A Synthetic Function for Energy-Delay Mapping in Energy Efficient Routing

Abdelmalik Bachir and Dominique Barthel
France Telecom R&D
Meylan, France
{Abdelmalik.Bachir, Dominique.Barthel}@francetelecom.com

Martin Heusse and Andrzej Duda
LSR-IMAG Laboratory
Grenoble, France
{Martin.Heusse, Andrzej.Duda}@imag.fr

***Abstract*— In traditional approaches to energy efficient routing, a node needs to receive routing messages from all of its neighbors to be able to select the best route. In a previous work, we have proposed a technique that enables the best route selection based on exactly one message reception [1]. Our protocol delays forwarding of routing messages (RREQ) for an interval inversely proportional to the residual energy. Energy-delay mapping techniques make it possible to enhance an existing min-delay routing protocol into an energy-aware routing that maximizes the lifetime of sensor networks. We have proposed some heuristic functions to perform the energy-delay mapping. This paper analyzes their limitations and derives a suitable synthetic function that guarantees that a node selects the best route with very high probability. We also identify comparative elements that help us to perform a thorough a posteriori comparison of the mapping functions in terms of the route selection precision. Simulation results show that our synthetic functions select routes with very high precision while keeping the propagation delay of routing messages reasonable.**

I. Introduction

Sensor networks are composed of wireless nodes that sense various environmental phenomena and maintain communication interconnection via multihop routing. These easily deployable, self-organized, and relatively low-cost networks are expected to be massively deployed in many applications such as habitat monitoring, disaster relief and surveillance [2]–[4]. The success of the applications relies on the network lifetime that depends on the life span of nodes. Hence, energy saving is the crucial factor in designing long-lived sensor networks, mainly because nodes are powered by batteries that may be costly, difficult, or even impossible to replace or recharge.

Designing a universal scheme for optimizing energy savings is challenging due to the variety of sensor network applications. However, for most of applications, measurements presented in the literature [5], [6] and obtained from our expriments (Table I[1]) show that radio communication is a major source of energy consumption. Therefore, many protocols at different layers have been proposed to address this issue [9]. In the rest of this paper, we focus on energy-efficient routing protocols [10].

TABLE I
Current Consumption Measurements for FreeScale MC 13192 SARD

Radio Idle (not ready to receive)	0.5 mA
Radio Tx (Transmit)	39 mA (at +4 dbm)
Radio Rx (Receive)	39 mA
MCU (Active)	10 mA
MCU (Partially Active)	8 mA
LED	4 mA
Accelerometer sensors	3 mA

At the routing layer, energy-efficient protocols use one strategy or a combination of them to maximize network lifetime: a) min energy metric and b) max-min residual energy metric. In min energy routing, nodes select the route that consumes the least amount of energy. Usually, nodes adjust their transmission power and construct a minimum energy topology to reduce the overall energy consumption of the network [11], [12]. The resulting topology guarantees that each node communicates with other nodes using the route that consumes the least amount of energy possible overall. In max-min residual energy routing, nodes estimate their residual energy and cooperate to prevent the most vulnerable ones from being overused avoiding in this way premature energy exhaustion [13]. Such protocols choose routes bypassing vulnerable nodes, which ensures load balancing and avoids early network fragmentation.

Many research results (see also Section V) conclude that an energy efficient routing protocol that maximizes the life span of a sensor network should combine both min energy and max-min residual energy metrics, because these two approaches are complementary. Indeed, at the beginning of the network life time, the network is dense and nodes have high residual energy: the use of a pure max-min metric may be counter effective—by trying to protect nodes with low residual energy, the max-min metric always selects routes for which the most vulnerable node has the highest residual energy; such a route may actually dissipate more energy than others. So, the min energy metric, which selects the route with the least energy consumption, is a better choice when nodes have enough energy, i.e. their residual energies are larger than a predefined threshold. The max-min residual energy metric should be used to protect nodes with low residual energy, i.e. less than a predefined threshold.

Although such hybrid protocols contribute to better network

[1]We carried out these measurements on the MC 13192 SARD sensor node. The measurements closely match the values announced in datasheets [7], [8].

lifetimes, they still have some drawbacks. In another work [1], we have identified the problem of superfluous routing messages that a node may receive while making the best routing decision. Indeed, in traditional routing protocols with the metrics such as min energy or max-min residual energy, a node needs to receive routing messages from all of its neighbors to be able to select the best route, because the messages contain values required for route selection. We argue that the reception and comparison of all the messages are not needed, since the node eventually selects only one route. To address this issue, we have proposed an approach that enables the best route selection based on exactly one message reception [1]. Our protocol delays forwarding of routing messages (RREQ) for an interval inversely proportional to the residual energy. In this way, the routing message on the best route arrives the first so that the node may ignore the superfluous routing messages that arrive afterwards. Nevertheless, the proposed energy-delay mapping does not guarantee that the selected route is always the best, because the intentional forwarding delay was based on heuristic functions [1].

