WEAR: a balanced, fault-tolerant, energy-aware routing protocol in WSNs

Kewei Sha

Wayne State University,
Detroit, MI 48202, USA
E-mail: kewei@wayne.edu

Junzhao Du

Xidian University,
Xi'an, Shaanxi 710071, PR China
E-mail: dujz@xidian.edu.cn

Weisong Shi*

Wayne State University,
Detroit,MI 48202, USA
E-mail: weisong@cs.wayne.edu
*Corresponding author

Abstract: As more and more real Wireless Sensor Network's (WSN) applications have been tested and deployed over the last five years, the research community of WSN realises that several issues need to be revisited from practical angles, such as reliability and security. In this paper, we address the reliability issue by designing a general energy-efficient, load balanced, fault-tolerant and scalable routing protocol. We first abstract four fundamental requirements of any practical routing protocol based on the intrinsic nature of WSN and argue that none of previous proposed routing protocols satisfies all of them at the same time. A novel general routing protocol called WEAR is then proposed to fill the gap by taking into consideration four factors that affect the routing policy, namely *the distance to the destination*, *the energy level of the sensor*, *the global location information* and *the local hole information*. Furthermore, to handle holes, which are a large space without active sensors caused by fault sensors, we propose a scalable, hole size-oblivious hole identification and maintenance protocol. Finally, our comprehensive simulation shows that, WEAR performs much better in comparing with GEAR and GPSR in terms of eight proposed performance metrics; especially, it extends the Lifetime of the Sensor Network (LSN) about 15% longer than that of GPSR.

Keywords: fault tolerant; energy aware; load balance; Wireless Sensor Networks (WSN); routing protocol.

Reference to this paper should be made as follows: Sha, K., Du, J. and Shi, W. (2006) 'WEAR: a balanced, fault-tolerant, energy-aware routing protocol in WSNs', *Int. J. Sensor Networks*, Vol. 1, Nos. 3/4, pp.156–168.

Biographical notes: Kewei Sha is a PhD student in the Department of Computer Science at Wayne State University. He received his BS from East China University of Science and Technology in 2001 and MS from Wayne State in 2005, both in Computer Science. His research interests include distributed systems and wireless sensor networks.

Junzhao Du is a Lecturer in Software College at Xidian University, PR China. His research interests include operating system, distributed systems and wireless sensor networks. He is also the Director of Postal Research and Development Lab at Xidian University and he has developed some mission critical applications used by China Post. This work was done when the author visited Wayne State University.

Weisong Shi is an Assistant Professor of Computer Science at Wayne State University. He received his BS from Xidian University in 1995 and PhD from the Chinese Academy of Sciences in 2000, both in Computer Engineering. His current research focuses on trusted resource sharing in peer-to-peer systems, mobile computing and wireless sensor networks. He has published more than 40 peer-reviewed journal and conference papers in these areas. He is the author of the book *Performance Optimization of Software Distributed Shared Memory Systems* (High Education Press, 2004). He has also served on technical programme committees of several international conferences, including the Chair of poster track of WWW 2005. He is a recipient of Microsoft Fellowship in 1999, the President outstanding award of the Chinese Academy of Sciences in 2000, one of 100 outstanding PhD dissertations (China) in 2002, 'Faculty Research Award' of Wayne State University in 2004 and 2005, the 'Best Paper Award' of ICWE'04 and IPDPS'05. He is a member of ACM, USENIX and IEEE.

1 Introduction

Recent advances in Wireless Sensor Networks (WSNs) research (Akyildiz et al., 2002; Crossbow Technology, Inc., http://www.xbow.com/; Estrin et al., 1999; 2002; 2003; Hill et al., 2000; Levis et al., 2003; Pottie and Kaiser, 2000) have enabled the proliferation of sensor network applications (Batalin et al., 2004; Simon et al., 2004; Szewczyk et al. 2004a,b; Xu et al., 2004; Zhang and Hou, 2004). Most recently deployed applications are small scale, where some simple routing protocols work very well. However, to realise the vision of a large-scale deployment in hostile places, the routing protocol must have the following features: energy-efficient, load balanced, fault-tolerant and scalable. We envision that to meet the diverse requirements of the ever increasing spectrum of WSN applications, researchers need to exploit practical, fault-tolerant routing protocols to build more robust WSN and extend their lifetime.

Routing protocols for WSN have been extensively studied (Akkaya and Younis, 2005) and can be classified as information exploiting routing (Intanagonwiwat et al., 2000), geographic routing (Karp and Kung, 2000; Yu et al., 2002), energy aware routing (Shah and Rabeay, 2002; Yu et al., 2002) and fault-tolerant routing (Deng et al., 2004). However, these protocols normally focus on achieving one goal or another so that they are not general enough for WSN applications. For example, the greedy geographic routing usually is energy-efficient but load imbalanced; data-centric routing is normally neither load-balanced nor fault-tolerant and most current fault-tolerant routing protocols are load imbalanced either. We argue that a good routing protocol for WSN should be designed by considering the characteristics of the sensor network and satisfying the basic requirements of a good routing protocol; however, previous routing protocols cannot achieve this goal.

In this paper, we design a general *energy-efficient*, *load balanced*, *fault-tolerant* and *scalable* routing protocol called WEAR. Firstly, we abstract four fundamental requirements of any practical routing protocol, based on the unique characteristics of WSN. Then, we propose a general routing protocol aiming to achieve multiple requirements of a good routing. The routing decision is based on a heuristic named *weight*, which is a combination of the four factors including *the distance to the destination*, *the energy level of the neighbour sensors*, *the global location information* and *the local hole information*. Hole is defined as a large space without active sensors, resulting from dead and/or fault sensors.

The contribution of this paper is fivefold: firstly, we are the first to provide a comprehensive analysis of the requirements of routing protocols in WSN, which serve as a general guide for future routing protocol design. Secondly, we propose a general routing protocol framework WEAR that takes energy-efficiency, load balance, fault tolerance and scalability into consideration. Thirdly, we design a hole size-oblivious protocol to identify and maintain the hole information in a dynamic sensor network environment. Fourthly, we define eight general performance metrics, which can be used as general metrics against which to systematically evaluate the routing protocols for WSN. Finally, we use a comprehensive simulation to demonstrate the efficiency and effectiveness of the WEAR protocol by comparing with GEAR (Yu et al., 2002) and GPSR (Karp and Kung, 2000) in terms of the proposed general performance metrics.

The rest of this paper is organised as follows. In Section 2, we abstract the requirements that a good routing protocol should satisfy, and list drawbacks of the previous work. The proposed WEAR protocol is depicted in Section 3, followed by the definition of the weight and heuristic mechanisms used in the routing in Section 4. The hole identification and maintenance protocols are described in Section 5. In Section 6, eight performance metrics are defined and the results from a comprehensive experiment are analysed and presented. Finally, we list the related work in Section 7 and draw the conclusion in Section 8.

2 Basic requirements of routing protocols

Unique characteristics of WSN, such as battery-backed power, failure-prone, less computing power and collaborative information processing, differ from the traditional TCP/IP network and ad hoc networks in general, especially on the requirements of routing protocol. In this section, we abstract four general requirements of any routing protocol in WSN as follows:

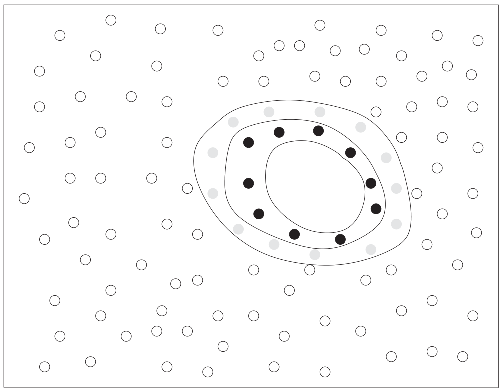
- *Energy efficient*: the most significant difference between the sensor network and the traditional network is the energy constraint of battery-backed sensors. And sensors sometimes are deployed in dangerous place so that it is impossible to recharge these sensors after the power is out. So Energy Consumption (EC) is a big consideration and energy-efficient routing protocol is required to extend the Lifetime of the Sensor Network (LSN).
- *Load balanced*: the main function of the sensor network is to collect interesting information from the monitoring field. Some applications like environment monitoring need the sensor network to run for a long time. So extending the LSN is an important goal of every routing protocol in the sensor network. As argued in Sha et al. (2004), imbalance load will deplete the energy of some sensors very quickly resulting in short LSN, and load balance plays a very important role in extending the LSN. So a good routing protocol should have a feature of load balance to extend the LSN.

