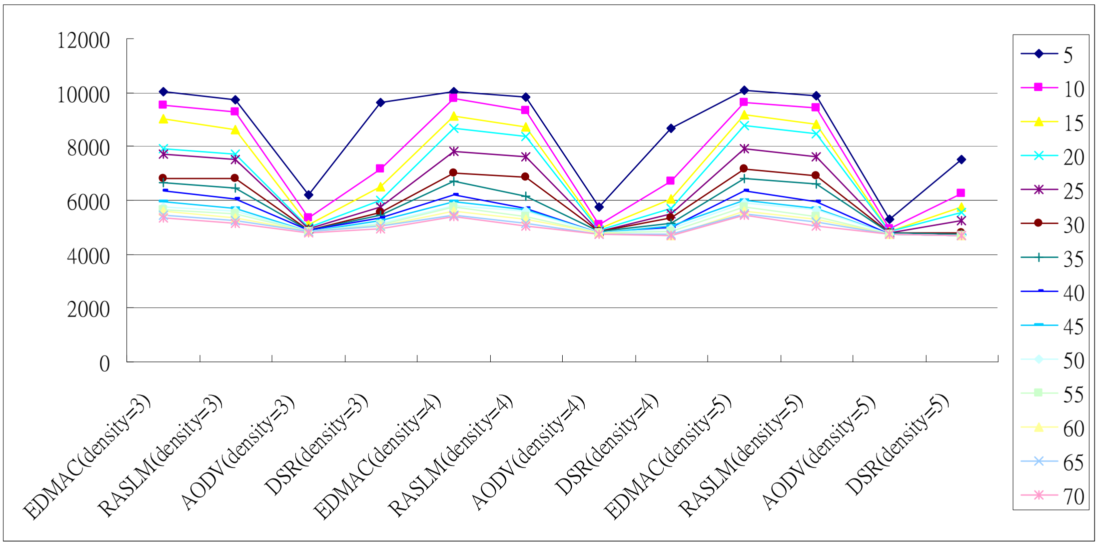


Figure 16 Node densities of the sensing field range from 3 to 5, and 5% to 70% of nodes were triggered randomly given the same node density to each corona.

Figure 15 shows that all schemes' lifetimes declined quickly. The reason is energy-hole problem [16]. In an event driven environment, increasing node densities causes the AODV and the DSR generating many more control packets, thus consuming much more energy. The AODV broadcasts hello messages periodically to maintain the routing path. This also consumes a lot of energy. Therefore, when node densities are higher (i.e., from 3 to 5), the two schemes had shorter lifetimes. But the lifetimes of the DSR are longer than those of the AODV. The reason is stated above.

When node densities increased, RASLM's and EDMAC’s lifetimes were longer, but AODV’s and DSR’s were not. The reason has been stated above. If we compare Figures 14 and 15, it is clear that when percentages of triggered nodes are 5%, 10% and 15%, the lifetimes in Figure 14 are relatively longer. The reason is that in the first experiment, mobile and static nodes are randomly distributed to the field, and mobile nodes can initially move across coronas, i.e., dying nodes can call for assistant nodes from other coronas. So lifetime was longer. The lifetime of the system then followed. But when percentages of triggered nodes are larger than 15%, the lifetimes in Figure 15 were relatively longer, indicating that when insufficient mobile nodes can go across coronas, our approach is better.

The RASLM and the EDMAC had longer lifetimes than those of the DSR and AODV. The reason is also that calling for mobile nodes can mitigate energy-hole problem and prolong network lifetime.

4.3 Non-uniform Node Density and Uniform Initial Node Energy

In the third experiment, different coronas were given different node densities. The number of nodes distributed to the sensing field was also 100, and initial energy of each node was 300K Joules, The numbers of mobile nodes and static nodes distributed to each corona, and their node densities are listed in Table 2.

Table 2 The numbers of nodes distributed to and node densities allocated to coronas for the third experiment.

Corona Inform.	**0**	**1**	**2**	**3**	**4**
Mobile	2	5	6	6	7
Static	5	13	16	19	21
Node Density	7	6	4.4	3.6	3.1
Corona total energy	21000	54000	66000	75000	84000

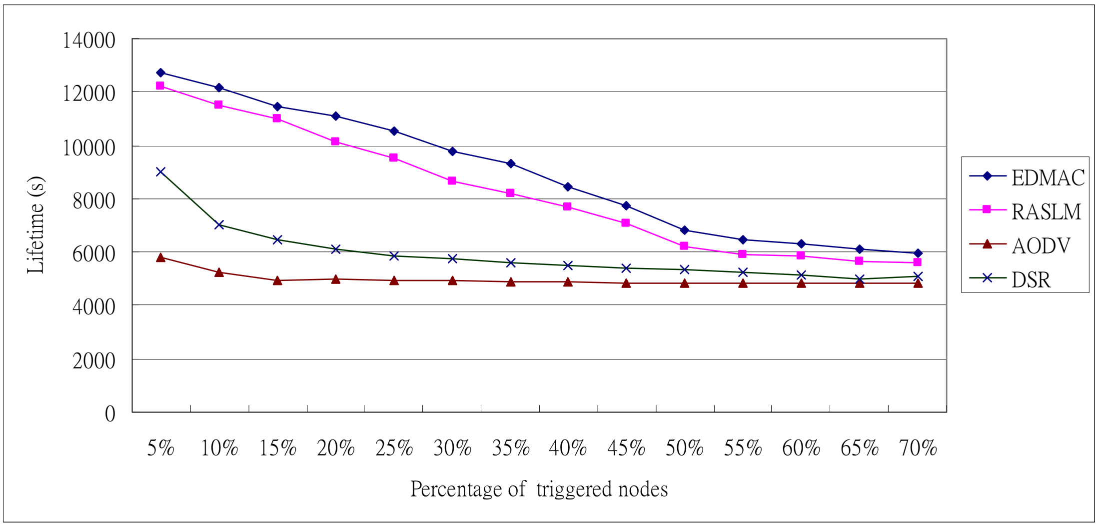


Figure 17 5% to 70% of nodes were triggered randomly given the same node initial energy level to each node and different node densities to different coronas.

Figure 17 shows that the AODV’s and the DSR’s lifetimes declined quickly. Nodes in both schemes did not die almost at the same time because several nearby and frequently occurred event points might force those nodes shared by several routing paths to exhaust their energy quickly, and nodes could not call for assistant nodes. What they could do was rerouting, which would again cause a huge number of RREQ packets, particularly for inner coronas since their densities were relatively higher than those of outer coronas, thus draining up node energy fastly. But the RASLM and EDMAC had no this problem.