In this paper, we address these limitations and propose a synthetic function instead of heuristic ones to make sure that a node selects the best route with very high probability. We also identify comparison elements that help us afterwards to perform a thorough a posteriori comparison of the mapping functions in terms of route selection precision.

II. Background

A. Diffusion Routing

Energy-delay mapping techniques enhance any min-delay routing including gradient routing used in *Directed Diffusion* [14]. Gradient routing is destination-initiated in the sense that data collectors (also called sinks) interrogate data publishers (also called sources) asking for specific data. This phase, similar to a route request in on-demand routing protocols, is called interest propagation. It establishes localized data-forwarding pointers (called gradients) from sources to sinks. The sources then stream the requested data back to the sinks according to the directions indicated by the gradients. Although there are different implementations of gradient routing, one phase pull directed diffusion is the best fit when few sinks collect the data published by many sources [15]. Since such situations are fairly frequent in sensor network applications, we consider without loss of generality the one phase pull directed diffusion[2] and enhance it with our solution based on delaying routing messages (RREQ) for an interval inversely proportional to the residual energy.

Our motivations for using diffusion are the following:

- Computational complexity is reduced to a minimum. Each node only needs to broadcast one interest message during the interest propagation phase and it only needs to receive one interest message to setup its routing table (it can ignore the subsequent interest messages related to that same interest). The latter property is particularly interesting, because we have designed a MAC protocol able to identify redundant frames before their complete reception [16]. In this way, a node may turn the radio off to avoid receiving superfluous interest messages, which saves energy.
- There is no overhead due to the exchange of extra information like hello or route metrics messages, which saves more energy and reduces the complexity of the routing protocol in terms of computation and memory occupation. Remind that sensor nodes have usually very limited capacities (for example, nodes used in our experiments have a 8-bit micro controller running at 16 MHz maximum speed and 4KB RAM).
- Routing tables only require one entry per active interest consisting of a pointer toward the next node downstream.
- It enables in-network processing to aggregate data based on attributes used in diffusion, which furthermore saves energy by reducing the size and the number of transmitted/received messages.

[2]which we will simply call diffusion.

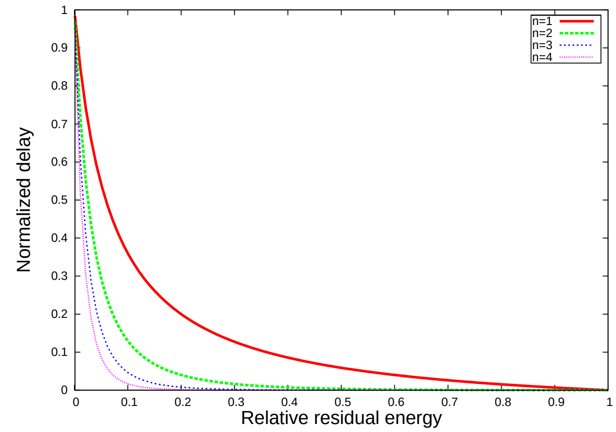

Fig. 1. Heuristic Mapping Functions

B. Heuristic Mapping Functions

Nodes using energy-delay mapping compute a forwarding delay based on their residual energy and defer forwarding of interest messages for this period of time. We have defined energy-delay mapping functions having the property that high residual-energy nodes forward messages without delay, in which case diffusion is equivalent to min energy routing. Nodes with lower energy delay forwarding for a time interval, which results in max-min residual energy routing.

To find a mapping function f with suitable properties, we have explored a family of decreasing convex functions of the form $(1/x)^{\eta}$, where η is a positive parameter. We have shifted and shrunk them so that they map $[0, 1] \to [0, 1]$: the residual energy in $[0, 1]$ into the normalized delay in $[0, 1]$. Fig. 1 presents the resulting set of functions labeled f_{η} with η taking integer values from 1 to 4.

The form of this set of functions can be controlled through two parameters. The first parameter, called sensitivity threshold, separates the min energy metric, when the flat part of the function is used, from the max-min residual energy metric, when the curvy part of the function is used. For example, the sensitivity threshold of function f_3 is around 0.5, which means that a node using this mapping function does not apply intentional delay when its residual energy is larger than 0.5. Therefore, if we have routes with nodes having residual energies larger than 0.5, the selected route will be the one with the min-delay, which very likely corresponds to the shortest path consuming the minimum energy[3].