- *Fault tolerant*: one characteristic of WSN is its high sensor failure probability either due to out of power or due to physical failure. In addition, it is very difficult to replace the failed sensors. After some sensors fail, there will be some holes in the sensor network which block the routing. Thus, the routing protocol for WSN should be fault tolerant, that is, the routing protocol should bypass the hole and prevent the hole enlarging.
- *Scalable*: another characteristic of WSN is its promising large-scale deployment, which may consist of thousands even ten thousands of sensors. As argued in Zhao and Guibas (2004), sensor networks are providing individual pieces of data. On the contrary, the novelty is that they provide data at a high density, over a large geographic extent. So the routing protocol for wireless sensor network should also be scalable. Localised (also known as decentralised) algorithm is the only way to address the scalability issues, as originally argued in Estrin et al. (1999), and recently advocated in Savvides et al. (2001) and Whitehouse et al. (2004). Thus, a good routing protocol should only use localised information to make it scalable.

From the above analysis, we find that a good routing protocol in WSN should be *energy-efficient*, *load balanced*, *fault-tolerant* and *scalable*. Next, we compare several typical previous routing protocols against these four requirements.

Before describing the details of our proposed protocol, we first check several existing routing protocols against these four requirements. Shortest path is a way to achieve the energy-efficiency. Xing et al. (2004) argue that the *greedy geographic routing* like GPSR is an attractive approach in WSN to achieve energy efficiency by using the shortest path and get scalability by only using the local information. However, it always takes the local shortest path so that it has a problem of depleting the energy of sensors on the shortest path, for example, by overexploiting the same shortest path along the hole to bypass a hole, GPSR enlarges the hole very quickly as shown in Figure 1. In the figure, the empty dots denote the alive sensors, and the dark dots represent the failed sensors. The smallest circle shows the shape of the initial hole. After handling some queries, sensors shown as black dots fail, the hole extends to the middle circle and further to the largest circle after grey sensors fail.

Figure 1 Hole enlargement using GPSR



Energy-aware routing allows sensors to take different paths based on the energy level of neighbour sensors. However, it will extend the path length a lot. *Multiple path routing* is used in multipath version of Directed Diffusion (Ganesan et al., 2002) to provide a kind of fault tolerance. However, it increases energy consumption a lot. Deng et al. (2004) use path repair to get fault tolerance, but they have not considered load balance. *Environment aware routing* is the best way to make routing protocol intelligent by exploring beneficial information and avoiding harmful information, for example, Directed Diffusion (Intanagonwiwat et al., 2000) uses the beneficial information, reinforced path, to achieve energy-efficient routing. However, the harmful information has not been used in the routing protocol yet.

3 WEAR protocol design

Having known the requirements of a good routing protocol for WSN, we are in a position to define a good routing protocol that satisfies these requirements. In this section, we propose WEAR, a Weighted Energy-Aware Routing protocol built upon several existing techniques. In the following, we describe the rationale behind our design first, then give a high-level description of the WEAR protocol, followed by a comparison with other protocols.

3.1 *Rationale of our design*

The rationale behind our design, that is, how to design a good protocol to satisfy all four requirements listed in Section 2, is listed below. First, we choose greedy geographic forwarding to achieve the goal of scalability because it only needs local information to make routing decision. And greedy geographic forwarding can also achieve energy-efficiency by taking the shortest path. The goal of load balance is achieved by using information exploiting routing which considers three factors, the global location information that shows the importance of the sensor in the sensor field, the energy level of sensors which denotes the EC history and the local hole information which records the hole information near the sensor. To get load balance, we prefer to route messages to sensors that are having higher remaining energy, less important and farther to the hole.

Fault-tolerant routing requires that the sensor network can still find a way from the source to the destination when there are some failure sensors and also control the enlargement of the hole. Our protocol achieves the goal of fault tolerance in two ways, bypassing the hole using right-hand rule proposed by Fang et al. (2004) and identifying the hole information and publishing it so that sensors can avoid routing message towards the hole.

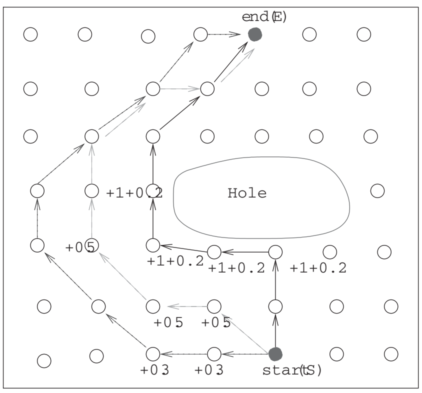
3.2 *The WEAR protocol: overview*

The basic idea of the WEAR protocol is to combine four factors, the distance to the destination, the energy level of sensors, the global location information and the local hole information, together to make the protocol a general routing protocol. The routing decision is made based on a heuristic named weight, which is the combination of the four factors. The detailed protocol is described below.

In the WEAR protocol, the routing has two modes, the greedy mode and the bypassing mode. In the greedy mode, there must be neighbours that are nearer to the destination than the current sensor. Then the current sensor forwards the message to the neighbour having the smallest weight value. If there is no neighbour closer to the destination than the current sensor, the routing enters the bypassing mode when the routing follows the right-hand rule proposed by Fang et al. (2004) to forward the message until the message reaches a node that is closer to the destination than the location where the bypassing mode starts.

The basic idea of WEAR is illustrated in Figure 2, which has a hole in the centre of the sensor field. In the figure, a message is sent from the start (S) to the end (E), and the numbers near the nodes represent the increase of the weight, for example, $+1 + 0.2$ means that a hole near the sensor increases the weight by 1, and EC increases the weight by 0.2. We can see that WEAR takes three different paths for the routing between the same pair of start and destination. At first, WEAR takes the path denoted by the solid line like in purely greedy geographic forwarding. After the hole information is known, the sensors' weight near the hole increases by 1.0, 0.5 or 0.3. And the energy decreasing of sensors on the black solid line further increases the value of weight. Secondly, WEAR follows the path represented by the dashed line. After the energy of the sensors on the dashed line decreases, weight of these nodes increases again. WEAR will automatically use the path shown in the dotted line. Thus, WEAR tries to avoid routing to the hole as well as distribute the load to the alternative paths. Compared with WEAR, GPSR only take the path depicted by the solid line because it always chooses the shortest path based on the distance to the destination and bypasses the hole when it gets stuck. GEAR may take part of the dashed path after it has some cost knowledge to the destination. However, the learned knowledge of GEAR is not as accurate as WEAR, so it cannot prevent routing messages along the hole. Furthermore, logically GEAR has to store information for each destination, which is not suitable to be used in memory constraint system like sensor networks. Thus, both GPSR and GEAR are not as effective as WEAR in load balance and hole enlargement control.

Figure 2 An example of the WEAR protocol



3.3 Discussion

WEAR is built on several assumptions. Firstly, location information is available by physical devices such as GPS (Bulusu et al., 2000) or topology discovery algorithm (Bulusu et al., 2001; Moore et al., 2004). Secondly, the sensors' location is stationary and the data sink is fixed. Thirdly, sensors can only communicate with the peers within the communication range and multi-hop routing is required to deliver the message. Fourthly, the neighbourhood information is obtained and maintained by other mechanisms, for example, SCORE (Al-Omari and Shi, 2005).

In terms of communication patterns in WSN, WEAR is most suitable to be used in the point-to-point routing, that is, unicast. However, it is also useful in other communication models, for example, if the query is an area-multicast, we will first route the query to the centre of the area by WEAR then propagate it by using constraint area flooding. How to combine WEAR with other communication models and in-network processing will be our future work. Compared with the pure greedy routing algorithm, WEAR may take longer path for individual routing, however, it will prolong the lifetime of the whole network, as validated in the following section.

From the above discussion, we can see that WEAR builds on top of two important notions: *weight* and *hole*. First, *weight* is a key to the success of WEAR. Also hole identification, propagation and maintenance are of primary concern as well. So in the next two sections we will formally define how to calculate the value of weight and propose a protocol on how to identify and maintain the hole.

4 Weight definition and calculation

Because weight is a key to make the decision during the routing, we are in a position to formally depict the definition of weight. As described in the last section, weight contains four factors, the distance to the destination, the energy level of neighbour sensors, the global location information and the local hole information. In this section, we will also give formal definition for each factor.