On the other hand, the AODV lifetimes in Figure 17 were not relatively longer than those in Figure 15 (see those plots of density=4 since they are all 100 nodes in the field), because a higher node-density corona generated more RREQ packets and maintained many more routing tables than those generated and maintained by a lower node-density corona, forcing the energy hole problems coming much earlier and consequently shortening the AODV’s and the DSR’s lifetime. In Figure 17, the EDMAC’s and RASLM’s lifetimes are significantly longer than those illustrated in Figure 15, showing that increasing inner-corona node densities can prolong network lifetime. However, the lifetimes, except on percentage of triggered node = 5% (they were almost equal), were all longer than those of Figure 14, indicating that increase node densities for inner coronas can lengthen the two schemes' lifetimes.

4.4 Uniform Node Density and Non-uniform Initial Energy

In the fourth experiment, a uniform node density environment was given, but nodes in different coronas were allocated different initial energy levels, the inner, the higher. The numbers of nodes distributed to the sensing field were 100, and node density was 4 for each coronas. The total energy was 300K Joules. The numbers of mobile nodes and static nodes distributed to different coronas and their initial node energy levels are listed in Table 3.

Table 3 The number of nodes distributed to coronas and initial node energy levels allocated to nodes in different coronas for the fourth experiment.

Corona / Inform.	**0**	**1**	**2**	**3**	**4**
Mobile	1	3	5	7	9
Static	3	9	15	21	27
Initial Node energy	3505.6J	3313.8J	3131.7J	2959.4J	2797.6J

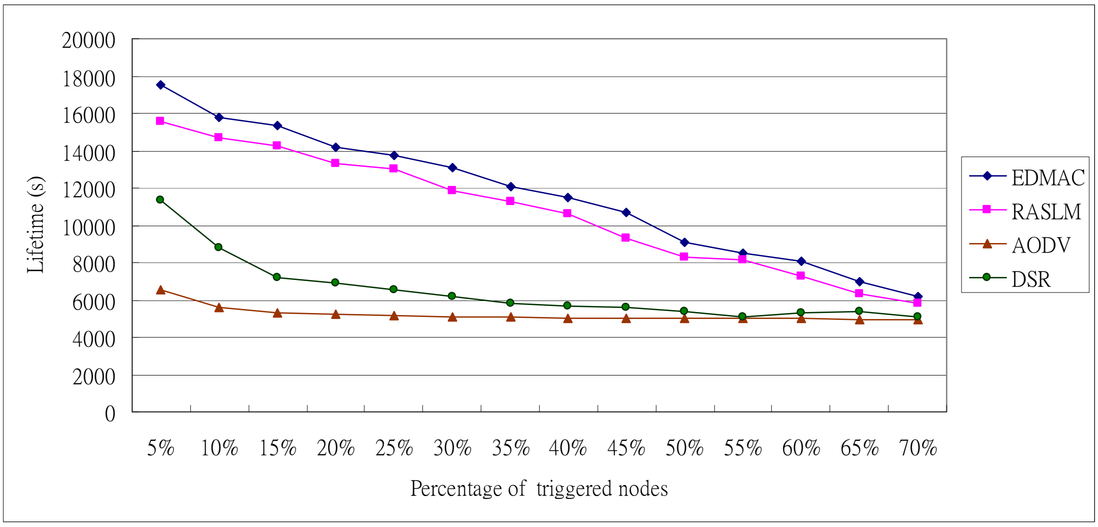


Figure 18 A uniform node density environment, in which the node densities of all coronas are 4, and the initial energy levels allocated to nodes of different coronas are listed in Table 3. 5% to 70% of nodes were triggered randomly.

Figure 18 shows that the lifetimes of the four schemes were individually and significantly longer than those illustrated in Figures 15 and 17 since higher initial node energy in inter coronas can postpone the occurrence of an energy hole. The AODV’s and the DSR’s lifetimes also declined quickly. The reason was stated above, even though nodes in an inner corona were allocated higher initial energy levels.

Comparing Figures 18 and 17, the numbers of nodes involved are all 100 and the total energy is 300K Joules. But one is non-uniform initial node energy, and the other is non-uniform node density. We can see that non-uniform initial node energy environment lasted longer.

4.5 Non-uniform Node Density and Non-uniform Node Initial Energy

In the fifth experiment, a non-uniform node density and initial energy environment was given. Nodes in an inner-corona node were allocated a higher initial energy level, and inner coronas were given higher node densities. The number of nodes distributed to the sensing field was 100, of which the sum of energy was 300K Joules. The numbers of mobile nodes and static nodes distributed to different coronas, and their initial node energy levels are listed in Table 4.

Table 4 The number of nodes and node densities distributed to coronas and node initial energy levels allocated to nodes of different corona for the fifth experiment.

Corona Inform.	**0**	**1**	**2**	**3**	**4**
Mobile	1	3	5	7	8
Static	5	10	16	21	24
Initial Node energy	3566.1J	3342J	3131.7J	2935.3J	2725.5J
Density	4.3	4.16	4	3.85	3.6
Corona total energy	21396.6	43446	65765.7	82188.4	87216

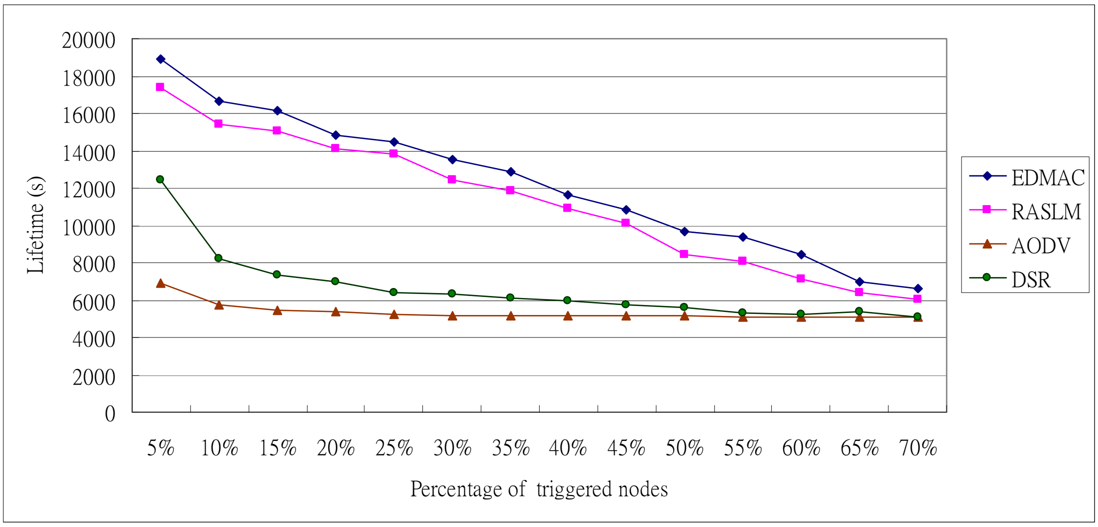


Figure 19 A non-uniform node density environment, in which the number of nodes and node densities distributed to coronas, the initial energy levels allocated to nodes of different coronas are listed in Tables 4. 5% to 70% of nodes were triggered randomly.

