This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Towards Reliable Scheduling Schemes for Long-lived Replaceable Sensor Networks

Bin Tong*, Zi Li*, Guiling Wang**, and Wensheng Zhang*
*Department of Computer Science, Iowa State University
**Department of Computer Science, New Jersey Institute of Technology
Emails: {tongbin,wzhang,zili}@cs.iastate.edu, gwang@njit.edu

***Abstract*—To address energy constraint problem in sensor networks, node reclamation and replacement strategy has been proposed for networks accessible to human beings and robots. The major challenge in realizing the strategy is how to minimize the system maintenance cost, especially the frequency in replacing sensor nodes with limited number of backup nodes. New duty cycle scheduling schemes are required in order to address the challenge. Tong et al. have proposed a staircase-based scheme to address the problem based on ideal assumptions of sensor nodes that are free of failure and have regular energy consumption rate. Since sensor nodes are often deployed in outdoor unattended environment, node failures are inevitable. Energy consumption rates of sensor nodes are irregular due to manufacture or environmental reasons. Hence, this paper proposes several new schemes to achieve reliable scheduling for node reclamation and replacement. Extensive simulations have been conducted to verify that the proposed scheme is effective and efficient.**

I. INTRODUCTION

Power constraint has persisted as a big challenge for sensor network design, especially when the network needs to operate for long time. To meet the challenge, various approaches have been proposed, including energy conservation [1], ambient energy harvesting [2]–[6], incremental deployment, and node reclamation and replacement [7], [8].

Energy conservation schemes can slow down energy consumption and thus prolong network lifetime, but energy cannot be replenished. Energy may be harvested from ambient sources such as sunlight, however, mature technology suitable for tiny sensor nodes is still absent. For example, the amount of energy that a solar cell can harvest is proportional to its surface area, but it is infeasible to equip a tiny sensor node with a large-size solar cell, besides the amount of available solar energy also depends on uncontrollable conditions such as cloudiness of the sky. Incrementally deploying new sensor nodes to take over nodes running out of energy appears to another approach. However, this approach may be costly because sensor node hardware cannot be reused, and more importantly, it causes pollution to the environment because hardware and dead batteries are left in the environment.

A *node reclamation and replacement (NRR)* strategy [7] has recently been proposed to address the power constraint problem for sensor networks accessible to human beings or mobile robots. With this strategy, a robot or human labor called *mobile repairman (MR)* periodically reclaims sensor nodes of low or no energy supply, replaces them with fully-charged ones, and brings the reclaimed sensor nodes back to a place called *energy station (ES)* for recharging. In realizing the strategy, the major challenge is how to minimize the network maintenance cost. The maintenance cost mainly consists of two parts, the hardware cost and the maintenance labor cost. The hardware cost mainly includes the cost to purchase backup sensor nodes to replace nodes that need to be recharged. The maintenance labor cost is mainly reflected in the frequency that the MR should be dispatched to perform reclamation and replacement. The number of backup sensor nodes is usually limited and recharging nodes that have been replaced takes nontrivial time. Given these constraints, how to minimize the maintenance frequency of the MR poses as a difficult problem. Conventionally, duty cycles of sensor nodes are scheduled in a balanced manner such that all nodes die at the similar time. If this philosophy is still applied, the MR is required to use a limited number of backup nodes to replace nearly all nodes within a short time period, which is an impossible mission. Hence, new protocols for scheduling duty cycles are demanded.

It is ideal that the duty cycles of sensor nodes are scheduled appropriately such that, every certain time interval, only a subset of sensor nodes with the same number as backup nodes is to deplete power and needs replacement, and the time interval should be longer than the time needed to recharge all sensor nodes that have been replaced. This way, as every time sensor nodes that need to be replaced are no more than backup nodes, they can all be replaced and hence the lifetime of the network can be maintained. As the time interval for replacing two sets of nodes is longer than the time needed to recharge sensor nodes that have been replaced, it is guaranteed that there is always enough number of fully-charged backup nodes when replacement is needed.

A *staircase-based* scheme [8] has been proposed to realize the above idea, under the assumption that any sensor node, if active, consumes energy at the same rate, and there is no sensor node failure. The scheme schedules the duty cycles of sensor nodes in a well-planned manner. Ideally, at any moment, all sensor nodes in the network form a staircase according to the amount of their residual energy. Sensor nodes with the lowest level of residual energy is in the lowest layer of the staircase, those with the second to the least level of residual energy is in the second to the lowest layer, and so on and so forth. The difference between any two adjacent levels of residual energy is constant, and the time for a node to consume it is larger than the time to recharge nodes replaced.

Although the staircase-based scheme has been shown to achieve a good performance on minimizing the frequency of maintenance service performed by the MR and meanwhile achieving area coverage [9]–[11], it does not consider two important reliability issues: (i) sensor node failures, and (ii) irregular energy consumption rate among sensor nodes. Either of them can destroy the staircase structure and compromise its performance. In this paper, we propose three schemes, namely, the staircase repairing, debit/credit scheme, and the energy consumption balancing scheme, to address these issues and to improve the reliability of scheduling. The staircase repairing

978-1-4244-5837-0/10/$26.00

scheme, and debit/credit scheme are primarily designed to handle sensor failures, while the energy consumption balancing scheme is primarily designed to handle irregular energy consumption rate. Extensive simulations have been conducted to verify the effectiveness and efficiency of the schemes.

The rest of the paper is organized as follows: Section II describes the system model. An overview of the proposed scheme is presented in Section III, which is followed by the detailed description in Section IV. Section V reports simulation results, and finally Section VI concludes the paper.

II. SYSTEM MODEL

We consider that a network of n sensors, denoted as $s_1, s_2, s_3, \cdots, s_n$, is deployed to a continuous field for long-term monitoring. The monitored field is divided into m small areas, denoted as $a_1, a_2, a_3, \cdots, a_m$, such that, within any area a_i, the required sensing coverage level is the same at any point of the area.