Definition of weight: *weight* is formally defined as

$$W_j = \alpha G_j + \beta L_j + \gamma RE_j + \lambda D_{jd}$$

where W_j is weight value of the jth sensor; L_j stands for the value of the local hole information of the jth sensor; G_j is the value of the global location information of the jth sensor; RE_j is the remaining energy of the jth sensor and D_{jd} is the distance between the jth sensor and the destination. α, β, γ and λ are four parameters denoting the significance of the four factors. Next, we give definition for each parameter separately.

The global location information G: as argued in Sha and Shi (2005) the nearer to the sink, the more important the sensor. So G is calculated as

$$G = c\frac{d_{\max} - d_{js}}{d_{\max}}$$

where c is a constant and d_{js} is the distance between the jth sensor and the sink. $d_{\max}$ is the maximum distance of any pair of nodes.

The Remaining Energy RE_j: now, we define the remaining energy of the sensor. That is simple normalised value of the initial energy minus the energy consumed during the message delivery:

$$\mathrm{RE}_j = \frac{E_0 - \mathrm{CE}_j}{E_0}$$

where E_0 is the initial energy of sensors. CE_j is the consumed energy of the sensor, which can be calculated using the model in Sha and Shi (2005).

The local hole information L_j: the local hole information is the measure of the influence of nearby holes. We think that the influence of the hole is related with the distance of the sensor to a hole and the space occupied by the hole. For the distance, the nearer to the hole, the more significant the influence is. We have two ways to measure the distance from the sensor to the hole. One is based on the hops of that sensor to the hole boundary, and the other is based on the geometric distance between the sensor and the hole centre. They are calculated by the following formula:

$$L_j = \sum_{h=0}^{n} A_h \left(1 - \frac{\mathrm{hops_i}}{\mathrm{Max_{hop}}}\right)$$

where L_j is the impact of the hole to the jth sensor; A_h is the area of the hole with ID h; $\mathrm{hops_i}$ means the number of hops between the sensor i and the hole boundary; $\mathrm{Max_{hop}}$ is the maximum number of hops the hole information will propagate; n is the total number of holes.

The distance to the destination D_{jd}: the factor D_{jd} is simply defined as the geometric distance between the jth sensor and the destination. D_{jd} is effective to the path length while G_j has contribution to the load balance. So both of them should be considered here.

As discussed just now, weight contains four components, the significance of which are implied by their corresponding coefficient parameters α, β, γ and λ. If we make any of them be zero, the corresponding component is no more considered, so WEAR can have variant configuration and applications can choose suitable configurations to satisfy their specific requirements. For example, if we make α, β and γ be zero, WEAR degrades to GPSR. If we make α and γ be zero, WEAR is similar to GEAR.

As we argue in the previous section, in the inborn distributed systems such as WSNs, the routing protocol should make the routing decision locally. In our routing protocol, the routing decision is made based on the value of weight, we show that it can be calculated locally. Firstly, the global location information and the distance to the destination are related with the relative distance to the sink and to the destination. In a location aware and location fixed sensor system, the global location information can be calculated locally based on the definition. Secondly, the local hole information can be easily calculated by the hole information calculation protocols, which is described in detail in Section 5. Thirdly, the value of the remaining energy is a totally localised concept. Thus, we find that the value of weight can be locally calculated, and our routing protocol can be easily applied in real deployment.

5 Hole information calculation

One of the important goals of WEAR is to bypass the hole and prevent the hole enlargement. To achieve this goal, we identify the hole and propagate the hole information to sensors near the hole first, then modify weight of sensors which results in different paths during the routing.

Before we give the detail of hole information calculation, we first give definition for hole information and define the hole size-oblivious message format we used in WEAR to record the hole information. Each hole has information of hole ID which uniquely identify the hole, hole owner who takes care of the hole updating, and the minimum and maximum x–y coordinates of hole boundary based on which we calculate the hole area.

A hole size-oblivious protocol is used to identify and maintain the hole information. Figure 3 shows a part of message format that is used in hole calculation. In the figure, the timer field is the time when the message is generated. The message originator field is the ID of the sensor that initialises the message. The field of hole owner records the maximum ID of sensors on the boundary of the hole. Max-x, max-y, min-x min-y fields are the maximum and minimum x–y coordinates of the hole boundary. Because the information is regardless of the number of sensors on the boundary of the hole, so the protocol is hole size-oblivious, which is very suitable to be implemented on the TinyOS platform. To calculate the hole information, we need to handle two processes, hole identification and hole maintenance, which are described in the following sections.

Figure 3 Hole related message format

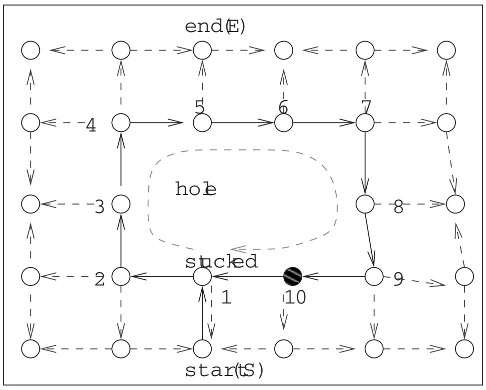
timer	message originator	hole owner	maxx	maxy	minx	miny	hole ID

5.1 Hole identification

The process of hole identification is started when the routing enters the bypass mode. The whole process consists of three steps, *hole locating*, *hole announcing* and *hole propagating*. The basic idea of the hole identification process is depicted in Figure 4, which has a hole in the centre caused possibly by running out of power or other reasons beyond our control. Next, the three steps of hole identification are described step by step. In order to make the following narration clear, we assume a message is sent from the start (S) to the end (E) as shown in Figure 4.

Step 1 *Hole locating*: the first step of this phase is to locate the hole in the sensor network, which is shown as the solid black lines in the figure. We use the hole locating algorithm proposed by Fang et al. (2004) to successfully locate the hole. First, when a message reaches a stuck node, it will generate a hole locating message, setting the value of maximum and minimum coordinator to its own coordinator and making itself be originator and owner. According to the right-hand rule described by Fang et al. (2004), the message will be routed around the edge of the hole. When a sensor on the edge of the hole receives this message, it will compare the ID with the owner ID. If its ID is larger, it will make its ID be the owner ID, at the same time update the value of max–min x–y coordinators. Then, the sensor forwards this message and record that it has seen this message. Several hops later, a sensor will see a message it previous has seen, which means that the hole has been located. Here we assume that the hole is covered by a rectangle, so we can easily calculate the area of the hole and the centre of the hole as its permeable ID based of the max–min x–y coordinate value. The solid black line in Figure 4 is an example of the identified hole with owner ID 10.

Figure 4 An example of hole identification



Step 2 *Hole announcing*: after locating the hole, we need to distribute the hole information to the sensors near the hole. First, we will tell the sensors on the edge of the hole the hole information then the hole on the edge can start hole propagation in a parallel way. The process informs the hole information to the sensors on the edge is called hole announcing as depicted by the dashed line insider the hole in Figure 4. In this step, the sensor which gets the complete hole information starts a hole announcing message containing necessary hole information. This message is routed around the edge of the hole and the sensors on the edge of the hole are aware of the complete information of the hole.

Step 3 *Hole propagating*: this step starts immediately after the hole announcing. After the hole announcing message arrives at the sensors on the edge of the hole, they will generate hole propagating message which contains the hole information such as the ID of the hole, the area of the hole and the hops to the hole edge. After the sensors not on the edge of the hole receives the hole propagating message, it will first check whether it is a new hole information either having new hole ID or having smaller hops value than current hops value for the same hole. If the information is new, it will update its local hole information list, increase the hops and forward it to its neighbours. Otherwise, it will get rid of that. The propagation will stop when the hops value reaches the maximum hops. The phase is denoted by the dotted grey line in Figure 4.

5.2 Hole maintenance

Due to the dynamic of the sensor network, the hole may enlarge or change the shape during the lifetime of the network. Therefore, maintaining the consistent hole information is also a big issue which should be handled. We think that the hole will change in two ways. Usually, holes in a sensor field will change in two styles: *hole enlargement* and *hole mergence*. The former case happens when some sensors near the hole fail, then the hole is enlarged. The hole mergence happens when some sensors fail, two or more holes will merge to a single hole. In this section, we describe the approach to handle the *hole enlargement* and *hole mergence*. Basically, the hole maintenance can be done in two ways, *periodical maintenance* and *reactive maintenance*. WEAR adopts a combination of both approaches.