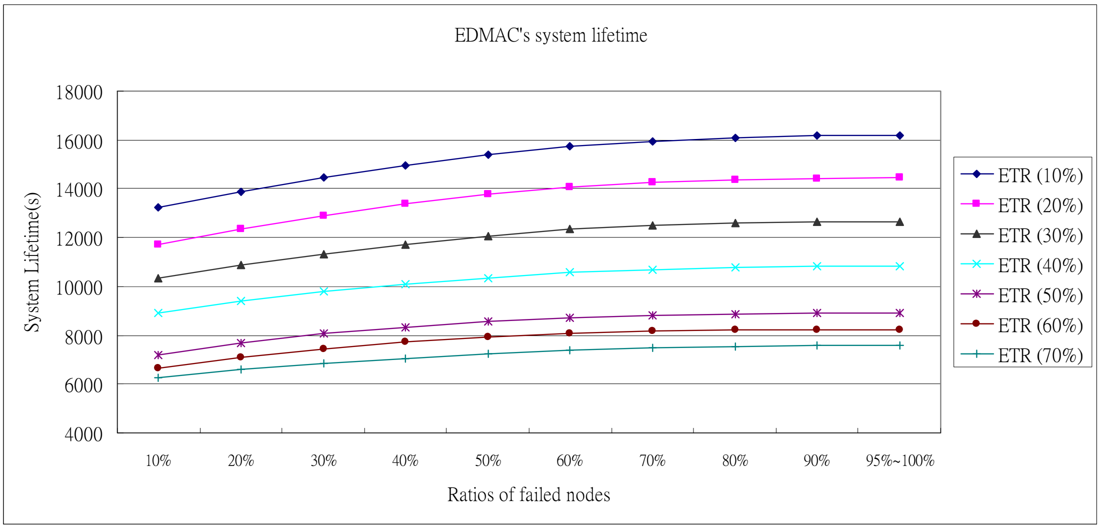
Figure 19 shows that the lifetimes of the four schemes are individually and significantly longer than those illustrated in Figures 17 and 18. Figure 17 shows that giving inner coronas higher node densities can prolong EDMAC's and RASLM's network lifetimes, and Figure 18 illustrates that giving inner-corona nodes higher initial energy levels and inner coronas higher node densities can further prolong the two schemes' network lifetimes. Comparing Figure 19 with Figure 17, even though node densities of inner corona of Figure 19 were lower (see Tables 4 and 2), but giving inner coronas higher corona total energy resulted in better performance. As comparing Figure 19 with Figure 18 and Table 4 with Table 3, we can conclude the similar result. The AODV's and the DSR's lifetimes also declined quickly, even though their inner coronas were allocated higher initial energy levels, and the EDMAC's lifetimes are longer than those of the RASLM. The reasons are stated above.

4.6 System Lifetime on Different Ratios of failed Nodes

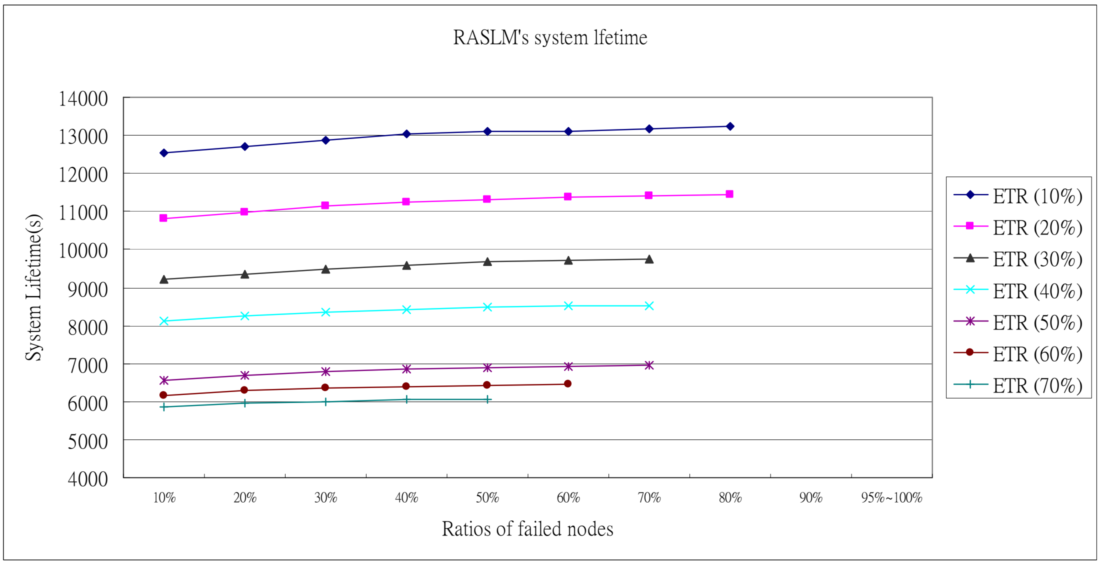
In the following, we redefine the network lifetime which is the time period from when the system starts to the time point when x% of nodes fails where x ranges from 10% to 100%. The propose of this control scheme is to make all nodes to die at the same time.