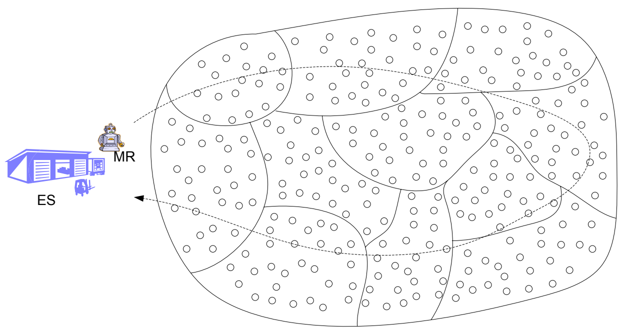


Fig. 1. System architecture

As shown in Fig. 1, the whole NRR system is composed of an *energy station (ES)*, a *mobile repairman (MR)*, and a sensor network. The ES stores a certain number (denoted as x) of backup sensors, and can recharge energy to sensors. The MR can be a human technician or a mobile robot. The MR can traverse the sensor network, reclaiming sensors of no or low energy, replacing them with fully-charged ones, and bringing the reclaimed ones back to the ES for recharging. Other assumptions of the system are as follows:

- All sensors are time synchronized. Time is divided into phases. A phase is a basic scheduling unit for duty-cycle scheduling; i.e., a sensor will not change its mode (active or sleeping) during a phase.
- Each sensor node knows its location.
- A sensor has two modes: *active* and *sleeping*. For every phase, if a sensor is in the active mode, its amount of energy consumption per phase complies to certain distribution, with the mean denoted as α_{mean}. If it is in the sleeping mode, its energy is unchanged. Let the energy of a fully-charged sensor be e. If a sensor is in the active mode all the time and the energy consumption per phase is always α_{mean}, its lifetime is denoted as T.
- For each area, the required sensing coverage level varies from N_{min} to N_{max}, subject to certain (e.g., Gaussian) distribution.
- Each area is deployed with $N_{max} + N_{back}$ (N_{back} is an integer greater than or equal to 1) *disjoint* sets of sensors, where each set of sensors can completely cover the area. That is, every point in the area can be covered by at least one sensor in each of the sets. We call these sets *coverage sets*. The reason for having more than N_{max} sets of sensors is to avoid service disruption at the time of node reclamation and replacement (Note: node reclamation and replacement cannot be completed in non-negligible time; hence, reclamation and replacement will inevitably disrupt the working of nodes that are reclaimed or newly placed).
- The system has x backup sensor nodes. The MR has orientation and localization ability such that it can travel to designated locales and perform sensor replacement task. We assume that the MR is able to carry x sensors at a time.
- Charging a sensor at the ES takes *non-negligible* time, which is denoted as τ. Note that, sensors can be recharged in parallel, we assume that it is possible to recharge all x backup sensors managed by the ES at the same time.

Design Goal. In this paper, we aim to design a collaborative scheduling scheme for sensors and the node reclamation and replacement algorithm for the ES/MR, such that (i) the sensor network can maintain the required area coverage for an infinite period of time, and (ii) the number of travels the MR should take is as small as possible (i.e., the average interval between two consecutive replacement trips is as large as possible).

III. PRELIMINARY: BASICS OF THE STAIRCASE-BASED SCHEME

To achieve guaranteed area coverage for an infinite period of time, two necessary tasks should be performed: firstly, sensors should collaboratively schedule their duty-cycles to achieve required area coverage; secondly, sensors and the ES/MR should coordinate to replenish energy into the network through node reclamation and replacement.

The staircase-based scheme in [8] includes three modules: the duty-cycling module, the sensor-ES interaction module, the node reclamation and replacement module.

Duty-Cycle Scheduling Module. This module runs on the sensor side. It mainly decides which sensors should be active and which ones should sleep for the current phase. Here, sensors in each area are grouped into disjoint coverage sets in which sensors can collaboratively cover the whole area. Coverage set is the atomic unit for sensor scheduling, which means all sensor in the same set at any time have the same active or sleep status. In other words, they always have the same remaining energy. In the scheme, since each sensor has the complete knowledge about all other sensors in the same area such as the remaining energy and which coverage sets they belong to, their views about the area are always consistent.

At the beginning of a phase, all sensor nodes wake up to participate in the scheduling. One sensor node will broadcast the coverage requirement (called coverage number) for the current phase. Different strategies can be utilized to select a node to broadcast the number. For example, the active sensor with the smallest ID in the last phase could be a reasonable choice. The surveillance number is application dependent and out of the scope of this paper. Intuitively, if some events that the system is interested in happens, the number could increase. Otherwise, it goes down. Once receiving the number, each sensor can run our proposed scheme to determine its status for the current phase. Due to the consistent view about the area, they can always reach the same scheduling decisions.

Interactions between Sensors and the ES Module. This module runs on the sensor side. It mainly sends out requests to the ES asking for reclamation. Two types of coverage sets exist in the scheme: primary set and backup set. As the name denoted, normally only the former could be in the active status. Shortly before a primary set depletes its energy, the work is shifted to a backup set. After the handoff, the primary set becomes a backup set waiting for replacement while the original backup one becomes a primary set and starts to work. Then, a *ready* message with format

$$ready\langle a, cs_i, c\rangle,$$

is routed to the ES. Again, the active sensor with the smallest ID could be responsible for sending such message out. Here, a is the ID of the area, cs_i is the ID of the coverage set needing to be reclaimed, and c is the total number of sensors in set cs_i. Once there is no more backup set to take over the work of the energy-depleted primary set, a *deadline* message is generated and sent to the ES with format

$$deadline\langle a\rangle,$$

where a is the ID of the area.

Node Reclamation and Replacement Module. This module runs in the ES side. It mainly determines how to deal with the received ready and deadline messages and when to send out the MR to do the replacement. If a ready message reaches ES and the sensor number in the to-serve replacement request list plus c in the message exceed x, then MR is dispatched to serve the earliest x requests, all of which are then removed from the list. If the sum does not reach x, the message is simply appended to the to-serve list. Upon receiving a deadline message, the MR goes out to serve all requests in the list plus the one in the deadline message. The list is emptied after that.

An Example: Suppose the network has m areas, and in each area, the coverage number is always N_{max}, i.e., all N_{max} primary coverage sets need to be active at any phase. We also assume that the amount of energy consumption per phase of every sensor node is α_{mean}.

For each area a_i, $1 \leq i \leq m$, its all N_{max} primary coverage sets form a "staircase", and the height of each stair is

$$e/N_{max},$$

where e is the amount of full energy of a sensor.

The network is shown in Fig. 2, in which each row shows the snapshot of remaining energy level of each coverage set in each area at a certain time instance. The network has $N_{max} = 4$ primary coverage sets and $N_{back} = 1$ backup coverage set. The remaining energy level of the four primary coverage sets forms a staircase with a stair height of $e/4$. In other words, if we sort the primary coverages according to their remaining energy level, any two adjacent coverage sets will have their remaining energy differing by $e/4$.

Our scheme defines an order in which areas are visited by the MR to reclaim and replace sensors in these areas. In Fig. 2, areas are sorted as a_1, a_2, a_3, a_4. For any two areas that are to be visited consecutively, their staircases have a phase difference δ, where δ and the height of a stair have the following relation:

$$e/N_{max} = m\delta, \tag{1}$$

In Fig. 2, at any time instance, the staircase of primary coverage sets in a_2 is $e/16$ higher than that of the primary coverage sets in a_1, the staircase of the primary coverage sets in a_3 is also $e/16$ higher than that of the primary coverage sets in a_2, and so on.