5.2.1 Hole maintenance protocol

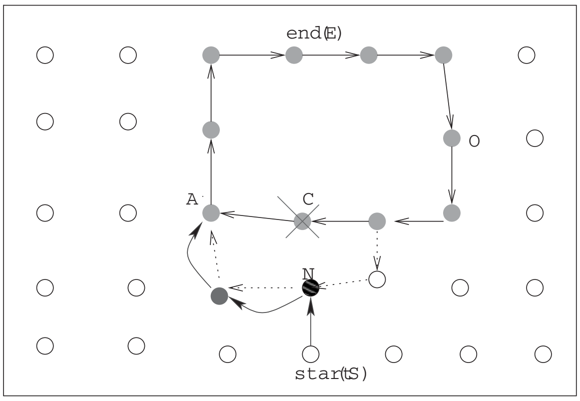
Periodical maintenance: in periodical maintenance, the hole information will be updated periodically. At the time of updating, the owner of the hole launches a hole reidentification process, which basically redoes the three steps in hole identification except that the hole ID is previously known. When other sensors receive the new hole information, it will update its value of the weight. If there are only hole enlargements, the sensor will only replace the old hole information by the new hole information, however, when there are some hole mergence, the sensor will remove the old hole information which is merged and add the new hole information. In other cases, when a sensor near the hole gets stuck between two periodical maintenance, it will launch a hole identification process. When the hole identification message arrives at a sensor which is on the edge of an old hole with the same stuck direction. It will generate a reply message containing the information of old hole to the new stuck sensor. Then, the stuck node gets the hole information. In this way, a lot of energy can be saved by not repeating the whole hole identification process; however, if the hole dynamics is high, this approach may have some problems because it may provide stale hole information.

Reactive maintenance: reactive maintenance is an alternative to the periodical maintenance. The difference between the reactive maintenance and periodical maintenance is that when a sensor on the edge of the hole receives a hole identification message, it will continue to forward the hole identification message to complete the whole process of hole reidentification. The advantage of this approach is that it can always keep the hole information fresh. However a lot of energy is consumed during the hole identification, especially when the sensor network's dynamic is very high. So it is only suitable when the hole changing rate is low. Because the WEAR protocol will not cause too much hole enlargement and mergence, we will take this approach in our simulation.

5.2.2 Hole enlargement

Hole enlargement is caused by the new failure sensors near the hole. In our approach, we use reactive maintenance. An example of hole enlargement is depicted in Figure 5. In the figure, shadow nodes connected by the solid black lines denote the old hole. When the node C with the cross-line fails, if the start S sends the message to the destination, the black node N is a new stuck node. So it will start a hole identification process. When the hole identification message denoted by the lines with empty arrow reaches the node A which has the old hole information. A finds that it is an enlargement of an old hole, so it will attach the ID of the hole to the hole identification message and forward that message. Eventually the hole is reidentified as depicted by the dotted line in the figure. After the new hole information is propagated, the nearby sensors will update the value of the weight.

Figure 5 An example of hole maintenance



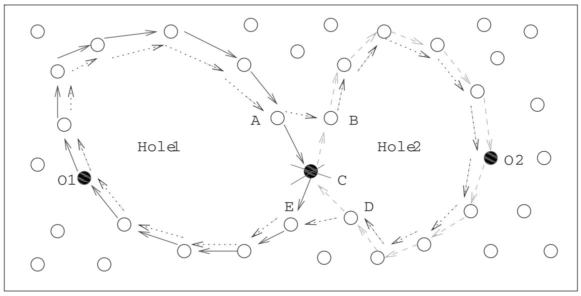
5.2.3 *Hole mergence*

Sometimes after some sensors located on the edge of the two or more holes fail, these holes may merge to one hole. When the hole merging appears, the new hole ID will become the combination of the IDs of the merged holes, and the hole information such as area will also be updated and propagated to the sensors near the hole. Figure 6 is an example of the hole mergence process. In the figure, there are two holes, hole 1 with owner O1 connected by the solid lines and hole 2 with owner O2 connected by the dashed lines. Sensor C is the connection of the two holes. Before the node C fails, the routing on the edge of the hole will not infer each other. After the sensor C fails, a hole mergence appears. Assume the owner of the hole 1 starts a hole update. The message will reach sensor A, which has hole 1's ID h1 and the hops information to hole 1 denoting it on the edge of the hole. Because the sensor C has failed, the message will route to sensor B, which contains the hole 1's ID h1 and the hops information to the hole 1 denoting that it is two hops to the hole 1, and it has hole 2's ID h2 and hops information to hole 2 denoting that it is on the edge of hole 2 as well. When B gets the message, it knows that there is a hole mergence because it is on the edge of hole 1 now. So it will combine the two holes ID, h1 and h2, as the ID of the new hole and forward the message. Eventually, the whole hole is identified as shown by the dotted lines in the figure. After the nearby sensors receive the new hole information, they will remove the old hole information with hole ID h1 and h2, then add the new hole information with ID h1, h2.

6 Performance evaluation

After describing the details of the WEAR protocol, we now turn our gear to the performance evaluation by focusing on how much does WEAR satisfying the four requirements abstracted in Section 2. Firstly, eight performance metrics are proposed as a general rule against which to compare different routing protocols. Based on these metrics, we systematically compare WEAR with two typical previous efforts, GPSR (Karp and Kung, 2000) and GEAR (Yu et al., 2002).

Figure 6 An example of hole mergence



6.1 *Performance metrics*

Although a large number of routing protocols have been proposed, most of them did the performance evaluation in an ad - hoc fashion using limited performance metrics. To remedy this situation, we propose and define eight general performance metrics to be used for evaluating the efficiency of any routing protocol.

Energy Consumption (EC): EC can provide a clear view of the energy efficiency feature of the routing protocol and EC of different individual sensors shows the load balance feature. EC of each sensor is simply defined as the normalised total amount of energy used in receiving or sending messages, that is, $\mathrm{EC} = E_c/E_0$, where E_c is the consumed energy and E_0 is the initial energy.

Lifetime of the Sensor Network (LSN): one of the most important goals of a routing protocol is to extend the LSN. LSN in this paper is defined as the maximum number of the queries successfully handled by the sensor network before its termination, which is defined as the moment the sensor network partitions or the Number of Failed Sensors (NFS) exceeds a predefined threshold. More detailed information is available in Sha and Shi (2005).

Load Imbalance Factor (LIF): load balance plays an important role in the LSN. LIF is used to quantitatively evaluate the load balance feature of the routing protocol, and it is formally defined as the variance of the remaining LSN, that is, $\mathrm{LIF} = \sum_{j=1}^{n}(E_j - E_{\mathrm{avg}})^2$, where n is the total number of sensors, E_j is the remaining lifetime of sensor j and E_{avg} is the average remaining lifetime of all sensors.

Number of Failed Sensors (NFS): imbalanced load will deplete the energy of some sensors with heavy load very quickly. NFS is the number of total failed sensors, which is another metrics defined to reflect the load balance of the routing protocol. If the load is balanced, the value of NFS should be small and the time when NFS becomes to be larger than zero will be large.

Path Length Extension Rate (PLER): to balance the load to different paths, the routing protocol may not always take the shortest path. Thus there is a tradeoff between the load balance and the path length. PLER is defined as how much the routing protocol extends the path length compared with GPSR. In one way, it is the ratio of the path length difference between other routing protocols and GPSR to the path length of GPSR; $\mathrm{PLER} = (\mathrm{PL}_{\mathrm{others}} - \mathrm{PL}_{\mathrm{GPSR}})/\mathrm{PL}_{\mathrm{GPSR}}$, where $\mathrm{PL}_{\mathrm{GPSR}}$ is the path length by using GPSR and

PL_{others} is the path length by other routing protocols. In the other way, PLER can be defined as the average extended path length in hops compared with GPSR, that is, $PLER = PL_{others} - PL_{GPSR}$. We will use both two definitions in the following sections.

Hole Extension (HEX): overexploiting sensors near the hole will enlarge the hole area. HEX is a metric to measure how fast the hole is enlarged and the dynamics of the total number of the hole, which reflects both load balance and fault tolerance of a routing protocol. In this paper, we assume sensors are evenly distributed in the sensor field, and $S_h = N_{fails} \times \rho$, where S_h is the area of the hole, N_{fails} is the NFS in the hole and ρ is the density of the hole, so HEX for each hole is defined as the NFS in the hole. From the view of the whole sensor network, HEX is defined as the number of holes in the sensor fields.