(1) Non-uniform Node Density and Uniform Initial Node Energy

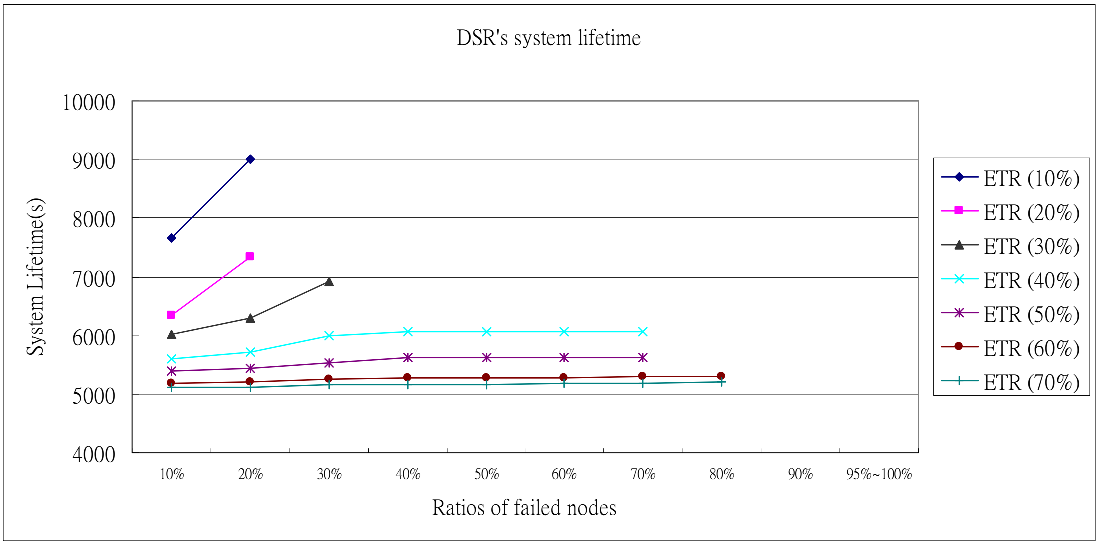
In the sixth experiment, we redid the third experiment and measured the system lifetimes on different ratios of failed nodes given different event-triggered ratios (ETRs) ranging from 10% to 70%, i.e., 10% to 70% of nodes were triggered randomly, to further validate the lifetimes of the EDMAC, RASLM, DSR and AODV.



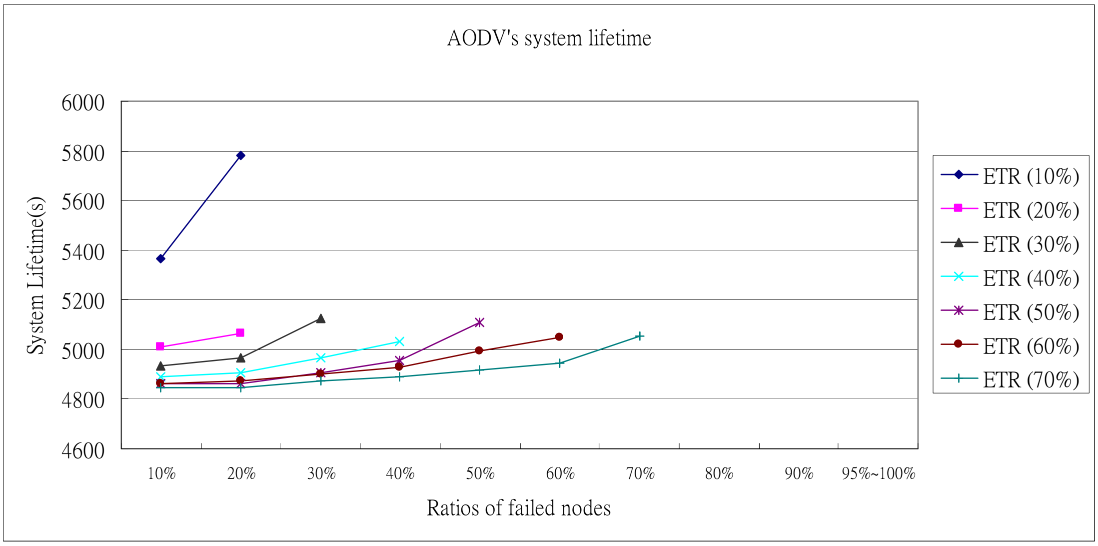
(a) The EDMAC’s system lifetimes.



(b) The RASLM’s system lifetimes.



(c) The DSR’s system lifetimes.



(d) The AODV’s system lifetimes.

Figure 20 The system lifetimes of the four tested scheme on different ratios of failed nodes given different event-triggered ratios (ETRs), ranging between 10% and 70%.

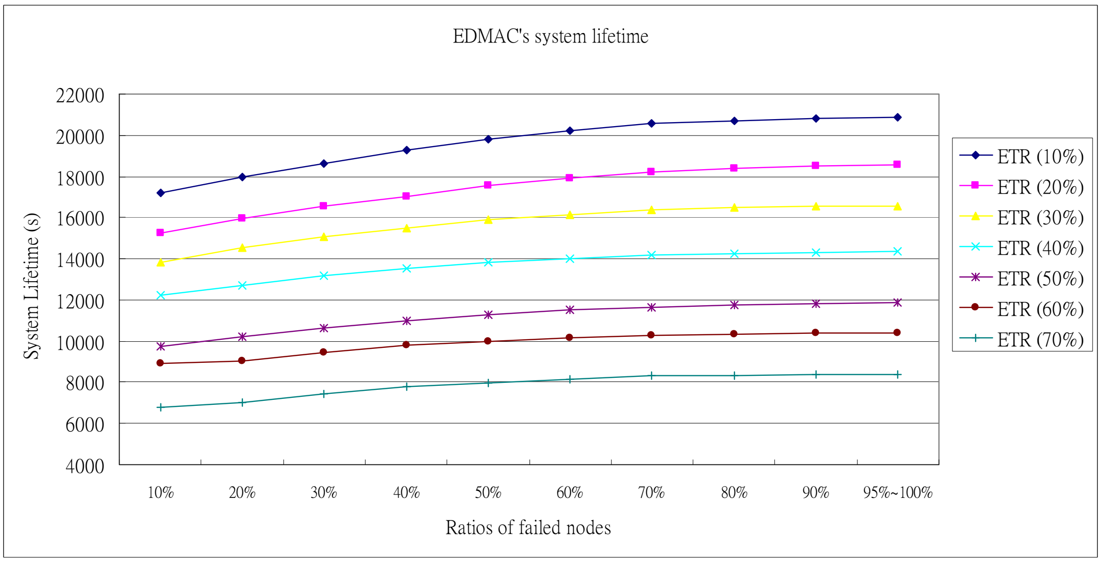
Figure 20 shows the experimental results, in which Figure 20a illustrates that the lifetime of the EDMAC can sufficiently utilize node energy i.e., even though a lot of nodes failed, but the EDMAC can continue its sensing and relaying tasks until 100% or almost 100% of nodes die. The reasons are that inner coronas were given higher node densities, and a static node can call for assistant nodes, thus postponing the occurrence of the energy hole problem, which in turn prolongs network lifetime and enables higher utilization of node energy. When ETR is 10%, system lifetimes increase sharply because the number of packets relayed by a node is not high, and fewer nodes died. When ETR is higher, system lifetimes increased a little slowly. Because when many events occurred at the same time, many nodes consumed energy quickly, forcing those nodes to die fastly, which consequently shortens system lifetimes.