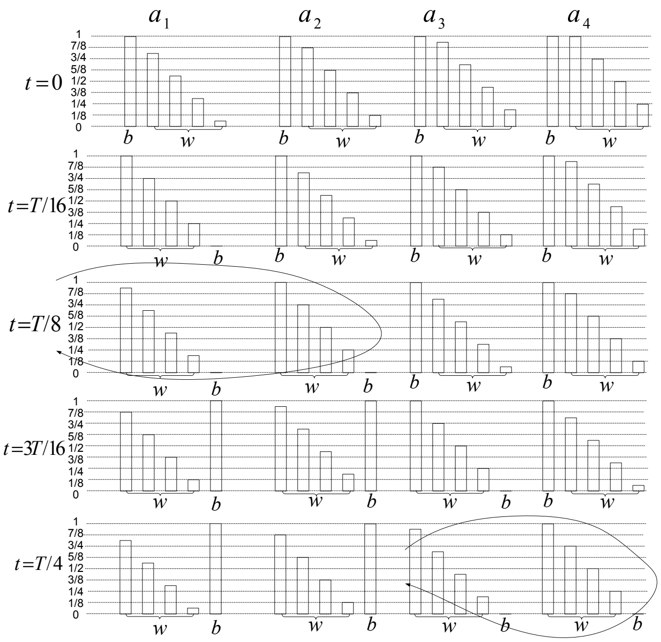


Fig. 2. Example 1: duty-cycle scheduling. Each bar represents a coverage set. $N_{max} = 4$, $N_{back} = 1$, $m = 4$, and $x = 32$. Each coverage set in every area has 16 sensors. "w" means primary set, and "b" means backup set.

Since the coverage requirement is always N_{max}, all the four primary coverage sets will be active at any time. In each area, after every $T/16$, where T is a sensor lifetime if being active all the time, a coverage set will use up it energy due to the staircase structure. Furthermore, due to the the phase difference δ, each area will use up its coverage set at different time instances, and these time instances are evenly distributed in a time interval of $e/16$.

When an area has used up a primary coverage set, a *ready* message will be sent to the ES to let it know that the coverage set is ready to be replaced. The primary set will select a backup coverage set with full energy, and shift its duty to the coverage set. Then the primary set becomes a backup set, and the selected backup set becomes a primary set.

In Fig. 2, at time $t = T/16$, a primary coverage set in area a_1 drains of its energy, and shifts its duty to the only backup coverage set. A ready message is also sent to the ES. Since at this time, the total number of nodes that are ready to be replaced is 16, which is less than $x = 32$, the MR will wait. At time $t = T/8$, a primary coverage set in area a_2 drains of its energy, and shifts its duty to the backup coverage set. A ready message is also sent to the ES. At this time, the total number of nodes that are ready to be replaced equals to x. Thus, the MR makes a replacement tour, replacing nodes in the backup sets of a_1 and a_2. Similarly, the MR makes another replacement tour at $t = T/4$, replacing nodes in the backup coverage sets of a_3 and a_4.

A. Other Scenarios

In the above, we assume that the staircase structure is already formed. However, when a sensor network starts operating, all sensors in the sensing field have full energy. Thus we need to form the staircase structure at the beginning. Furthermore, coverage numbers vary between N_{min} and N_{max} in general. The schemes to work in these scenarios were proposed in [8].

IV. Proposed Schemes

We propose three schemes besides a naive scheme to cope with sensor failure and irregular energy consumption rates.

A. Sensor Failure Detection

In our scheme, when a sensor fails, its 1-hop neighbors are responsible for detecting it. Specifically, at the beginning of a phase, if a node u is supposed to be active in a phase, it broadcasts a *on-duty* message to its 1-hop neighbors. This message can be broadcast several times to make sure all u's neighbors receive it. If u's neighbors do not receive this message, they consider that u has failed. In this case, a *failure* message is broadcast to the whole area, and all sensors in the area know the information of the failed sensor, including its ID, and the ID of its coverage set.

B. Naive Scheme

The original staircase-based scheme does not consider irregular energy consumption rate and sensor failures. The scheme can be extended as follows to consider reliability issues:

(i) Whenever a sensor node in a coverage set is drained of energy, a *ready* or *deadline* message is sent to the ES. Note that what message to be sent depends on whether there are full-energy backup coverage sets. The entire coverage set then becomes a backup coverage set, and is to be replaced as a whole later.

(ii) Whenever a sensor node fails, we treat it as being drained of all energy. The coverage set that the failed node belongs to becomes a backup set, and is to be replaced later.

The problem with this naive solution is that when the first sensor in a coverage set dies or fails, ready and deadline messages are sent out with irregular interval, which may corrupt the staircase structure.

Fig. 3 shows an example when using the naive scheme to deal with failures. Fig. 3(a) shows a normal case when there is no failure in an area. As can be seen, the area sends out a ready message every $T/4$. If we sort the primary coverage sets according to their remaining energy, any two adjacent primary coverage sets have their remaining energy differing by $e/4$.

Fig. 3(b) shows a case when a sensor node in coverage set 3 fails at $t = T/8$. In the naive scheme, this coverage set becomes a backup coverage set, and coverage set 0, which is a backup coverage set with full energy, becomes a primary coverage set. Note that coverage set 0 starts to work $T/8$ earlier than in Fig. 3(a). Furthermore, the difference in remaining energy between coverage sets 0 and 1 is $T/8$, and the difference in remaining energy between coverage sets 2 and 4 is $T/4$. The staircase structure is deformed at this time, i.e., ready messages are sent out at intervals $\{T/2, T/4, T/8, T/8\}$, not at the fixed interval of $T/4$ as in Fig. 3(a). At time instance $t = T/4$, since coverage set 3 has not been replaced, a deadline message is sent out, and both coverage sets 3 and 4 will be replaced. As can be seen, the pattern of intervals for sending out ready messages remains, which causes deadline messages to be sent out again and again in future.

Since when the MR performs a replacement in response to a deadline message, it may not carry all x batteries, the MR has to performs more replacements to avoid service disruption of the network.