Query and Reply Successful Delivery Rate (QSDR, RSDR): queries or replies will get lost during the routing because of the failure of sensors or the path length exceeding the maximum path length allowed to deliver a message. This metrics is used to measure the successful rate of the message delivery. It is defined to show the fault tolerance of the routing protocol. We define QSDR and RSDR as the ratio of the number of successful delivered queries and replies to the number of the total number of expected query messages.

6.2 Simulation set-up

We decide to build a scalable WSN simulator by ourself because of the scalability concern of existing simulators, such as TOSSIM and NS-2 with wireless extension. All of them runs very slowly when the system scale is larger than 1000 nodes. The simulation is done in a large-scale discrete event-driven sensor network simulator, Capricorn (Sha et al., 2004), which reads topology information from an external topology generator, and simulates a package-level message delivery and the power consumption of each node. Note that the asymmetric and unreliable wireless link is not simulated in Capricorn, however, we believe that this feature is not a crucial factor to the proposed WEAR protocol and other protocols in this context.

In our simulation, 2012 sensors are scattered to a 1000×1000 m^2 sensor field, with some preset and random holes inside, for example, we preset holes centred in location (Deng et al. 2004; Estrin et al. 2002), (Deng et al., 2004; Szewczyk et al., 2004b), (Santi and Cheesa, 2002; Shah and Rabeay, 2002), and (Crossbow Technology, Inc., http://www.xbow.com/; Xu et al., 2004), as illustrated by the four cycles in Figure 7, where star nodes denote alive sensors and other nodes represent the area without sensors. Other simulation parameters are listed in Table 1, most of which are taken from the white papers from commercial products vendors. We implement three routing algorithms, GPSR, GEAR and WEAR in the simulator and simulate active query application. During each period of time, a query message is sent from the sink at the left-bottom corner to the destination, which is randomly chosen from the whole sensor ID space. After the destination gets the query message, a reply message is sent back to the sink. Thus in each round of query two messages, one query message and one reply message, are routed. In the following section, we simply use query denoting for the query message and reply denoting for the reply message. Because sensors near the sink depleted much faster than others, we increase the initial energy of sensors within three hops of the sink to several times as normal sensors, which can be done either by using some more powerful sensors or by replicating sensors located in that area. We believe this is a reasonable assumption for most real sensor deployment. The four parameters, α, β, γ and λ, for weight calculation, are set to -1, 2.0, -1 and 6. These values are chosen based on our experiences with other parameters setting during the test. How to choose suitable parameters in the definition of *Weight* is up to the specific requirement of different applications. For example, delay sensitive application will set the value of λ large, while the application that intends to control the hole enlargement will set the value of β large. How to fine tune the parameters intelligently will be our future work. In our implementation, GEAR uses parameters suggested by Yu et al. (2002). During simulation, only failure caused by out of power is simulated. We leave the effect of other failures to our future work.

Figure 7 An overview of the simulated sensor field, which consists of 2012 sensors deployed in a 1000×1000 m^2 with four predefined holes embedded inside

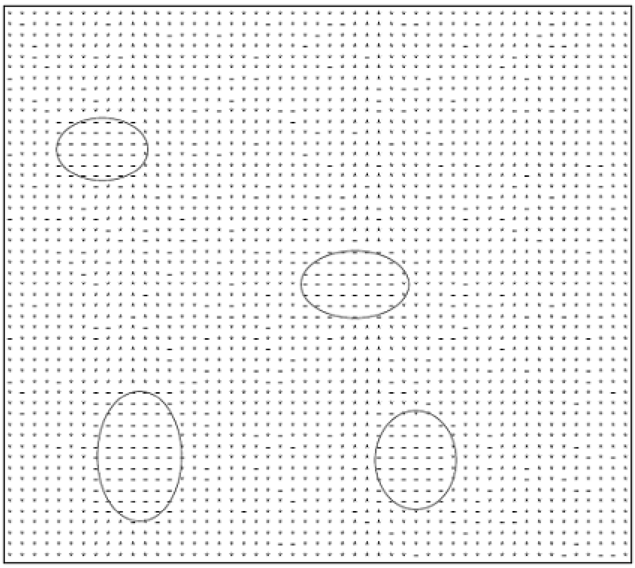


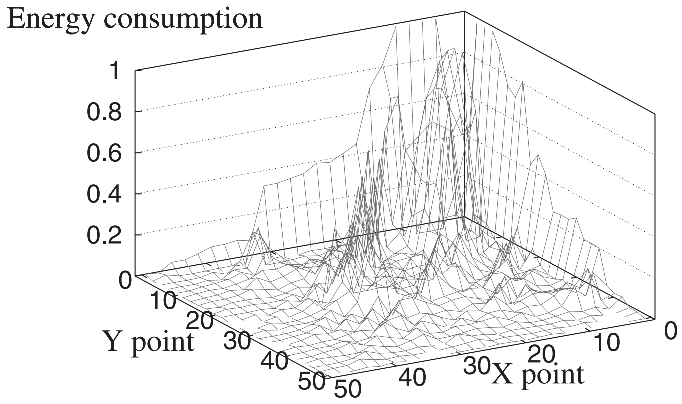
Table 1 Simulation parameters

Variables	*Values*
Communication range	30 m
Number of nodes	2012
Total energy of each sensor	1.725 Joules
Packet size	240, 1200 bits
Energy dissipated for receiving	50 nJ/bit
Energy dissipated for transmission	50 nJ/bit
Energy dissipated for transmit amplifier	100 pJ/bit/m^2
Bandwidth	40 kbps

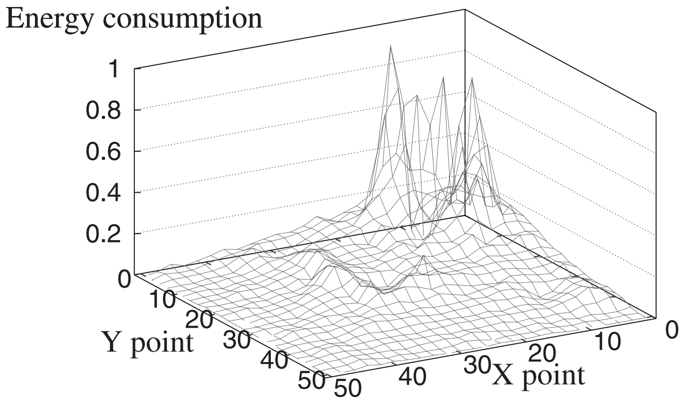
6.3 Energy consumption

Energy consumption is a metric system to measure the efficiency and the lifetime of WSN. Figure 8 shows a snapshot of EC by using three different routing protocols when 300 queries have been processed. In the figures, the x- and y-axis stand for the location of sensors and z-axis is the EC. We find that the energy consumption in GPSR is very imbalanced as shown in Figure 8(a). Sensors located on the edge of the sensor field and near the hole consume much more energy. In GEAR the EC is also not so balanced that some sensors consume more energy than others shown as the peaks in Figure 8(b), however, WEAR balances the EC a lot. From Figure 8(c), we can see that the EC distribution is related with the location of sensors, that is., sensors have the same distance to the sink consume similar energy.