In Figure 20b, when ETR=10% and 20%, radio of failed nodes can only achieve 80%. This means under the circumstances, due to the energy hole problem no more packets can be delivered to the base station, and re-routing can not succeed. Thus nodes consumed no more energy. In Figures 20c and 20d, when ETR=10% and 20%, packets are not hugely generated and only small portion of nodes relays packets. So energy hole problem occurred relatively later. That is why their lifetimes relatively longer than those when ETRs were larger. Also, the energy hole occurred when those relaying died, leaving other nodes with lot of energy. So it is impossible for them to consume energy since no packet can be delivered to the base station. When ETR increases, many more nodes relay packets at the same time, leading to the fact that network lifetimes were shorter.

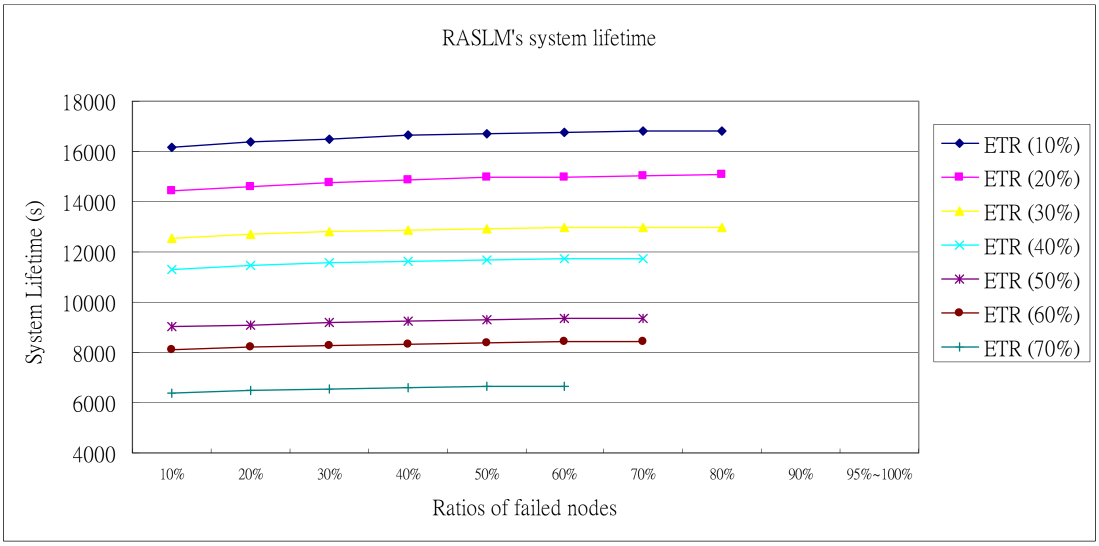
Comparing Figures 20a, 20b, 20c and 20d, it is clear that except the EDMAC, other schemes' lifetimes could not achieve 95% to 100% of failed nodes. The reason also is energy hole problem. Nodes in an inner corona totally died, but nodes in outer coronas were still alive. Their packets can not be delivered to the base station. The EDMPA lasts until all nodes' death. This can effectively prolong system lifetime, and increase node energy utilization.

(2) **Uniform Node Density and Non-uniform Initial Energy**

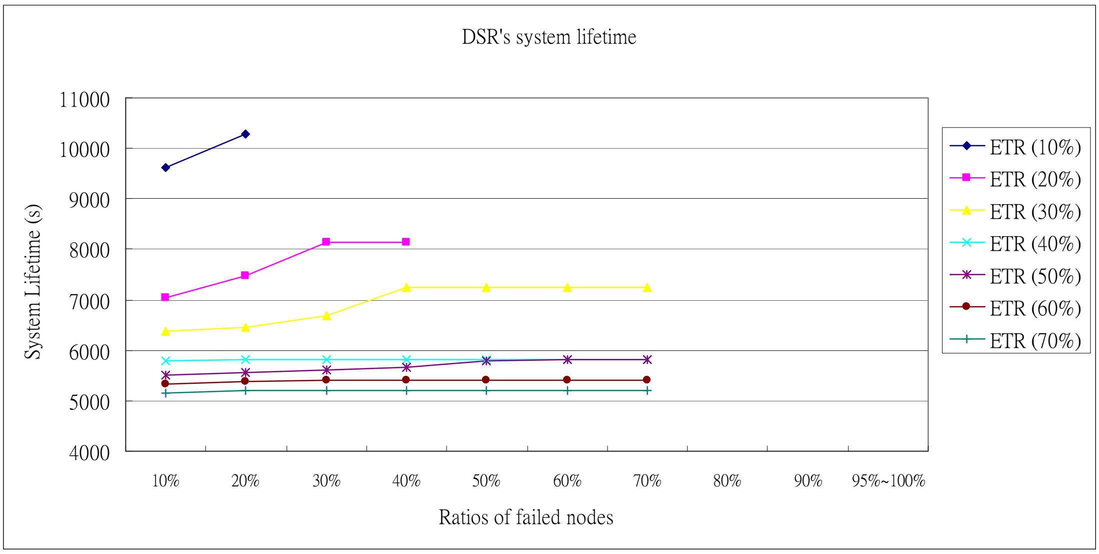
In the seventh experiment, we redid the fourth experiment and also measured the



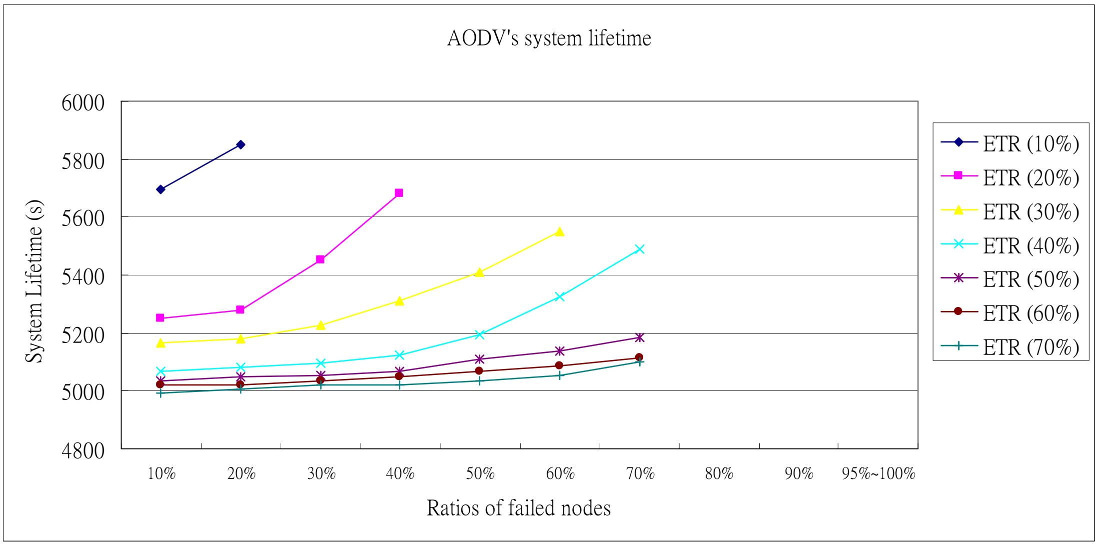
(a) The EDMAC’s system lifetimes.