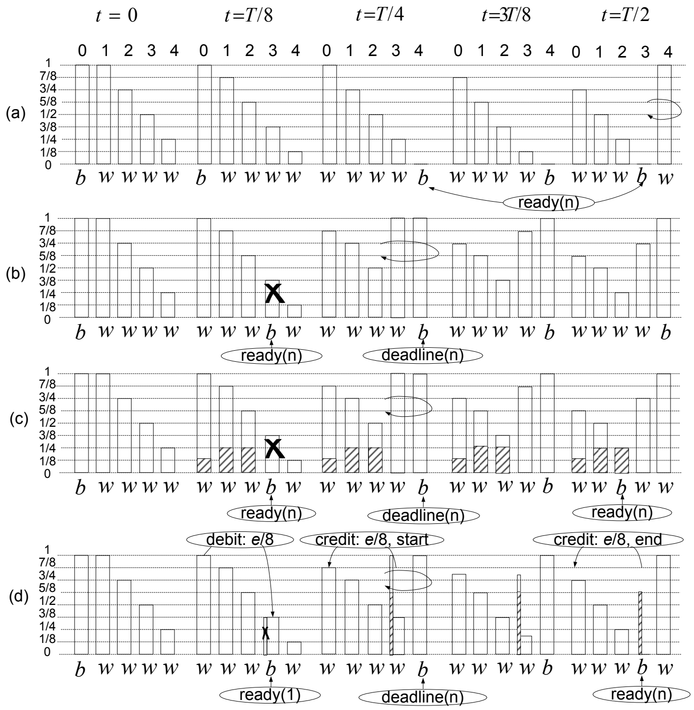


Fig. 3. Failure Example for an area. Each bar represents a coverage set. $N_{max} = 4$, $N_{backup} = 1$. Each coverage set has n sensors. When there is no deadline message in the system, the area will be visited by the MR every $T/4$. Each bar under a U-turn arrow represents a sensor which is just replaced. The parameter in ready/deadline messages is the number of sensors that need to be replaced concerning this message. (a) Failure-free case: the area sends out an $ready(n)$ message every $T/4$. (b) Naive scheme for failure. (c) Staircase repairing scheme. Shaded part of remaining energy will not be used. (d) Debit/credit scheme. Shaded part of remaining energy will not be used.

C. Staircase Repairing Scheme

The basic idea for staircase repairing scheme is that when the staircase structure is deformed due to failures, we repair the structure, such that each area still sends out ready messages at fixed intervals as before the failure. Specifically, when failure happens, we "reduce" the remaining energy of some stairs, such that if we sort the primary coverage sets according to their remaining energy level, any two adjacent coverage sets still keep their remaining energy differing by e/N_{max}. Note that we do not physically reduce remaining energy of a coverage set, instead we replace it earlier.

Fig. 3(c) shows an example. When a sensor node in coverage set 3 fails at $t = T/8$, coverage set 3 becomes a backup coverage set, and coverage set 0 becomes a primary coverage set. Then we "reduce" coverage set 0's remaining energy by $T/8$. This is done by sending out ready message when coverage set 0's remaining energy drops to $e/8$, instead of 0. As a result, the $e/8$ energy in coverage set 3 will not be used, and this part of remaining energy is shaded in Fig. 3(c). We further reduce coverage sets 1 and 2's energy by $e/4$, not $e/8$. Now if we sort the primary coverage sets 0,1,2, and 4 according to their remaining energy, any two adjacent coverage sets will keep their remaining energy differing by $T/4$.

At time instance $t = T/4$, a deadline message will be sent out, since coverage 3 has not been replaced. Both coverage sets 3 and 4 will be replaced at the time. After that, the area will send out a ready message every $T/4$, which is same to the normal case in Fig. 3(a). The staircase repairing algorithm is described in Alg. 1.

Algorithm 1 Staircase repairing algorithm: for sensor node u

```
Notations:
    $h[N_{max}]$: array that records the height of each stair $i, 0 \leq i \leq N_{max}$
    $stair(u)$: node $u$'s stair
    $bottom(u)$: if $u$'s remaining energy is lower than $bottom(u)$, then it is ready to be replaced
    $e(u)$: $u$'s remaining energy
Initialization:
    $bottom(u) \leftarrow 0$
Upon receipt of a failure message: $failure\langle v, stair(v)\rangle$
    if $stair(u) > stair(v)$ then
        if $stair(u) = N_{max}$ then
            $bottom(u) \leftarrow bottom(u) + h[0]$
        else
            $bottom(u) \leftarrow bottom(u) + e/N_{max}$
At the beginning of each phase
    if $e(u) < bottom(u) + \alpha_{mean}$ then
        if there is a backup set with full energy then
            Shift its duty to the backup set
            Send out a ready message
        else
            Send out a deadline message
        $u$ changes its role to backup
```

D. Debit/Credit Scheme

In the staircase repairing scheme, when a coverage set has a failed node, the whole coverage set are replaced. On the other hand, the debit/credit scheme only requests for replacing the failed node at the time of failure. The scheme adopts the notions of debit and credit from banking systems. Specifically, when a coverage set has a failed sensor, it becomes a backup coverage set, and another backup coverage set with full energy becomes a primary coverage set. This new primary coverage set starts to work earlier than expected because of the failure, and its remaining energy will be lower than its expected level. We can view this as that the failed coverage set "borrows" a certain amount of energy from the new primary coverage set. As long as the failed coverage set has its failed sensor replaced, it starts to "return" the energy it owes to the new primary set, until the energy level of the new primary set goes back to the expected level.

Fig. 3(d) shows an example. When a sensor in coverage set 3 fails at time instance $t = T/8$, it becomes a backup coverage set, and coverage set 0 becomes a primary coverage set. Meanwhile, a ready message is sent to the ES. Note that this message only asks for replacing the failed sensor node, not the entire coverage set. Since coverage set 0 starts to work $T/8$ earlier, we treat this as that coverage 0 debits $e/8$ energy to coverage set 3.

At time instance $t = T/4$, a deadline message is sent to the ES since both coverage sets 3 and 4 are not able to work normally. As a result, the failed sensor node in coverage set 3 and the entire coverage set 4 are replaced. Coverage set 3 will start to credit back $e/8$ energy to coverage set 0.

Since both coverage set 0 and 3 are primary coverage sets, the return of energy is done opportunistically. Specifically, if in a phase coverage 0 is supposed to be active, and coverage set 3 is not, coverage set 3 will be active in place of coverage set 0[1].

In the example shown in Fig. 3 (d), at time instance $t = 3T/16$, coverage set 3 has returned $e/16$ to coverage set 0, and at time instance $t = T/2$, coverage set has returned all $e/8$ energy to coverage set 0. At this time, the staircase structure goes back to the same shape as in Fig. 3 (a), and a ready message will be sent out at $t = T/2$.

E. Energy Consumption Balancing Scheme

In the case of irregular energy consumption rate, we still adopt the notion of disjoint coverage sets and staircase structure as in [8]. However, the formation of staircase structure is based on the mean of energy consumption rate α_{mean}. In other words, the staircase structure is formed as if all the sensors consume their energy at the rate α_{mean}.