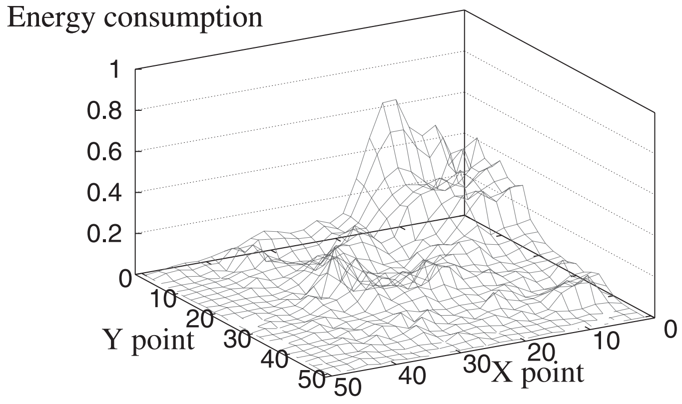
Figure 8 Snapshots of EC of three routing protocols: (a) GPSR, (b) GEAR and (c) WEAR



(a)



(b)

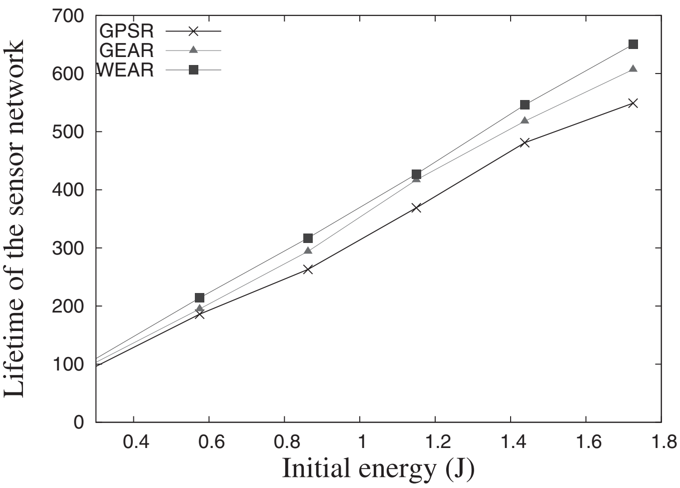


(c)

6.4 Sensor network lifetime

According to the definition of the LSN, LSN is the maximum successful queries that the sensor network can handle. The sensor network terminates when it divides or more than 100 sensors fail, when we think that the sensor network becomes to be useless with about 5% of the whole sensor failed. Figure 9 shows the comparison of LSN by using the three routing protocols. The x-axis is the value of the initial energy of normal sensors and the y-axis depicts the value of lifetime of the sensor network. From the figure, we can find that the lifetime increases with the increase of the initial energy of the sensors in all cases, but WEAR has about 5% and 15% longer lifetime than GEAR and GPSR, and the lifetime of the sensor network increases faster by using WEAR than others with the increase of the initial energy.

Figure 9 Comparison of the lifetime of sensor network



6.5 Effect of LIF

Our WEAR protocol aims to balance the load, which is reflected by the LIF. Based on our definition, the lower the LIF, the better the routing protocol. Figure 10 depicts the quantitative value of LIF by using the three routing protocols. In the figure, the x-axis means the number of processed queries and the y-axis is the value of load imbalance factor. We can observe that the load imbalance factor increases with the increase of the number of queries, however, WEAR has the smallest LIF among all the three and GPSR has almost two times larger LIF than the other two, which denotes that WEAR do balance the load during the routing by considering global location information, local hole information and the energy level of its neighbours.

6.6 Number of Failed Sensors

Due to the different LIF in the three routing protocols, the NFS in the sensor network varies. Figure 11 shows the relation of the NFS denoted as x-axis with the increase of the number of processed queries depicted as y-axis by using three routing protocols. From the figure, we can see that WEAR has smaller NFS than both GPSR and GEAR, for example, when 500 queries are sent out, the NFS in GPSR is doubled compared with that in WEAR, while WEAR is 50% better than GEAR. Both GEAR and WEAR delay the time when the first sensor fails compared with GPSR. For instance, no sensors fail before 100 queries have been processed by using GPSR and before 300 queries have been processed by using GEAR, but before 400 queries have been processed by using WEAR. Thus, WEAR delays the time when the first sensor fails for 100 and 300 queries than GEAR and GPSR separately.

Figure 10 Comparison of the LIF

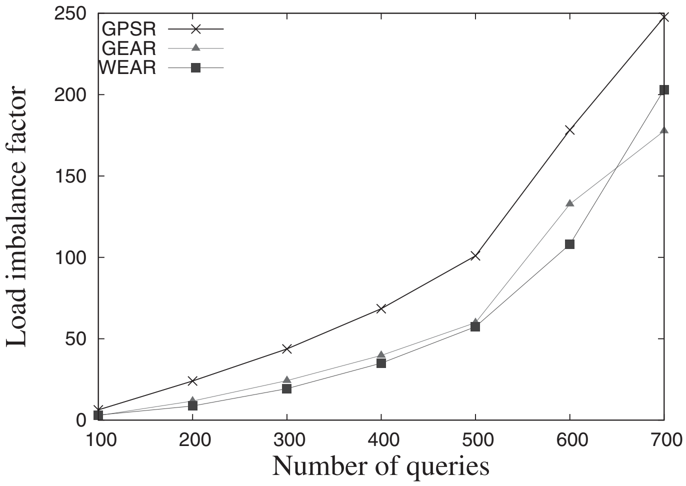
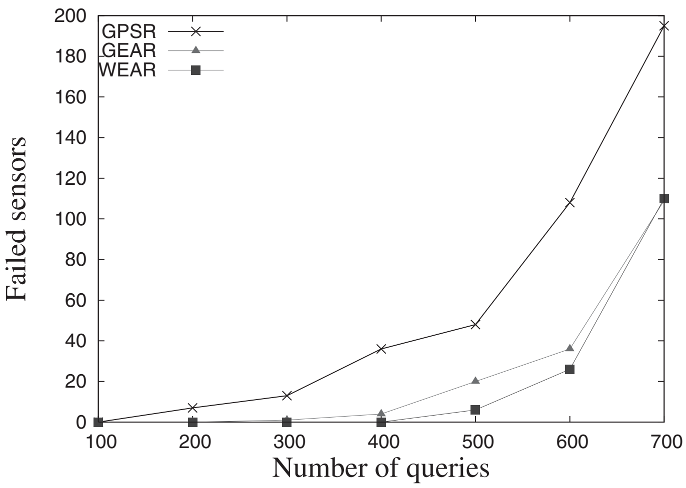


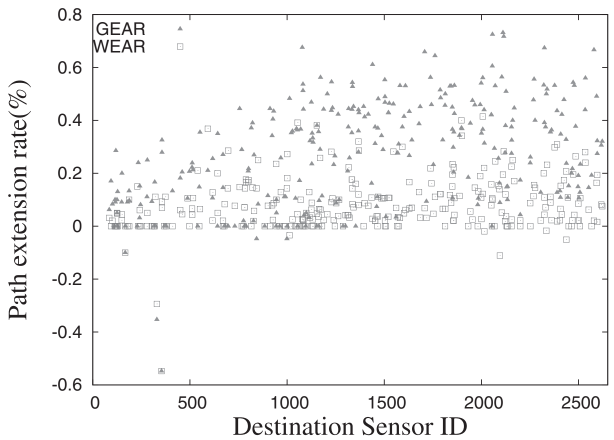
Figure 11 Comparison of the NFS



6.7 *Path length extension rate*

As defined in Section 6.1, the PLER is defined as a ratio of extended path length to the path length of GPSR. Figure 12 shows the path length extension rate of GEAR and WEAR. In the figure, the x-axis shows the ID of the destination sensors while the y-axis is the PLER. We can find that both WEAR and GEAR take longer path length to the destination in most cases because they take different paths to balance the load and avoid holes, but they sometimes take shorter paths as shown by the negative part occasionally because local greedy strategy of the GPSR sometimes does not guarantee the global path length optimisation. Compared with GEAR, WEAR is better, that is, WEAR extends the path length less than 10% at most cases while GEAR extends the path length normally more than 10%. In our experiments, WEAR on an average takes 1.8 more hops to deliver query and 2.4 more hops to route reply than GPSR while GEAR averagely takes 2.3 more hops in query path and 10.9 more hops in reply path than GPSR.

Figure 12 Comparison of the PLER of GEAR and WEAR to GPSR



6.8 *Hole extension*

Now we are in a position to examine the hole enlargement rate, that is, HEX, of different routing protocols. Because the sensors are evenly distributed in the sensor field with the same density, we calculate the number of failed sensors in the hole to approximate the hole area. Figure 13 is the hole area extension information using three routing protocols. The x-axis in the figure is the number of processed queries and the y-axis denotes the area of total holes. From the figure, we can see that the area of the holes enlarges very quickly by using GPSR, while it enlarges very slow using GEAR and WEAR. The total hole area caused by WEAR is a bit smaller than that of GEAR. GEAR and WEAR also control the hole enlargement before the first 400 queries as shown in Figure 13. On the other side, from the simulation, we also observe that the number of total holes changes very frequently in GPSR and finally it becomes to a very small number which means the mergence of holes while it changes slowly in GEAR and WEAR, which coincides with our goal to control the enlargement of holes.