(b) The RASLM’s system lifetimes.



(c) The DSR's system lifetimes.



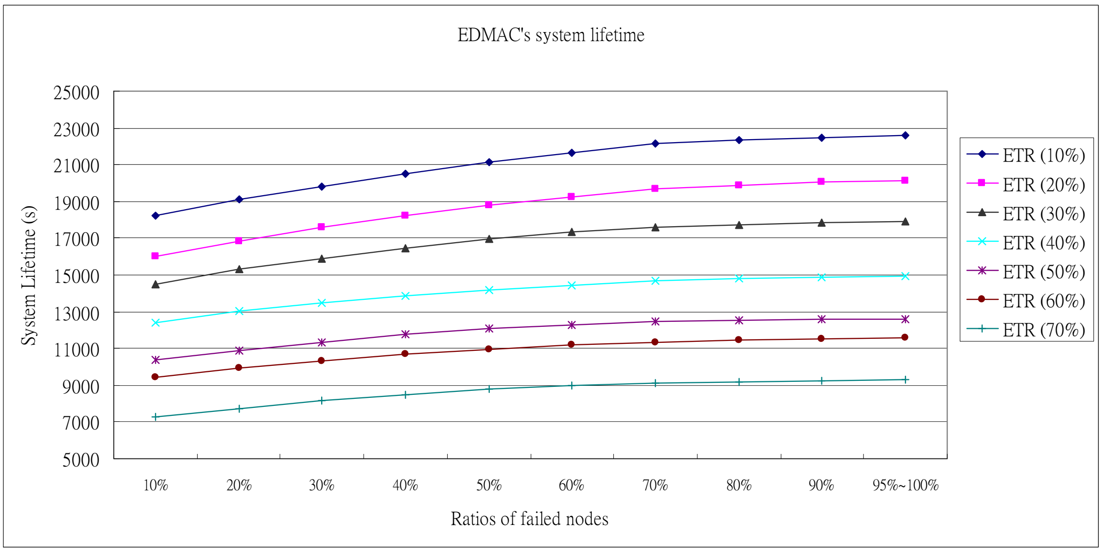
(d) The AODV's system lifetimes.

Figure 21 The system lifetimes of the four tested schemes on different ratios of failed nodes given different event-triggered ratios (ETRs), ranging between 10% and 70%.

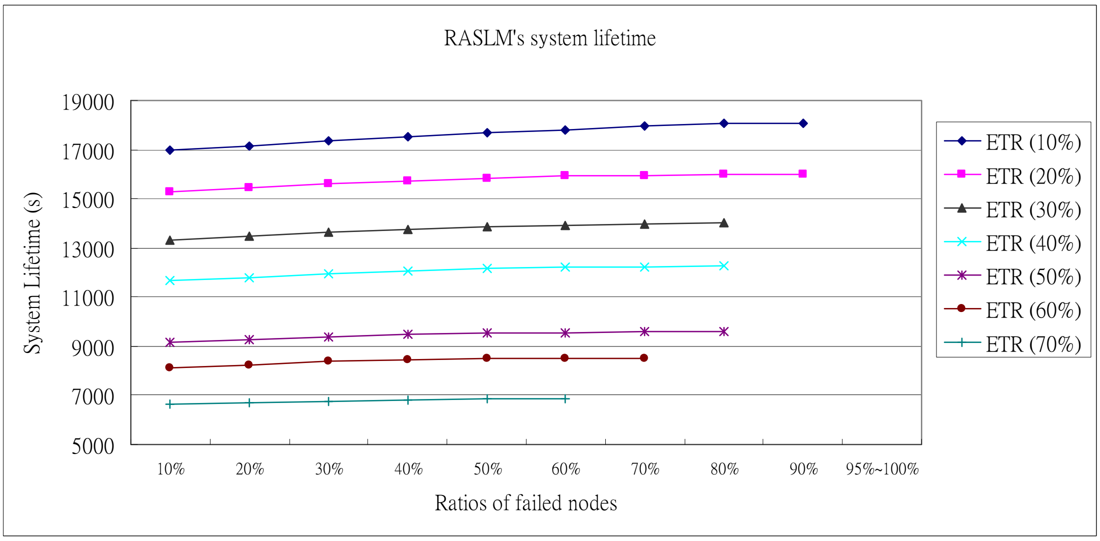
By comparing Figure 21a with Figure 20a, we can see that the EDMAC's lifetimes are longer, indicating that giving higher initial energy level to a node in an inner corona can really prolong the system lifetime. Figures 20a and 21a also illustrate that when a lot of nodes failed, the EDMAC could continue its sensing and relaying tasks. By comparing Figures 21a, 21b, 21c and 21d, it is clear that the EDMAC outperformed the other three schemes.

(3) Non-uniform Initial Energy and Node Density

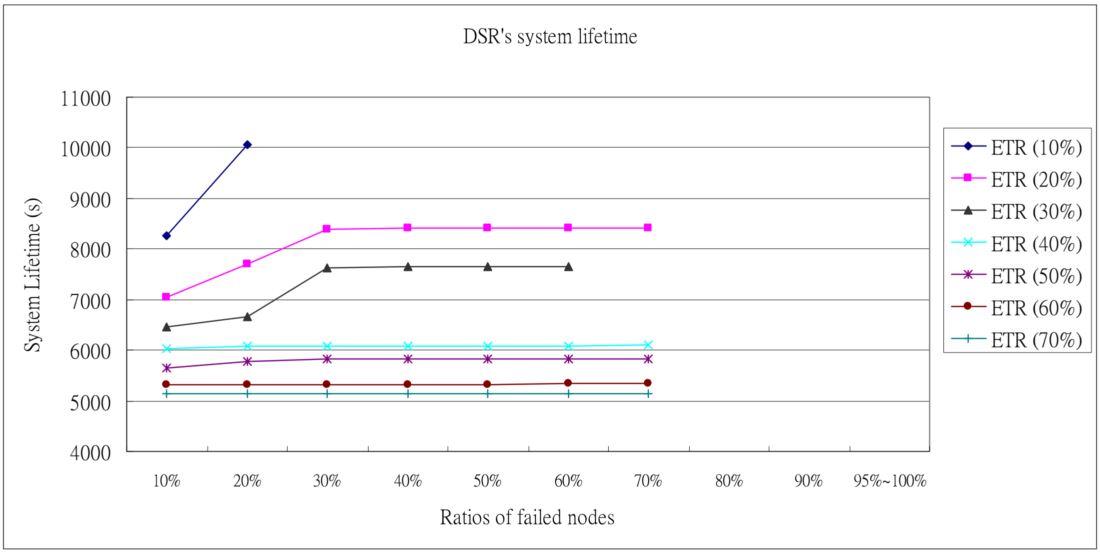
In the eighth experiment, we redid the fifth experiment and measured the system lifetimes.