The second parameter is the convexity of the function that determines the ability of the mapping function to perform max-min routing. The purpose of the convexity is to have the intentional delay applied by the node with the minimum residual energy on a route being dominant. In this way, the route with the max-min residual energy will be selected, because the routing message of this route will have the smallest delay. The convexity parameter determines the precision of the approximation in Eq. 30 (see Appendix): the more convex the mapping function, the better the approximation. For example, function f_4 has stronger convexity than other functions in the considered set so that it performs better max-min routing.

These heuristic functions have some drawbacks: a) they are likely to be sub-optimal and b) there is a correlation between the convexity of the function and the sensitivity threshold. That is, if we need more precise max-min routing, we will have a smaller sensitivity threshold (e.g. 0.2 for f_4).

To overcome these drawbacks, we propose in the next section a synthetic mapping function that allows exact min to max-min delay mapping according to an uncorrelated predefined threshold. This mapping function is to be used in the situation in which residual energies of nodes are expressed as step functions and not continuous ones.

III. Synthetic Mapping Function

A. System Model

We use the following definitions and assumptions:

- Each node is able to measure its relative residual energy ζ, $(0 \leq \zeta \leq 1)$.
- We call γ the battery protection threshold, $(0 < \gamma < 1)$.
- A node is *vulnerable*, if its residual energy is less than battery protection threshold γ.
- A node is *critical* for a route (to which it belongs), if it has the least amount of residual energy among all the nodes forming that route.
- The *residual energy of a route* is equal to the residual energy of the critical node for that route.
- A *route is vulnerable*, if its residual energy is less than γ.

We assume that there is an *ideal* routing protocol that maximizes the lifetime of a sensor network. According to literature (see Section V), the ideal protocol combines both min energy and max-min residual energy metrics. We assume [4] that the ideal protocol relies on the battery protection threshold concept [17], that is, the ideal protocol uses the min energy metric to select routes as long as there is no any vulnerable route to save energy per packet transmission. Otherwise, when all the routes become vulnerable, the ideal protocol uses the max-min residual energy metric to protect the most vulnerable nodes.

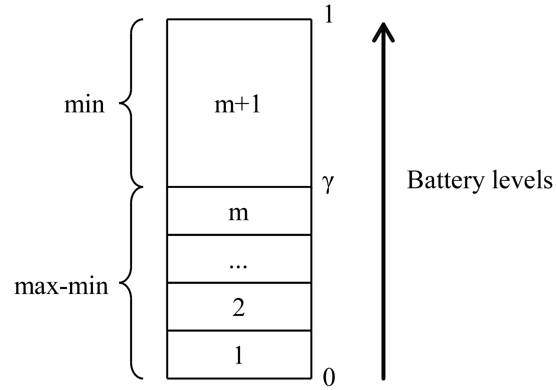

Fig. 2. Energy Levels

In actual implementations of routing protocols, the energy-delay mapping function would likely be discrete and tabulated. Indeed, a node may read its battery voltage or internal resistance and perform table lookup to get the corresponding level of its residual energy. Therefore, we can assume that residual energies of nodes are discrete. We aggregate all the energy levels greater than γ into one energy level as shown in Fig. 2. We call m the number of energy levels that are less than γ. We assign to each node an energy level l depending on its residual energy. We can say that a node with residual energy ζ has energy level l if

$$(l-1)\frac{\gamma}{m} < \zeta \leq l\frac{\gamma}{m}. \tag{1}$$

If ζ is larger than γ, the node has energy level of $m+1$. Explicitly,

$$l = \begin{cases} \left\lceil \frac{m\zeta}{\gamma} \right\rceil & \text{if } \zeta \leq \gamma \\ m+1 & \text{otherwise.} \end{cases} \tag{2}$$

Let g be a synthetic function that maps residual energy into intentional forwarding delay d: $d = g(\zeta)$. As we use discrete energy levels instead of continuous residual energy, function g depends on m. Therefore, intentional forwarding delay $d^{(l)}$ that corresponds to energy level l is the following:

$$d^{(l)} = g_m(l). \tag{3}$$

[3]This is true when nodes use the same transmission power and wireless links have the same error rate.

[4]Note that the question of the ideal routing protocol is still open, since the definition of network lifetime itself is still open. In this paper, we consider the time to partition as the definition of the network lifetime.