6.9 *Query/Reply success delivery rate*

We use QSDRs and RSDRs to measure the fault tolerance of the three routing protocols. In our simulation, the messages will not be delivered when either its path length exceeds the maximum allowed path length (similar to the idea of TTL in the internet), or the sensor receiving the message is out of power before it forwards the message. We find that WEAR provides better QSDRs and RSDR as shown in Figures 14 and 15. In both the figures, the x-axis is the number of processed queries and the y-axis represents the rate of successfully delivered queries or replies. From these two figures, we find that replies have lower successful delivery rate than queries, which may result from the larger amount of EC of the reply causing more probability of depleting the energy of the sensor before it forwards the message. WEAR has a highly successful delivery rate that is always more than 96% in delivering queries and 93% in delivering replies, on the other hand, GPSR has the lowest successful delivery rate. Furthermore, similar to the first sensor failure time, WEAR delays the time when the successful delivery rate drops from 100% compared to GEAR and GPSR.

Figure 13 Comparison of the HEX

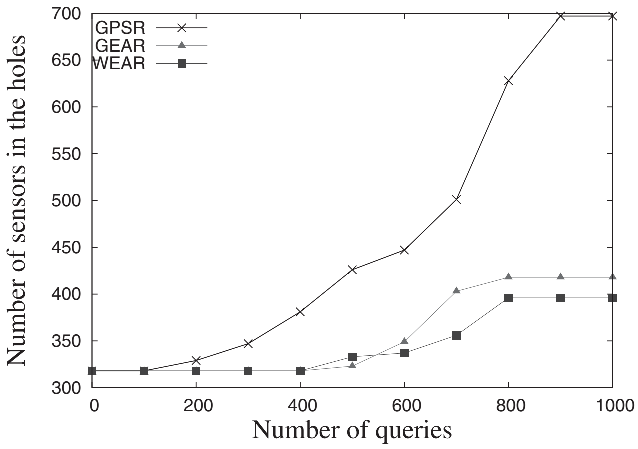


Figure 14 Comparison of the query successful delivery rate

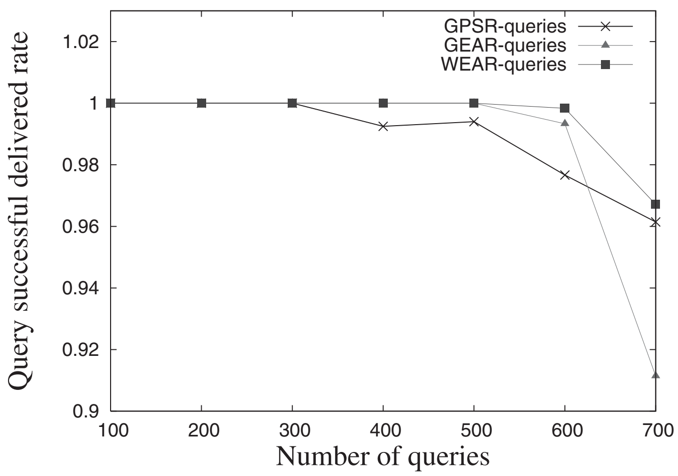
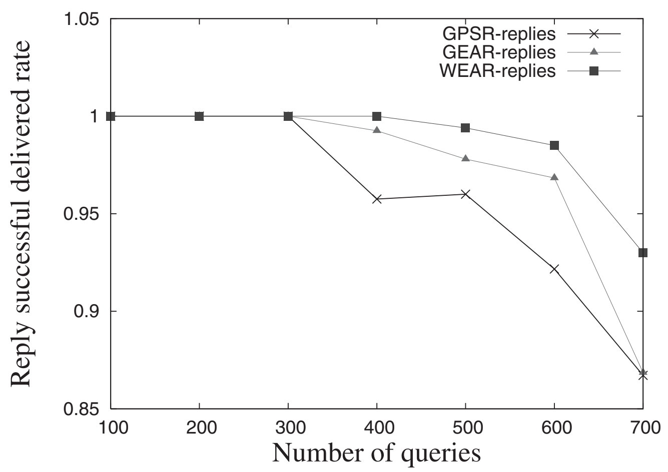


Figure 15 Comparison of the RSDR



From the above-mentioned sections, we conclude that WEAR extends the lifetime of WSN, decreases the LIF, increases the message successful delivery rate and controls the number of the failed sensor and the hole enlargement at the cost of extending the path length a little bit. Now we are in a position to examine the effect of different parameters in Section 6.10.

6.10 Effect of different parameters

As we described in Section 4, designing routing protocols in sensor network should consider the characteristics of the wireless sensor network as well as applications. It is up to applications to choose the suitable parameters to achieve some special goals. For example, if the application is very sensitive to the delay of the gathered information, it will choose the first set of parameters in Table 2, which takes the shortest path as GPSR. If the application wants to constrain the individual HEX, it can set a large value for β, which can prevent routing messages to the edge of the hole.

To evaluate the effect of different parameters on weight definition in Section 4, we intentionally set up five set of experiments as listed in Table 2. The first set of parameters only considers the distance to the destination so that WEAR performs like GPSR. In the second set, WEAR is similar to GEAR since it ignores the hole information and global location information. However, from the simulation, we find that it still performs better than GEAR. The third set of parameters does not consider the global location information, and the fourth one does not consider the distance to the destination. Finally, all the four components are taken into consideration. From the table, we see that the first set has the best PLEX, which means the shortest path, but it has worst LIF and NFS because it never consider load balance. For LSN, QSDR and RSDR, the fifth set performs best because it gets optimised from different aspects. The HEX value of the fifth set is also small. Comparing the second set with the fifth set in terms of NFS and HEX, the performance of the fifth set is a little worse, which shows that to keep each hole small may cause other sensors far to the hole die quickly. The fourth set performs bad generally, because the long path it takes results in significant energy inefficiency. Thus, we argue that the distance to the destination is a vital factor to the path length and LSN. In summary, the preliminary analysis in this section shows some interesting observations on the effect of these parameters. We plan to have a more comprehensive analysis in our future work.

7 Related work and discussions

WEAR builds on a large body of related work in general area of routing protocols of WSN. The interested reader is referred elsewhere (Akkaya and Younis, 2005) for good surveys of different techniques. Instead of describing each separately, we group related efforts into five broad categories: *greedy geographic routing*, *load balanced routing*, *energy aware routing*, *fault tolerant routing* and *information exploiting routing*.

Table 2 Comparison of different set of parameters

set ID	α	β	γ	λ	*LSN*	*LIF*	*NFS*	*PLER* (hops)	*QSDR*	*RSDR*	*HEX*
1	0	0	0	6	610	247.7	195	0	0.87	0.867	465
2	0	0	−1	6	664	194.4	108	1.93	0.948	0.937	384
3	0	2	−1	6	664	196.2	110	2.41	0.948	0.931	390
4	−1	2	−1	0	600	199.2	195	30.12	0.857	0.827	469
5	−1	2	−1	6	677	203.1	110	2.89	0.967	0.943	387

Greedy geographic routing: GEAR (Yu et al., 2002) and GPSR (Karp and Kung, 2000) are two greedy geographic routing protocols that are close to our work. Both of them have not considered the global information and the local hole information. Especially, GPSR is a purely greedy geographic routing protocol. Furthermore, the traffic concentrates on the perimeter of the sparser planar graph in the perimeter node using GPSR make node on planar graph depleted quickly. Thus, they are not so load balanced and fault tolerant.

Load balanced routing: several load balance protocols have been proposed in the literature. IQ in Sha et al. (2004) balances the load at the query level, while WEAR is working at lower level. Gao and Zhang propose a load balanced short path routing protocol in Gao and Zhang (2000), which argues load balance as well as greedy routing. However, it is only designed to be used in applications with sensors located in a narrow strip. Chang and Tassiulas (2004) uses a flow augmentation algorithm and a flow redirection algorithm to balance the energy consumption, while their method requires a full knowledge of traffic demands and cannot handle the network dynamics.

Energy-aware routing: GEAR is the closest approach to WEAR that learns neighbours' energy level in routing. Shah and Rabaey (2002) propose an energy aware protocol. They keep using a set of good paths at different time with some probability depending on the energy metric. Comparing with Shah and Rabaey (2002), WEAR is more general to consider more factors and keeps much less information. Younis et al. (2002) design an energy-aware routing for cluster-based sensor network, but the cluster-based scheme is argued to be energy inefficient.