The basic idea to deal with irregular energy consumption is to balance energy consumption among sensor nodes. In other words, if a sensor node consumes energy at a high rate, we can schedule the sensor node less frequently. On the other hand, if a sensor node consumes energy at a low rate, we can schedule the sensor node more frequently.

This can be done in two ways:

(i) If a sensor with energy consumption rate higher than α_{mean} is supposed to be active, neighboring sensors with relatively lower energy consumption rate (thus relatively higher remaining energy) can take its role.

(ii) If sensors in one geographical area consume energy faster than α_{mean} on average, we need to schedule these sensors less frequently, since if these sensors die, all coverage sets need to be replaced. In this case, we will need to find other geographical areas in which sensors consume energy slower than α_{mean} on average. Then we form a chain or tree for energy transfer. Note if one replacing sensor node is sufficient to cover the area of replaced sensor node, a chain is formed. Other wise, a tree is needed. Fig. 4(a) shows an example of energy transfer chain. In Fig. 4(a), node 0 has a faster energy consumption rate, and thus its remaining energy level is lower than expected. Node 4 has a slower energy consumption rate, and thus its remaining energy level is higher than expected. When node 0 is supposed to be active, while node 1 is not, node 1 will be active in place of node 0. When node 1 is supposed to be active, while node 2 is not, node 2 will be active in place of node 1, and so on. As a result, node 0 will save its energy for one phase, while node 4 will lose its energy for one phase. Fig. 4(b) shows an example of energy transfer tree where each node needs two of its neighbors to cover it when it is not active.

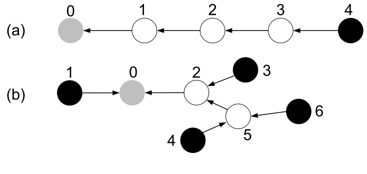


Fig. 4. Energy transfer methods. Gray nodes are recipients of energy, dark nodes are providers of energy, and other nodes serve as intermediate nodes to help transfer energy. (a) Energy transfer chain. (b) Energy transfer tree.

We call the two methods *energy transfer* methods, since sensor nodes with lower energy consumption rate offer its energy to help sensors with high energy consumption rate. Our scheme fulfills this object with low communication overhead.

---

[1] For phases with coverage number 4, the return of energy will not occur.

1) Terms and Notations: In our scheme, each node u maintains the information about its coverage set, and the corresponding stair. Each node also maintains the location information of its neighbors. Our scheme is composed of two algorithms: energy providing algorithm and energy requesting algorithm. Before presenting the algorithms, we give the key terms and notations.

- For each sensor u, we define *remedy coverage requirement*, $rc(u)$, which is the percentage of the area that needs to be covered by its replacing sensors if u is not active. The maximum remedy coverage requirement is 100%.
- For each sensor u, we define a coverage combination of u as a set of u's neighboring sensors that can cover at least $rc(u)$ of sensing area of u if u is not active. Since u knows the locations of all its neighbors, it can derive combinations of its neighbors that satisfy $rc(u)$. Each combination may have different number of sensors in it. Each node u maintains a set $C(u)$ which stores all the combinations that satisfy $rc(u)$.
- Each sensor node u has a status, $st(u)$, which could be *provider*, *savable*, and *non-savable*. The status *provider* means the sensor node has surplus remaining energy to offer to other nodes. The status *savable* means the node can receive energy from a providing sensor node through an energy transfer chain or a tree. The status *non-savable* means the node cannot receive energy from providing nodes.
- For each sensor u, we define a valid coverage combination of u, $VC(u)$, which is a subset of $C(u)$. For each sensor in each combination belonging to $VC(u)$, it has surplus energy itself (its status is *provider* or it can obtain energy from other nodes (its status is *savable*).
- Each node u's remaining energy is denoted as $e(u)$, while its expected remaining energy, i.e., the height of its stair, is denoted as $s(u)$.

2) Providing Energy: The energy providing algorithm is run every certain number I of phases, which I is system parameter. At the beginning of a phase in which the energy providing algorithm is scheduled to run, if a node u finds its remaining energy level is higher than its expected energy level, i.e., the height of its corresponding stair, by at least a given threshold t_p, it marks its status as *provider*, and broadcasts a *provide* message to its neighbors. Given the average consumption rate α_{mean}, t_p can be calculated as $t_p = a\alpha_{mean}$, where a is a system parameter. The provide message has the following format: $provide\langle u\rangle$.

If node u's remaining energy level is not higher than its expected energy level by least t_p, it set its status to *non-savable*.

When a node v receives multiple provide messages, it checks whether the senders of these messages can form a coverage combination that satisfy $rc(v)$. If such a combination C can be found, node v adds C to its valid coverage combination $VC(v)$. If node v's status is not *savable* or *provider*, it changes its status to *savable*, and broadcasts a new provide message: $provide\langle v\rangle$.

If node v's status is already *savable* or *provider*, node v does not send out provide message.

This process is carried on. When there is no provide message being transmitted in the area, all sensor nodes that can be provided with energy by other nodes, no matter whether these nodes are its neighbors or not, will be marked as *savable* or *provider*.

Alg. 2 describes the procedure of providing energy.

Algorithm 2 Providing energy: for sensor node u at the beginning of designated phases

```
Notations:
  $C(u)$: $u$'s coverage combination set
  $VC(u)$: $u$'s valid coverage combination set, which is a subset of $C(u)$. For all
  sensors in each combination in $P(u)$, their status is provider or savable
  $pro(u)$: $u$'s set for received provide messages
  $st(u)$: $u$'s current status, $s$:savable, $n$:non-savable, $p$:provider
  $e(u)$: sensor $u$'s remaining energy
  $s(u)$: sensor $u$'s expected remaining energy, i.e., the height of its stair
  $t_p = a\alpha_{mean}$: threshold for providing energy
Initialization:
  $VC(u) = \phi$
  if $e(u) - s(u) > t_p$ then
      mark its status as $provider$: $st(u) = p$
      broadcast $provide\langle u\rangle$
  else
      $st(u) = n$
Upon receipt of a provide message: $provide\langle v\rangle$
  Add the message to $pro(u)$: $pro(u) \leftarrow pro(u) \cup provide$
  Check whether a combination $C$ can be found to cover $u$ using $v$
  if $C$ can be found then
      Add $C$ to $VC(u)$: $VC(u) \leftarrow VC(u) \cup C$
      if $st(u) = n$ then
          $st(u) = s$
          // $u$ can an intermediate node to provide $v$'s energy to other nodes
          broadcast $provide\langle u\rangle$
```

3) Requesting Energy: The energy requesting algorithm is run at every phase. At the beginning of a phase, if a node u is supposed to be active according to the duty-cycle schedule, and finds its remaining energy is lower than its corresponding stair by at least a given threshold t_r, it requests help from other sensors. Given the average consumption rate α_{mean}, t_r can be calculated as $t_r = b\alpha_{mean}$, where b is a system parameter.