TABLE II

NOTATION

p_γ	probability that a node is not vulnerable
$\vert\mathcal{R}\vert$	number of disjoint routes between the source and the sink
$\vert R_k\vert$	length of route R_k
n	number of intermediate nodes on the longest route between the source and the destination
$P_{min}(k)$	probability that route R_k is not vulnerable
$P_{maxmin}(k)$	probability that route R_k is vulnerable
P_{maxmin}	probability that an ideal protocol selects a vulnerable route

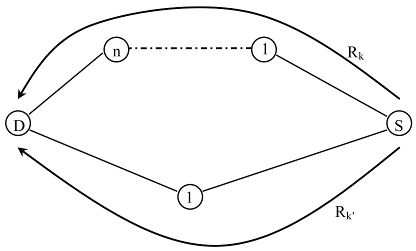

Fig. 3. The Worst Case

B. Deriving Synthetic Mapping Function

Assume we have source node S and destination node D. We call $\mathcal{R}$ the set of all possible routes between the source node and the destination node.

Let us consider route R_k, $(R_k \in \mathcal{R})$. We call $|R_k|$ the number of intermediate nodes on route R_k (source node and destination node are not included). We use the following notation to represent R_k, $R_k = N_{k1} - \cdots - N_{ki} - \cdots - N_{k|R_k|}$, where N_{ki} represents an intermediate node on route R_k.

We propose to derive synthetic function g that meets our goals even in the worst case. It is obvious that g needs to be decreasing to have $g(l) < g(l')$ for all $l > l'$. Besides, g also needs to be convex to mitigate the effect of increasing delay cumulated along longer routes. Fig. 3 shows the worst case example that can be expressed with two routes R_k and $R_{k'}$. Route R_k has the maximum route of length $|R_k| = n$ and residual energy level l, whereas route $R_{k'}$ has the minimum route of length $|R_{k'}| = 1$ and residual energy level $l-1$. In this case, $D^{(R_k)}$, the interest propagation delay on route R_k should be less than $D^{(R_{k'})}$. It is sufficient to have

$$D^{(R_{k'})} = D^{(R_k)} + 1. \tag{4}$$

therefore,

$$\sum_{i=1}^{|R_{k'}|} D_{k'i} = \sum_{i=1}^{|R_k|} D_{ki} + 1, \tag{5}$$

where D_{ki} is the delay incurred by node N_{ki}. Actually, delay D_{ki} is composed of two delays: intentional delay d_{ki} caused by the synthetic mapping function and inherent system delay δ_{ki} that includes computation and transmission delays as summarized in Table IV. For example, in contention-based MACs such as 802.11-inspired MACs, the system delay also includes the maximum jitter, used to alleviate the rate of collisions caused by simultaneous access to the channel:

$$D_{ki} = d_{ki} + \delta_{ki}. \tag{6}$$

In the worst case, nodes on route R_k experience maximum system delays, i.e. $\delta_{ki} = \delta_{max}$ and nodes on route $R_{k'}$ experience minimum system delay $\delta_{k'i} = 0$. Also, all nodes on route R_k have their energy level equal to l, i.e. $d_{ki} = g_m(l)$ for $i = 1, \cdots, |R_k|$ and the node on route $R_{k'}$ has its energy level equal to $l-1$, i.e. $d_{k'i} = g_m(l-1)$ for $i = 1, \cdots, |R_{k'}|$. Therefore, Eq. 5 can be rewritten as:

$$g_m(l-1) = n\left[g_m(l) + \delta_{max}\right] + 1. \tag{7}$$

We set $g_m(m+1)$ to 0. This means that non vulnerable nodes do not apply any intentional delay, so the min-delay routing turns into min-hop routing: we get non vulnerable routes selected according to min-hop routing, which is equivalent to min energy routing as we assume identical links (nodes use the same transmission power and have the same transmission error probability). Therefore, we have

$$g_m(l) = \begin{cases} (n\delta_{max} + 1)\frac{n^{m-l+1}-1}{n-1} & \text{if } \ l \leq m \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

IV. PERFORMANCE EVALUATION

A. Methodology

We propose to compare the performance of our protocol based on min-delay routing with energy-delay mapping with the ideal protocol.

Since an energy efficient protocol combines two metrics, min and max-min, we need to use two performance indices for evaluation. For the min metric, we introduce the global gain ratio defined as the global energy consumption ratio between our protocol and the ideal one. In our simulations, this is equivalent to measuring the ratio between the number of hops, because we assume that we do not use transmission power control and links have the same error probability. More specifically, gain $\mathcal{G}$ is defined as:

$$\mathcal{G} = \frac{\sum |R_{our}|}{\sum |R_{ideal}|} \tag{9}$$

where $\sum$ means the sum over all simulation runs.