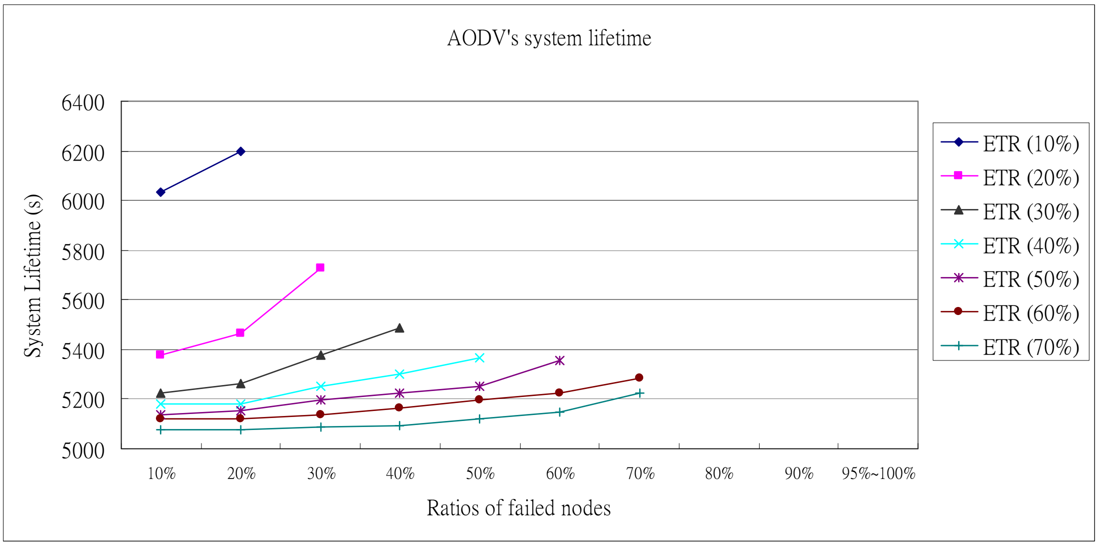
(a) The EDMAC’s system lifetimes.



(b) The RASLM’s system lifetimes.



(c) The DSR’s system lifetimes.



(d) The AODV’s system lifetimes.

Figure 22 The system lifetimes of the four tested schemes on different ratios of failed nodes given different event-triggered ratios (ETRs), ranging between 10% and 70%.

By comparing Figure 22a with Figures 20a and 21a, it is clear that this experiment has the longest system lifetimes. The reason that is increasing both inner coronas’ node densities and inner-corona nodes’ initial energy levels can effectively prolong a network’s lifetime. By comparing Figures 22a, 22b, 22c and 22d, we can see that the EDMAC still outperformed the other three schemes.

Chapter 5. Conclusions and Future Research

In this study, in order to reduce energy consumption and prolong network lifetimes, we proposed the control scheme, the EDMAC, which calls for a mobile node as the assistant node to help a dying node to sense the environment and relay packets, for wireless sensor networks. A total eight experiments were performed. The first is a periodical event-driven environment in which nodes are uniformly distributed to the field and each node has the same initial energy. Experimental results show that the EDMAC’s network lifetimes are longer than those of the other three tested schemes. When more events occur simultaneously, the EDMAC’s network lifetime is still longer than those of the other three. The second is also a uniform node density and uniform initial node energy environment. But each corona was distributed a specific amount of mobile nodes. The experimental results also show that the EDMAC has the longest lifetime. The third environment is the one with non-uniform node density and uniform initial node energy. The fourth is one with uniform node density and non-uniform initial energy, and the fifth is the one with non-uniform node density and non-uniform initial node energy. The results of three experiments also illustrate that the EDMAC outperformed the other tested schemes. The results of the sixth, seventh and eighth experiments show the eighth environment i.e., non-uniform node density and non-uniform initial node energy environment is the best. Because increasing inner coronas’ node densities and initial node energy both can truly prolong network lifetimes, and this experiment was performed on the one that compromises the two environments. So its network lifetimes are the longest. The worst one is the second environment. Since we fixed the numbers of mobile nodes for each corona, and all nodes’ initial node energy levels are the same. Inner coronas die very quickly, resulting in an energy hole much earlier than others. The environments of the sixth, seventh and eighth experiments are respectively the same as those of the fourth, fifth, and sixth. The results show that an EDMAC system can last until when 95% to 100% of nodes fail.

In the future, we would like to derive the EDMAC's reliability and behavior models. So, users can predict the reliability and behavior of the system before using it. Furthermore, we will try to integrate MAC layer protocols, like sleeping mode scheduling, with the EDMAC to further improve its system efficiency and lifetime so as to save as much energy as possible, especially when ETR of a sensor network is high. These constitute our future research.

References

[1] M. H. Chiang, G .T. Byrd, “Neighborhood-Aware Density Control in Wireless Sensor Networks,” IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp.122-129, June 2008.

[2] F. L. Villafuerte, K. Terfloth and J. Schiller, “Using Network Density as a New Parameter to Estimate Distance,” International Conference on Networking, pp. 30-35, April 2008.

[3] R. S. Chang and C. J. Kuo, “An Energy Efficient Routing Mechanism for Wireless Sensor Networks,” International Conference on Advanced Information Networking and Applications, vol. 2, pp. 308-312, April 2006.

[4] D. Wei and H. A. Chan, “Clustering Ad Hoc Networks: Schemes and Classifications,” Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, vol. 3, pp. 920-926, Sept. 2006.

[5] S. Sivavakeesar, G. Pavlou, C. Bohoris, and A. Liotta, “Effective Management Through Prediction-based Clustering Approach in the Next-Generation Ad Hoc Networks,” IEEE International Conference on Communications, vol. 7, pp. 4326-4330, June 2004.