Node u checks its current status, and if the status is *non-savable*, then u cannot get help from other nodes. On the other hand, if u's status is *savable*, it checks its valid coverage combination set $VC(u)$, and randomly picks one combination. Node u then sends a request message to each sensor node in the selected combination. The format of a request message is: $request\langle u, s(u) - e(u)\rangle$, where $s(u) - e(u)$ is its energy deficit.

When a node v receives a request message, it first checks its status. If its status is *savable*, it reserves α_{mean} energy for sensor u. Then it randomly selects a combination in $VC(v)$, and forwards the request message to each of the sensor nodes in $VC(v)$.

If v's status is *provider*, v waits for a give time-out period τ, in which all requests should have been propagated to v. Then node v reserves energy for these requests. Assuming the number of requests is n, if v's energy surplus $e(v) - s(v)$ is sufficient to serve all the requests, i.e., $s(v) - e(v) > n\alpha_{mean}$, then v reserves $n\alpha_{mean}$ energy. Otherwise, v will not serve all the requests, instead it chooses $\lfloor \frac{s(v)-e(v)}{n\alpha_{mean}} \rfloor$ requests with highest energy deficit, and sends out a *reject* message to other non-selected requesting sensors.

After reserving appropriate amount of energy, if v finds its remaining energy has dropped below $s(v) + b\alpha_{mean}$, it checks its valid coverage combination set $VC(v)$. If $VC(v)$ is not empty, it changes its status to *savable*. Otherwise, it changes its status to *non-savable*, and broadcasts a *cancel* message: $cancel\langle v\rangle$.

When a node w receives this message, it checks its valid coverage combination set $VC(w)$, and removes the combinations that include node v. If $VC(w)$ becomes empty after the removal and its status is not *provider*, node w marks itself as *non-savable*, and broadcasts a *cancel* message: $cancel\langle w\rangle$.

On the other hand, if node $VC(w)$ is not empty after the removing the combinations that includes v, node w will broadcast a *provide* message. This step is necessary since there are nodes reachable from w whose status has been changed to *non-savable* due to the cancel message $cancel\langle v\rangle$.

After this process terminates, sensor v, who initiates the cancel message will not receive any request message any more.

Handling of Reject Messages: Reject messages will be forwarded back to the requesting sensor. During the forwarding, each intermediate node v will cancel the reservation that is made when receiving the corresponding request message. Further, if v has sent the corresponding request message to other nodes, it will send these nodes a *withdraw* message, and any node that receives this message will cancel the reservation that is made when receiving the corresponding request message.

Note that a providing node only sends reject messages when its remaining energy drop to below its expected energy level. Once this happens, it will not provide any energy to other nodes (its status is not *provider*) until the beginning of the next cycle for broadcasting provide messages.

Alg. 3 describes the procedure of requesting energy. For purpose of clarity, it does not include handling of reject messages.

Algorithm 3 Requesting energy: for sensor node u at the beginning of every phase

```
Notations:
    $VC(u)$: $u$'s valid coverage combination set
    $req(u)$: $u$'s set for received request messages
    $st(u)$: $u$'s current status, $s$:savable, $n$:non-savable, $p$:provider
    $e(u)$: sensor $u$'s remaining energy
    $s(u)$: sensor $u$'s expected remaining energy, i.e., the height its stair
    $t_r = b\alpha_{mean}$: threshold for requesting energy
    $t_p = b\alpha_{mean}$: threshold for providing energy
Initialization:
if $s(u) - e(u) > t_r$ then
    if $u_s = s$ then
        Randomly select a combination $C$ in $VC(u)$, and send a request
        $request\langle u, s(u) - e(u)\rangle$ to each sensor in $C$
Upon receipt of a request message from a neighbor $w$: $request\langle v, s(v) - e(v)\rangle$
Reserve $\alpha_{mean}$ energy for $w$
if $st(u) = p$ then
    Add the message to $req(u)$: $req(u) \leftarrow req(u) \cup \langle w, request\rangle$
    if timer $T$ is not started then
        Start timer $T$
else
    // $u$'s status must be savable, i.e., $st(u) = s$
    Randomly select a combination $C$ from $VC(u)$, and send $request\langle v, s(v) -$
    $e(v)\rangle$ to each sensor in $C$.
Upon timer $T$ fired
// $u$ must initiate a provide message, i.e., $st(u) = p$
Assume $u$ receives $n$ request messages
if $e(u) - s(u) > n\alpha_{mean}$ then
    Reserve energy $n\alpha_{mean}$
else
    Choose $\lfloor \frac{e_u - s_u}{\alpha_{mean}} \rfloor$ request messages with the highest energy deficit, and send
    a reject message to other requesting sensors
Reset the timer
Assume the reserved amount of energy is $e_r$
if $e(u) - e_r < s(u) + t_p$ then
    if $VC(u) = \phi$ then
        $st(u) = n$
        Broadcast cancel message $cancel\langle u\rangle$
    else
        $st(u) = s$
Upon receipt of a cancel message: $cancel\langle v\rangle$
Remove all combinations in $VC(u)$ which includes $v$
if $st(u) = s$ then
    if $VC(u) = \phi$ then
        Broadcast cancel message $cancel\langle u\rangle$
        $st(u) = n$
    else
        Broadcast provide message $provide\langle u\rangle$
```

F. Discussion

Since charging batteries takes non-negligible time, if the number of backup sensors x is too small, then the MR may not be able to replace all sensor nodes in time. In other words, some deadlines will be missed. We have derived a lower bound on the number of backup sensors in the system. The basic idea is that the energy replenishment rate, which is determined by x (x sensors can be recharged in parallel) should be greater than the energy consumption rate. The detailed derivation is similar to the one in [8], and is omitted due to space limit.

TABLE I
GENERAL EXPERIMENTAL SETTINGS

field size	$1000m * 1000m$
# of areas	80
sensing range	$20m$
transmission range	$40m$
N_{min}	1
N_{max}	4
N_{back}	1
recharging time	6 hours
sensor's lifetime time	240 hours (5 days)
# of sensors per coverage set	16 (by default)
sensor's full energy	1440 units
phase length	10 minutes
α_{mean}	0.1 unit/minute
provide broadcasting interval I	$\{25, 50, 75, 100, 125\}$ phases
cut-off time	4800 hours (200 days)

V. PERFORMANCE EVALUATION

We built a custom simulator using C++ to evaluate the performance of the proposed scheme.