For the max-min metric, we introduce another performance index: the criticality of a route C. It depends on the residual energy ζ of that route and on the battery protection threshold γ.

$$C = \begin{cases} \zeta/\gamma & \text{if } \zeta < \gamma \\ 1 & \text{if } \zeta \geq \gamma \end{cases} \tag{10}$$

TABLE III

COMPARISON ELEMENTS

	Min (Ideal protocol)	Max-Min (Ideal protocol)
Min (Our protocol)	Case 1	Case 3
Max-Min (Our protocol)	Case 2	Case 4

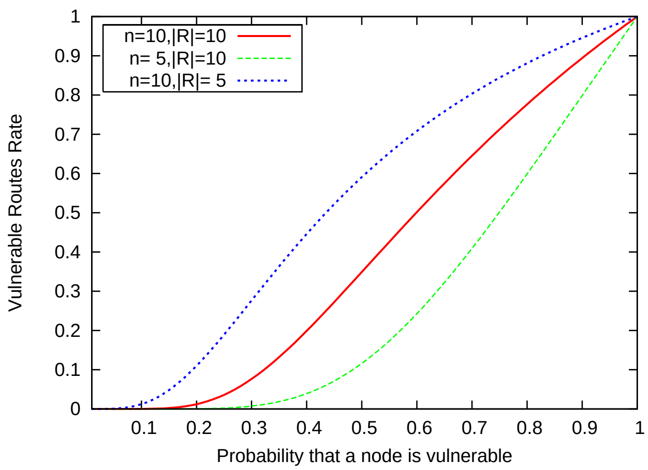

Fig. 4. The Rate of vulnerable routes with respect to the probability that a node is vulnerable. $|\mathcal{R}|$ is the number of disjoint routes between the source and the destination.

We compute the criticality ratio between the criticality of a selected route and the criticality of the ideal route. More specifically, criticality ratio $\mathcal{C}$ is defined as:

$$\mathcal{C} = \frac{\sum C_{our}}{\sum C_{ideal}} \tag{11}$$

Table III summarizes the elements of comparison between our protocol and the ideal one. We distinguish four cases depending on the difference and the resemblance of the metrics used by our protocol and the ideal one.

Case 1: In this case, since both the selected and the ideal routes are not vulnerable (the use of min metric), we only measure the global gain ratio.

Case 2: In this case, the selected route is vulnerable, but the ideal route is not. This case sometimes happens when our protocol fails to find a non vulnerable route, usually when non vulnerable routes are far longer than vulnerable ones. In this case, we measure the average criticality ratio.

Case 3: This case is impossible.

Case 4: In this case, both the selected and the ideal routes are vulnerable (the use of max-min metric). In this case, we measure the average criticality ratio.

B. The Proportion of Vulnerable Routes

We propose to analyze the probability with which a node uses min or max-min metrics to select routes. This probability depends on many parameters shown in Table II.

The ideal protocol picks out a route according to the max-min metric if all the routes are vulnerable. Then,

$$\begin{aligned} P_{maxmin} &= \prod_{k=1}^{|\mathcal{R}|} P_{maxmin}(R_k) \\ &= \prod_{k=1}^{|\mathcal{R}|} (1 - P_{min}(R_k)) \end{aligned} \tag{12}$$

A route is not vulnerable iff all the intermediate nodes on that route are not vulnerable. Therefore,

$$P_{min}(R_k) = \prod_{i=1}^{|R_k|} p_\gamma, \tag{13}$$

where $|R_k|$ is the length of route R_k. So,

$$P_{maxmin} = \prod_{k=1}^{|\mathcal{R}|} \left(1 - \prod_{i=1}^{|R_k|} p_\gamma\right) \tag{14}$$

The mean $E[P_{maxmin}]$ is the following:

$$E[P_{maxmin}] = (E[1 - p_\gamma^L])^{|\mathcal{R}|}, \tag{15}$$

where L is a discrete random variable in $[1, n]$

$$\begin{aligned} E[1 - p_\gamma^L] &= \sum_{i=1}^{n} (1 - p_\gamma^i) \cdot P\{L = i\} \\ &= \frac{1}{n}\left(n - \sum_{i=1}^{n} p_\gamma^i\right). \end{aligned} \tag{16}$$

Finally,

$$E[P_{maxmin}] = \left[1 - \frac{1}{n}\left(\frac{p_\gamma}{1 - p_\gamma}\right) + \frac{p_\gamma^n}{n}\left(\frac{p_\gamma}{1 - p_\gamma}\right)\right]^{|\mathcal{R}|} \tag{17}$$

From Eq. 17 and Fig. 4, we conclude that the probability of selecting a route according to max-min (i.e. all the routes are vulnerable) decreases when the number of routes $|\mathcal{R}|$ increases. This means that in dense networks in which there are many alternative routes, finding a route, which is not vulnerable, becomes very likely. We also notice that probability P_{maxmin} increases when the number of intermediate nodes n increases, which is quite expected. Besides, when probability p_γ that a node is not vulnerable increases, probability P_{maxmin} that all the routes are vulnerable decreases, because the number of vulnerable nodes decreases.