Fault-tolerant routing. Gupta and Younis (2003) proposed a fault-tolerant clustering ; Santi and Chessa (2002) gives a fault-tolerant approach. Both of them tried to recover the detected faulty nodes, which is actually infeasible when WSN is deployed to a forbidden place. Another fault tolerant protocol by Datta (2003) is posted, but it is specific for a low-mobility and single-hop wireless network. Other work such as fault-tolerant data dissemination by Khanna et al. (2004) uses multi-path to provide the fault-tolerance, which has to keep more system states to achieve the goal.

Information exploiting routing: Data-centric routing such as Direct Diffusion (Intanagonwiwat et al., 2000) use interest to build the gradient and find a reinforced path to collect data. RUGGED by Faruque and Helmy (2004) direct routing by propagating the events information. However, all of them pervade useful information. On the contrary, WEAR distributes harmful hole information. Similar to WEAR, GEAR tries to learn the hole information. However, the hole information propagation is much faster and more sufficient in WEAR than that in GEAR. Furthermore, GEAR needs to keep a large amount of information for every destination.

8 Conclusions

In this paper, we analyse the requirements of the routing protocol in WSN and propose a general routing protocol framework to satisfy these requirements. Eight general performance metrics are proposed to evaluate the performance of the routing protocols in WSN. A comprehensive simulation has been conducted to compare WEAR with GEAR and GPSR in terms of these performance metrics. Simulation results show that WEAR is indeed a load balanced, fault-tolerant, energy-efficient routing protocol. The next step includes two directions:

1 further exploring the relationship between four components of the weight definition and extending WEAR to be adaptive according to the status of sensor network and

2 implementing WEAR in a real waste management wireless sensor network platform collaboratively developed at Wayne State University (Shi and Miller, 2004).

References

Akkaya, K. and Younis, M. (2005) 'A survey on routing protocols in wireless sensor networks', *Elsevier Ad Hoc Network Journal*, Vol. 3, pp.325–349.

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) 'A survey on sensor networks', *IEEE Communications Magazine*, Vol. 40, No. 8, pp.102–114.

Al-Omari, S. and Shi, W. (2005) 'Redundancy aware topology control in wireless sensor networks', *Technical Report MIST-TR-2005-012*, Wayne State University.

Batalin, M., et al. (2004) 'Call and responses: experiments in sampling the environment', *Proceedings of ACM SenSys 2004* November.

Bulusu, N., Estrin, D., Girod, L. and Heidemann, J. (2001) 'Scalable coordination for wireless sensor networks: self-configuring localization systems', *Proceedings of Internationsl Symposium on Communication Theory and Applications*, July.

Bulusu, N., Heidemann, J. and Estrin, D. (2000) 'GPS-less low cost outdoor localization for very small devices', *IEEE Personal Communications Magazine*, Vol. 7, No. 5, October.

Chang, J. and Tassiulas, L. (2004) 'Energy conserving routing in wireless ad-hoc networks', *INFOCOM00*, March.

Datta, A. (2003) 'Fault-tolerant and energy-efficient permutation routing protocol for wireless networks', *International Parallel and Distributed Processing Symposium (IPDPS'03)*, April.

Deng, J., Han, R. and Mishra, S. (2004) 'A robust and light-weight routing mechanism for wireless sensor networks', *First Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS 2004)*, June.

Estrin, D., Culler, D., Pister, K. and Sukhatme, G. (2002) 'Connecting the physical world with pervasive networks', *IEEE Pervasive Computing*, Vol. 1, No. 1, pp.59–69.

Estrin, D., et al. (1999) 'Next century challenges: scalable coordination in sensor networks', *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, August.

Estrin, D., Michener, W. and Bonito, G. (2003) 'Environmental cyberinfrastructure needs for distributed sensor networks', *A Report From a National Science Foundation Sponsored Worksop*, August.

Fang, Q., Gao, J. and Guibas, L.J. (2004) 'Locating and bypassing routing holes in sensor networks', *Proceedings of IEEE Conference on Computer Communications (INFOCOM'04)*, March.

Faruque, J. and Helmy, A. (2004) 'RUGGED: routing on fingerprint gradients in sensor networks', *Proceedings of IEEE International Conference on Pervasive Services*, July.

Ganesan, G., et al. (2002) 'Highly resilient, energy efficient multipath routing in wireless sensor networks', *Mobile Computing and Communications Review (MC2R)*, Vol. 1, No. 2.

Gao, J. and Zhang, L. (2004) 'Load balanced short path routing in wireless networks', *Proceedings of IEEE Conference on Computer Communications (INFOCOM'04)*, March.

Gupta, G. and Younis, M. (2003) 'Fault-tolerant clustering of wireless sensor networks', *Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2003)*, March.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. and Pister, K. (2000) 'System architecture directions for networked sensors', *Proceedings the Ninth ASPLOS'00*, pp.93–104, November.

Intanagonwiwat, C., Govindan, R. and Estrin, D. (2000) 'Directed diffusion: a scalable and robust communication paradigm for sensor networks', *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, August.

Karp, B. and Kung, H.T. (2000) 'Gpsr: greedy perimeter stateless routing for wireless networks', *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, August.

Khanna, G., Bagchi, S. and Wu, Y. (2004) 'Fault tolerant energy aware data dissemination protocol in sensor networks', *2004 International Conference on Dependable Systems and Networks (DSN'04)*, June.

Levis, P., Lee, N., Welsh, M. and Culler, D. (2003) 'Tossim: accurate and scalable simulation of entire tinyos applications', *Proceedings of the First ACM SenSys'03*, November.

Moore, D., Leonard, J., Rus, D. and Teller, S. (2004) 'Robust distributed network localization with noisy range measurements', *Proceedings of ACM SenSys 2004*, November.

Pottie, G. and Kaiser, W. (2000) 'Wireless integrated network sensors', *Communications of the ACM*, Vol. 43, No. 5, pp.51–58.

Santi, P. and Chessa, S. (2002) 'Crash faults identification in wireless sensor networks', *Computer Communications*, Vol. 25, No. 14, pp.1273–1282.

Savvides, A., Han, C. and Strivastava, M.B. (2001) 'Dynamic fine-grained localization in ad-hoc networks of sensors', *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01)*, July.

Sha, K. and Shi, W. (2005) 'Modeling the lifetime of wireless sensor networks', *Sensor Letters*, Vol. 3, pp.126–135.

Sha, K., Shi, W. and Sellamuthu, S. (2006) *Load Balanced Query Protocols for Wireless Sensor Networks*, Book chapter of Sensor Network Operations, S. Phoha, T.F. La Porta and C. Griffin (Eds). Wiley-IEEE Press, May.

Sha, K., Zhu, Z. and Shi, W. (2004) 'Capricorn: a large scale wireless sensor network simulator', *Technical Report MIST-TR-2004-001*, Wayne State University, January.

Shah, R. and Rabaey, J. (2002) 'Energy aware routing for low energy ad hoc sensor networks', *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'02)*, March.

Shi, W. and Miller, C. (2004) 'Waste containment system monitoring using wireless sensor networks', *Technical Report MIST-TR-2004-009*, Wayne State University, March.

Simon, G., Ledeczi, A. and Maroti, M. (2004) 'Sensor network-based countersniper system', *Proceedings of ACM SenSys 2004*, November.

Szewczyk, R., Mainwaring, A., Polastre, J. and Culler, D. (2004a) 'An analysis of a large scale habitat monitoring application', *Proceedings of ACM SenSys 2004*, November.

Szewczyk R., et al. (2004b) 'Habitat monitoring with sensor networks', *Communications of the ACM*, Vol. 47, No. 6, pp.34–40.

Whitehouse, K., Sharp, C., Culler, D. and Brewer, E. (2004) 'Hood: a neighborhood abstraction for sensor networks', *Proceedings of ACM MobiSys'04*, June.

Xing, G., Lu, C., Pless, R. and Huang, Q. (2004) 'On greedy geographic routing algorithms in sensing-covered networks', *Fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, May.

Xu N., et al. (2004) 'A wireless sensor network for structural monitoring', *Proceedings of ACM SenSys 2004*, November.

Younis, M., Youssef, M. and Arisha, K. (2002) 'Energy-aware routing in cluster-based sensor network', *Proceedings of ACM/IEEE MASCOTS'2002*, October.

Yu, Y., Estrin, D. and Govindan, R. (2002) 'Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks', *Technical Report Computer Science Department Technical Report UCLA/CSD-TR-01-0023*, UCLA, May.

Zhang, H. and Hou, J. (2004) 'On deriving the upper bound of α-lifetime for large sensor networks', *Proceedings of the Fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, May.

Zhao, F. and Guibas, L. (2004) *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann Publishers.