A. Experimental Settings, Metrics and Methodology

Table I shows system parameters we used in the simulation. We consider a sensor network composed of 80 areas. The network field size is $1000m * 1000m$. Each area has $(N_{max} + N_{back})$ disjoint coverage sets, and each of which is able to cover the whole area. The number of sensors in each coverage set is 16 unless otherwise mentioned. The method for deploying sensor nodes is as follows: For each coverage set, we randomly deploy its first sensor, and then deploy other sensor in a way such that every two adjacent sensors are $25m$ apart.

In the experiments, we normalize the full energy level of a sensor to 1440 units and the mean value of energy consumption rate, α_{mean}, is 0.1 unit/minute if the sensor is active. Thus, each sensor's lifetime T is 240 hours, i.e., 5 days. The length of a phase is set to 10 minutes. Coverage numbers for each area vary between N_{min} and N_{max}. N_{min} is set to 1, and N_{max} is set to 4 in all experiments.

In reality, coverage number is determined by the application, as well as the real-time frequency and distribution of events. In our simulation, coverage number complies to a truncated Gaussian distribution, which is $Gau(\mu = N_{min}, \sigma = 2)$ truncated to the range $[N_{min}, N_{max}]$.

The performance metrics include:

- *Average replacement interval*: Average time between two consecutive replacement tours made by the MR.

- *Average utilization of the MR*: The MR may not carry x sensors in each replacement tour due to the replacement deadlines set by each area. Average utilization of the MR is the average ratio of the number of backup sensors actually carried by the MR to x.
- *Communication overhead* The total number of control messages, including provide, request, cancel, reject, withdraw messages per area per phase.

We consider the following sets of scenarios:

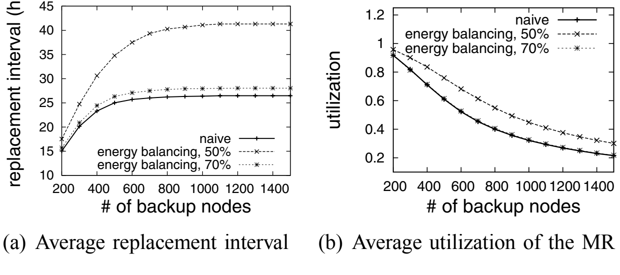
(i) The system is error-free, but sensors consume energy at different rates. We model the energy consumption rate in the following way: For each sensor u, its have a mean value $\alpha_{mean}(u)$ of its energy consumption rate, which is determined by manufacture reasons. $\alpha_{mean}(u)$ is a random variable which complies to $Gau(\alpha_{mean}, \sigma_1)$. Sensor u's energy consumption rate $\alpha(u)$ at a certain phase is another variable which complies to $Gau(\alpha_{mean}(u), \sigma_2)$. σ_1 and σ_2 are system parameters.
In this scenario, we study the performance of the energy balance scheme under different the system parameters, including the number of backup sensors x, σ_1 and interval for broadcasting provide messages I.

(ii) The system has failures, and also sensors consume their energy at different rates. We model failure events in the following way: at any phase, a failure event occurs with a certain probability f_p. If there is a failure event in a phase, the failure could happen at any sensor node with equal probability.
In this scenario, we fix system parameters σ_1, σ_2 and interval for broadcasting provide messages I, and compare the performance of the staircase repairing and the debit/credit scheme under different system parameters, including the number of backup sensors x and f_p.

For each experiment, our proposed scheme is executed for a long time period, starting at 0 and ending at a *cut-off* time. The cutoff time is set to 4800 hours, i.e, 200 days, for all experiments. Furthermore, we run each simulation for 50 times for each of the performance metrics.

B. Scenario I: Irregular Energy Consumption Rate

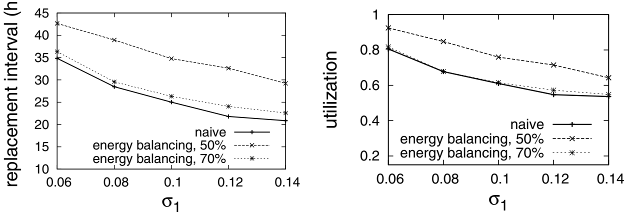
In this section, we study the performance of the proposed energy balancing scheme comparing with the naive scheme, and the impact of different parameters on the performance of the energy balancing scheme. In all experiments in this scenario, all sensor nodes have the same *remedy coverage requirement*, and the value can be either 50% or 70%. Further, the threshold for providing energy, and the threshold for requesting energy are $4\alpha_{mean}$, i.e., $t_p = t_r = 4\alpha_{mean}$.



(a) Average replacement interval (b) Average utilization of the MR

Fig. 5. Impact of x

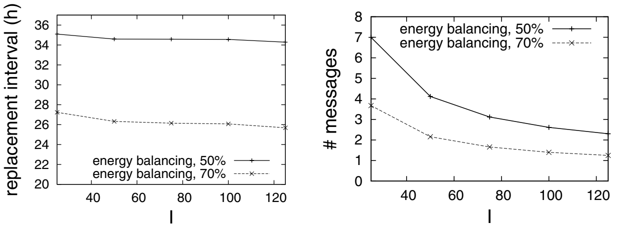
1) Impact of Number of Backup Sensors: In this experiment, we vary the number of backup sensors from 200 to 1500, and fix the provide broadcasting interval to 50 phases. We compare the energy balancing scheme when the remedy coverage requirement is 50% and 70% for all sensors, with the naive scheme in terms of the average replacement interval and the average MR utilization. As can be seen from Fig. 5, the energy balancing scheme has much better performance when the coverage requirement is 50%. The reason is that sensor nodes can find many providing sensors to help them. Further, all curves in 5(a) level off when x exceeds a certain value. This is because irregular energy consumption deforms the staircase structure, which incurs deadline messages being sent. In this case, the MR does not fully utilize the x backup sensors. In other words, more backup sensors will not help increase the replacement interval. The energy balancing scheme postpones the point when the replacement interval stops to increase.



(a) Average replacement interval (b) Average utilization of the MR

Fig. 6. Impact of σ_1

2) Impact of σ_1: In this study, we fix the number of backup sensors to 500, and provide broadcast interval to 50 phases. σ_1 is varied among $\{0.06, 0.08, 0.1, 0.12, 0.14\}$. σ_2 is fixed at 0.2. Fig. 6 shows the average replacement interval and average MR utilization for the energy balancing scheme when the remedy coverage requirement is 50% and 70%, and the naive scheme. As shown in the figure, both performance metrics decrease when σ_1 increases. Larger σ_1 causes more irrgularity on the time interval for a sensor to use up a coverage set, and thus causes more deadline messages being sent.