C. Worst Case Interest Propagation Delay

Assume that there are n intermediate nodes $N_1, ..., N_n$ between the source and the destination. Each node N_i has residual energy level l_i. On route $R = N_1 - ... - N_n$, node N_i receives the interest at time t_i (we assume the destination sends the interest at time 0):

$$\begin{cases} t_1 &= \delta_1 \\ t_2 &= (g(l_1) + \delta_2) + \delta_1 \\ t_3 &= (g(l_2) + \delta_3) + (g(l_1) + \delta_2) + \delta_1 \\ \vdots & \\ t_{n+1} &= \sum_{i=1}^{n} (g(l_i) + \delta_{i+1}) + \delta_1 \end{cases} \tag{18}$$

TABLE IV

SUMMARY OF DELAYS USED IN SIMULATION

Transmission time	41.6ms (52 bytes at 10kbps)
Computation time	15 to 45ms, uniform
MAC random back-off	0 to 10 * transmission time, uniform

where t_{n+1} is the time when the source receives the interest.

In the worst case, all intermediate nodes N_i, $i = 1, ..., n$ have residual energy levels of 1 (i.e. $l_i = 1$ for all $i = 1, ..., n$) and all system delays $\delta_i = \delta_{max}$ for all $i = 1, ..., n$. Hence, the maximum interest propagation delay in the worst case corresponds to the maximum value of t_{n+1}, which is:

$$\begin{aligned} D_{max} &= n\left(n^{m-1}-1\right)\left(\delta_{max}+\frac{1}{n-1}\right) \\ &= O(n^m \delta_{max}). \end{aligned} \tag{19}$$

D. Simulations

We have run a series of simulations to evaluate the precision of route selection by our protocol based on the proposed energy-delay mapping function compared with the ideal protocol based on the battery-protection threshold. In each simulation run, we have distinguished four cases discussed in Section IV-A. For each case, we have measured the corresponding gain and the criticality ratios. We have also measured the average end-to-end interest propagation delay to evaluate the trade-off between the protocol precision and the delay.

We have carried out 10^4 simulation runs. Each run generates 10 disjoint routes from the source to the sink. To cover a large number of different topologies, we assign a uniformly distributed random number of intermediate nodes to each route. The length of any route does not exceed n intermediate hops. To model the residual energy of nodes, we use the Gaussian distribution $G(\mu, \sigma)$ with mean μ and standard deviation σ. Each node has residual energy distributed according to $G(\mu, \sigma)$; we discard the values of $G(\mu, \sigma)$ outside the interval $[0, 1]$. We have set the battery protection threshold γ to 0.2, because it has been shown that this value results in better performance [18].

As shown in Fig. 4, the rate of vulnerable routes in the network depends on p_γ, the probability that a node is vulnerable, which in turn depends on μ and σ. We have varied μ and σ to compare the precision of our mapping functions, heuristic and synthetic, in different situations. To represent three different situations, we take $\mu = \sigma = 0.5$, $\mu = \sigma = 0.2$, and $\mu = \sigma = 0.1$. We have chosen function f_3, which corresponds to $\eta = 3$ in Fig. 1, as a representative for heuristic functions, because its sensitivity threshold is near the battery protection threshold γ. As a representative for synthetic functions, we take function g_m derived in Eq. 8 with different values for m, $m = 1, \ldots, 5$. For each mapping function, we analyze the precision, evaluated by the gain and criticality ratio parameters, and the average delay to obtain the best precision-delay trade-off. Note that synthetic functions achieve similar precision as the ideal protocol when the best route fits in the min-energy part of routing. Also, Case 2 never happens, because the intentional delay is chosen to avoid this case.