[6] F. Y. Leu, W. C. Wu, and H. W. Huang “A Routing Scheme with Localized Movement in Event-Driven Wireless Sensor Networks,” the International Conferences on Advances in Data and Web Management, vol. 5446, pp. 576 - 583, April 2009.

[7] S. Yi and S. Hongchi, “Coverage and Energy Tradeoff in Density Control on Sensor Networks,” the International Conference on Parallel and Distributed Systems, vol.1, pp. 564-70, July 2005.

[8] Y. Xu, J. Heidemann, and D. Estrin, “Geography informed energy conservation

[9] M. Marta, and M. Cardei, “Using Sink Mobility to Increase Wireless Sensor Networks Lifetime,” the International Symposium on World of Wireless, Mobile and Multimedia Networks, pp. 1-10, June 2008.

[10] Y. Yang, and M. Cardei, “Movement-Assisted Sensor Redeployment Scheme for Network Lifetime Increase,” the ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, pp. 13-20, Oct. 2007.

[11] 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.

[12] T. van Dam and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” the international conference on Embedded networked sensor systems, pp. 171-180, November 2003.

[13] W. Ye, J. Heidemann, and D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” the Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1567-1576, June 2002.

[14] Y. Liu and L.M. Ni, “A New MAC Protocol Design for Long-term Applications in Wireless Sensor Networks,” the International Conference on Parallel and Distributed Systems, vol. 1, pp. 1- 8, Dec. 2007.

[15] W. Wang, V. Srinivasan, and K. Chua, “Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks,” the annual international conference on Mobile computing and networking, pp. 270-283, Aug. 2005.

[16] X. Wu, G. Chen, and S.K. Das “Avoiding Energy Holes in Wireless Sensor Networks with Non-uniform Node Distribution” IEEE Transactions on Parallel and Distributed Systems, vol.19, issue 5, pp.710-720, May 2008.

[17] J. Y. Choi, J. H. Lee, and Y. J. Chung, “Minimal Hop Count Path Routing

[18] R. Lijie, G. Zhongwen, and M. Renzhong. “Prolonging Sensor Network Lifetime with Initial Energy Level Assignment,” International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 231- 236, Aug. 2008.

[19] L. E. Ren, Z. W. Guo, and R. Z. Ma “Energy Assignment to Prolong the Lifetime in Heterogeneous Wireless Sensor Network,” the International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4057-4061, July 2008.

[20] S. Hong, Y. J. Choi and S. J. Kim, “An Energy Efficient Topology Control Protocol in Wireless Sensor Networks,” International Conference on Advanced Communication Technology, vol. 1, pp. 573-571, Feb. 2007.

[21] V. Enigo and V. Ramachandran, “An Energy Efficient Congestion Control Protocol for Wireless Sensor Networks,” IEEE International Advance Computing Conference, pp. 1573-1578, Mar. 2009.

[22] C. M. Huang, H. H. Ku and H. Y. Kung, “Efficient Power-consumption-based Load-sharing Topology Control Protocol for Harsh Environments in Wireless Sensor Networks,” Communication, Institution of Engineering and Technology, vol.3, pp. 859-870, 2009.

[23] M. Esseghir and N. Bouabdallah, “Node Density Control for Maximizing Wireless Sensor Network Lifetime,” ACM International Journal of Network Management, vol. 18, issue 2, pp. 159-170, Mar. 2008.

[24] R. Vidhyapriya and P. T. Vanathi. “Energy Aware Routing for Wireless Sensor Networks,” International Conference on Signal Processing, Communications and Networking, pp. 545-550, Feb. 2007.

[25] C. E. Perkins and E. M. Royer,” Ad-hoc on-demand distance vector routing,” the IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, Feb. 1999.

[26] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” Kluwer International Series in Engineering and Computer Science in Mobile Computing, vol. 353, pp. 153-181, Sept. 1996.

[27] C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers,” ACM SIGCOMM Computer Communication Review, vol. 24, issue 4, pp. 234-244, Oct. 1994.

[28] Y. Yoo and D. P. Agrawal, “Mobile Sensor Relocation to Prolong the Lifetime of Wireless Sensor Networks,” IEEE Vehicular Technology Conference, pp. 193-197, May 2008.

[29] Y. C. Wang, W. C. Peng and Y. C. Tseng, “Energy-Balanced Dispatch of Mobile Sensors in a Hybrid Wireless Sensor Network,” IEEE Transactions on Parallel and Distributed Systems, issue 99, PP. 1-14, 2010.

[30] Y. Liu, H. Zhu, K. Xu, and W. Teng, “An Improved Route Repair Approach of Wireless Sensor Networks,” the International Multi-Symposiums on Computer and Computational Sciences, vol. 2, pp. 662-665, June 2006.

[31] J. Luo and J. P. Hubaux. “Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks,” Joint Conference of the IEEE Computer and Communications Societies. vol. 3, pp. 1735-1746, Mar. 2005.

[32] M. Miller and N. Vaidya, “A MAC protocol to Reduce Sensor Network Energy Consumption Using a Wake-up Radio,” IEEE Transactions on Mobile Computing, vol. 4, pp. 228-242, May 2005.

[33] R. C. Shah, and H. M. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks,” IEEE of Wireless Communications and Networking

[34] J. Chang and L. Tassiualas, “Maximum Lifetime Routing in Wireless Sensor Networks,” IEEE/ACM Transactions on Networking (TON), vol. 12, issue 4, pp. 609-619, Aug. 2004.

[35] K. Treeprapin, A. Kanzaki, T. Hara, and S. Nishio,” On a Mobile Sensor Control Method for Uniform Sensing in Sparse Sensor Networks,” the International Conference on Mobile Data Management Workshops, pp.125-132, April 2008.

[36] F. EI-Moukaddem and E. Tomg, “Energy efficient routing in mobile wireless sensor networks,” Computer Science and Engineering, Michigan State University, East Lansing, Michigan, Tech. Rep. MSU-CSE-09-4, Feb. 2009.

[37] G. Wang, G. Cao, and T. L. Porta, “Movement-Assisted Sensor Deployment,” the Joint Conference of the IEEE Computer and Communications Societies, vol. 4, pp. 2469-2479, March 2004.

[38] Y. C. Wang, C. C. Hu, Y. C. Tseng; “Efficient Placement and Dispatch of Sensors in a Wireless Sensor Network,” IEEE Transactions on Mobile Computing, vol. 7, issue 2, pp. 262-274, Feb. 2008.