(a) Average replacement interval (b) Average utilization of the MR

Fig. 7. Impact of provide broadcast interval I

3) Impact of Provide Broadcast Interval I: In this study, we investigate the impact of provide broadcast interval I on the performance of the energy balancing scheme when the remedy coverage requirement is 50% and 70%, respectively. Fig. 7(b) shows the average number of control messages, including provide, request, cancel, reject, withdraw messages per area per phase. As can be seen, the communication overhead of the energy balancing scheme is fairly low. For instance, when the remedy coverage requirement is 50% and I is 50 phases, each area sees 4.12 messages on average.

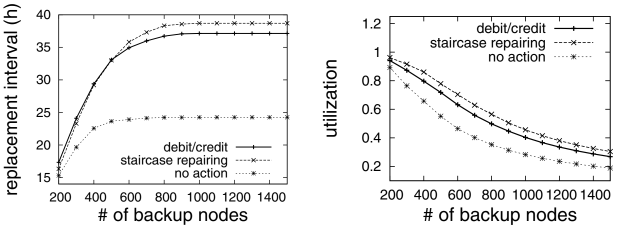
C. Scenario II: Failures and Irregular Energy Consumption Rate

We set $\sigma_1 = 0.05$, $\sigma_2 = 0.2$, the provide broadcast interval $I = 50$ phases, the remedy coverage requirement for all sensors to be 50%, and the threshold for providing energy and the threshold for requesting energy to be $4\alpha_{mean}$. We vary the number of backup sensors x and failure probability f_p, and compare the performance of the staircase repairing scheme and the debit/credit scheme.

1) Impact of Number of Backup Sensors x: In Fig. 8, we compare three schemes in which "no action" means we only run the energy balancing scheme without other schemes to deal with failure. The other two schemes work together with the energy balancing scheme. As can be seen from Fig. 8(a), the staircase repairing scheme and the debit/credit schemes perform much better than "no action". Furthermore, the debit/credit scheme performs better than the staircase repairing scheme when x is small, but performs worse when the x exceeds a certain value. This can be explained as follows.

When x is small, the demand for replacement exceeds x quickly, and thus the MR's replacing activities is mainly driven by the demand for replacement exceeding x. In the staircase repairing scheme, when a sensor node dies, the whole coverage set is replaced. This increases the demand for replacement, and hence the MR performs more replacements. However, when x is large enough, deadline messages will arrive before the demand for replacement exceeding x, and the MR's activities become driven by deadline messages. Since when the MR sets out for replacement in response to a deadline message, the demand is often less than x, a little more demands do not have a significant impact. This can also be seen from Fig. 8(b), which shows that the staircase repairing scheme has higher MR utilization.

There is also another reason that explains why the staircase repairing scheme performs better than the debit/credit scheme when x is large. In the staircase repairing scheme, when a sensor in a coverage set fails, the sensors in coverage sets with a higher stair will "reduce" their energy to one stair down. However, sensors in such a coverage set will not reduce the same amount of energy. Sensors with higher remaining energy could reduce more amount of energy, while sensors with lower remaining energy could reduce less. In other words, a sensor failure gives sensors in some coverage sets an opportunity to narrow down their difference in remaining energy level, thus helps keep the staircase structure.

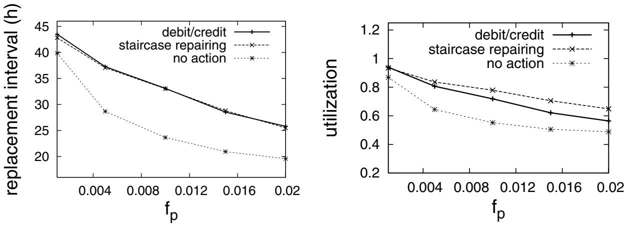


(a) Average replacement interval (b) Average utilization of the MR

Fig. 8. Impact of x

2) Impact of Failure Probability f_p: In this study, we fix $x = 500$, and vary failure probability f_p among $\{0.001, 0.005.0.01, 0.015, 0.02\}$.

Fig.9 shows that as f_p increases, both average replacement interval and average utilization of the MR decreases. This is because more failures cause more deadline messages being sent, which forces the MR to perform replacement without making full use of available backup sensors.



(a) Average replacement interval (b) Average utilization of the MR

Fig. 9. Impact of failure probability f_p

VI. CONCLUSION

In this paper, we proposed three schemes to address sensor failure and irregular energy consumption rate in realizing the NRR strategy under area coverage model. These schemes achieve the design goal of reliably and efficiently supporting replaceable long-lived sensor networks. The simulation results show the scheme are both effective and efficient.

VII. ACKNOWLEDGMENTS

This work was partially supported by NSF Grands CNS-0831874, CNS0716744, CNS-0831906, and CNS-0834585.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, 2002.
[2] K. Zeng, K. Ren, W. Lou, and P. J. Moran, "Energy-aware geographic routing in lossy wireless sensor networks with environmental energy supply," in *Proc. of QShine '06*, Waterloo, Ontario, Canada, 2006.
[3] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proc. of IPSN '05*, Los Angeles, CA, 2005, pp. 457–462.
[4] A. Kansal, J. Hsu, M. B. Srivastava, and V. Raghunathan, "Harvesting aware power management for sensor networks," in *Proc. of DAC '06*, San Francisco, CA, 2006, pp. 651–656.
[5] A. Kansal and M. B. Srivastava, "An environmental energy harvesting framework for sensor networks," in *Proc. of ISLPED '03*, Seoul, Korea, 2003, pp. 481–486.
[6] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," in *Proc. of ACM SIGMETRICS '04*, New York, NY, 2004, pp. 223–234.
[7] B. Tong, G. Wang, W. Zhang, and C. Wang, "Node reclamation and replacement for long-lived sensor networks," in *Proc. of IEEE SECON '09*, Rome, Italy, 2009.
[8] B. Tong, Z. Li, G. Wang, and W. Zhang, "On-demand node reclamation and replacement for guaranteed area coverage in long-lived sensor networks," in *Proc. of QShine '09*, Las Palmas de Gran Canaria, Spain, 2009.
[9] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. D. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *TOSN*, vol. 1, no. 1, 2005.
[10] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Wireless Ad Hoc and Sensor Networks: An International Journal*, vol. 1, no. 1-2, pp. 89–123, 2005.
[11] X. Bai, S. Kuma, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proc. of MobiHoc '06*, Florence, Italy, 2006, pp. 131–142.