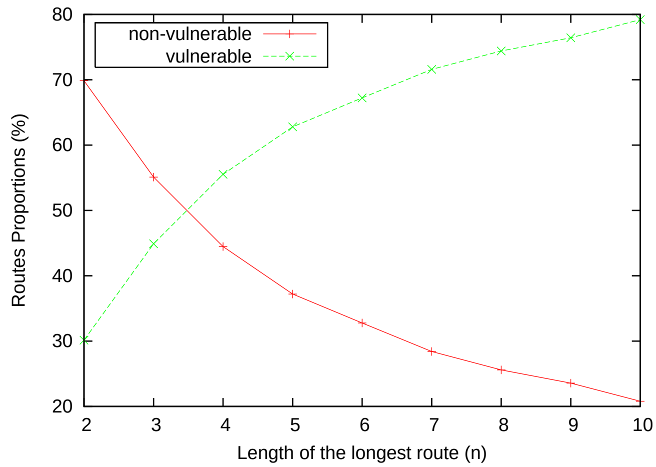

Fig. 5. Routes Proportions

When $\mu = \sigma = 0.5$, there are very few vulnerable routes, around 1%. In this case, we restrict the analysis only to the precision of the min-energy part of heuristic function. Contrary to synthetic functions that select routes with the same length as the ideal protocol, heuristic function f_3 selects routes that are on the average 4% larger than the ideal routes. This is a consequence of the sensitivity threshold of function f_3 being slightly higher that the battery protection threshold. Moreover, function f_3 wrongly uses the max-min metric instead of using the min one for routes with residual energy greater than the battery threshold and less than sensitivity threshold.

When $\mu = \sigma = 0.2$, the proportion of vulnerable routes increases with route lengths from 0.15%, if the maximum route length is 2, to 20%, if the maximum route length is 10. Case 2, in which function f_3 fails to find a non vulnerable route, happens in 4% of the runs. This is due to the convexity of function f_3, that makes the delay on short vulnerable routes not being dominant. Note that we do not have these problems with synthetic functions.

For the precision of vulnerable route selection, function f_3 selects vulnerable routes with a criticality ratio of 95%, which means that residual energy of selected routes is 5% less than the one of the ideal route, whereas synthetic function g_5 selects routes with a criticality ratio of 98%.

When $\mu = \sigma = 0.1$, the proportion of vulnerable routes increases up to 80% for networks with the length of routes up to 10 nodes. Fig. 5 plots the proportion of runs with Case 4 and Case 1, which corresponds to the proportion of vulnerable routes and non-vulnerable routes respectively. We do not plot runs with Case 2, because they are fairly rare, under 2%. In this aging network, function f_3 perfectly selects non-vulnerable routes and selects vulnerable routes with an average criticality ratio of 97%. In this case, the average interest propagation delay is around 2.23 seconds for routes of length up to 10 nodes.

Fig. 6 shows the corresponding criticality ratios and average

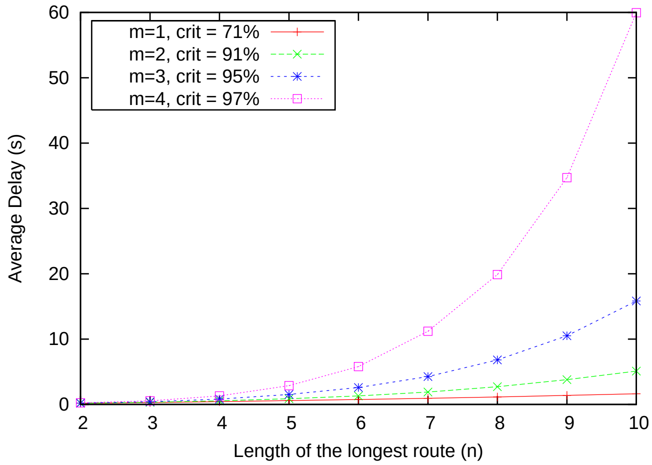

Fig. 6. Average Delay

delays for synthetic function g_m, $m = 1, \ldots, 4$. We show that to obtain to obtain a criticality ratio of 97%, we need four levels of residual energy below the battery protection threshold, *i.e.* $m = 4$. We also show that this criticality ratio requires the average delay of 59.95 seconds. We can conclude that synthetic function g_4 is a good candidate for networks with routes of length up to 10 nodes, as it selects routes with high precision whatever the residual energy distribution of nodes. Indeed, synthetic function g_4 perfectly achieves the min part of the ideal routing and efficiently selects vulnerable routes: only 3% difference between the residual energies of the ideal route and the selected route, which corresponds to the criticality ratio of 97%. We argue that the average delay is not very long because this delay is only used when refreshing routes or finding new ones and it does not affect data delivery latency. We believe that a 1 minute delay to refresh routes is tolerable in network with low dynamicity and steady tasks.

V. Related Work

Toh et al. [17] have proposed CMMBCR (Conditional Max-Min Battery Capacity Routing) for the network lifetime maximization problem. CMMBCR is a combination between MTPR, the min energy metric, and MMBCR, the max-min residual energy metric. In their proposal, they define battery protection margin γ, $(0 \leq \gamma \leq 100)$ and differentiate two kinds of routes: A and Q. Q is the set of all possible routes between a source and a destination nodes. A, a subset of Q, is the set of the routes having residual energy greater than γ, i.e. all the nodes on each route in A have residual energies larger than γ. The protocol is the following: when there is no route in A with residual energy below γ (i.e. all the possible routes contain vulnerable nodes), the protocol selects a route in Q according to the max-min residual energy routing (MMBCR) to protect the most vulnerable nodes. Otherwise, when there is at least one route in A, the algorithm selects a route in A according to the min energy routing (MTPR) to save energy. Note that γ is the parameter that controls the trade-off between MMBCR and MTPR.

Misra et al. [19] take the link transmission cost between nodes into account and propose MRPC (Maximum Residual Packet Capacity) to improve the previous protocol. They model the link transmission cost according to the link error rate and the physical distance between nodes. They introduce a node-link metric C_{ij}, for each link $i \rightarrow j$, that depends on the residual energy B_i of node i, and on the transmission power ζ_{ij} needed to send a packet from i to j. Explicitly, $C_{ij} = B_i/Eij$. The node-link metric determines the lifetime of the link $i \rightarrow j$. The lifetime Life$_R$ of route R depends on the lifetime of the most vulnerable link on this route, Life$_R = min\{C_{ij}\}$, where $i \rightarrow j$ is a link on route R. The protocol is then straightforward: given a set of routes between a source and a destination node, choose the route with the largest lifetime. Note that basic MRPC is a pure max-min residual energy routing, which could have undesirable behavior by always tending to protect the most vulnerable link. To cope with this issue, Misra et al., propose CMRPC (Conditional MRPC) that uses life protection threshold γ by analogy to the battery protection threshold [17]. That is, CMRPC first tries to select the route with the minimum energy consumption among the routes whose lifetimes are larger than γ. Otherwise, if there is no route satisfying this condition CMRPC switches to MRPC. Simulation results show that CMRPC improves the performance of MPCR, in terms of lifetime maximization, only if the control parameter γ is well determined.

Li et al. [20] address the network lifetime maximization problem with max-min zP_{min}, an on-line message routing protocol. It first computes P_{min}, the minimum energy needed to transmit a packet from a source node to a destination node across all possible routes. It then uses max-min residual energy metric to pick a route, thereby balancing the load among different nodes, unless the cost is higher than zP_{min}, $(z \geq 1)$, in which case, it falls back to the min metric thus avoiding excessive energy consumption. The authors propose a centralized algorithm based on the gradient descent technique to determine the optimal value of z. Further on the same authors describe a distributed version of the algorithm [21], but it requires establishing synchronized mini slots at the MAC layer.

Shah et al. [22] consider the drawbacks of pure minimum energy routing for the survivability of the network. They propose a probabilistic route selection scheme to relieve workload of minimum energy routes. Their protocol is the following: given a set of routes between a source and a destination node, assign to each route the probability of being selected so that the minimum energy route has the highest probability. Then, forward packets on routes according to their probabilities. Note that routes with too much energy consumption, by analogy to the max min zP_{min} algorithm [20], are assigned zero probability and will never be selected. However, this protocol requires to explicitly transmit link cost information and to receive packets from all routes in order to compute the corresponding selection probabilities.

The above papers [17], [19], [21], [22] emphasize the idea of combining the minimum energy and max-min residual energy metrics to optimize the lifetime of sensor networks. However, the distributed nature of these protocols requires explicit transmission of the energy information which is counter productive with respect to energy optimization. Taking this overhead into account and inspired by other papers [23], [24], Guo [25] proposes a lightweight broadcast scheme for network lifetime maximization. His protocol encourages nodes with high residual energy to retransmit a broadcast message and works as follows. When a node receives a broadcast message, it delays the retransmission of this message to see if there is another node with higher residual energy. This delay is inversely proportional to the residual energy of the node. Guo's algorithm reduces the number of nodes forwarding a broadcast message without the overhead of explicitly exchanging the residual energy information, but it may miss some nodes in a sparse network. Besides, it does not implement the minimum energy nor the max-min residual energy routing.

VI. Conclusion

We have presented a synthetic mapping function that enables an existing min-delay routing protocol to be enhanced into an energy-aware routing that maximizes the lifetime of sensor networks. The resulting routing protocol combines the advantages of being min-delay and energy-aware. The energy-aware scheme prevents vulnerable nodes from being overused, which avoids early network partition and the min-delay scheme makes it possible for a node to select routes based on one routing message reception, which avoids consuming extra energy for receiving superfluous routing messages.

References

[1] A. Bachir and D. Barthel. Localized Max-Min Remaining Energy Routing for WSN using Delay Control. *Proceedings of the IEEE ICC*, Korea 2005.

[2] R. Szewczyk et al. Habitat Monitoring with Sensor Networks. *Communications of the ACM*, 47(6):34–40, 2004.

[3] E. Cayirci, T. Cuplu. SENDROM: Sensor Networks for Disaster Relief Operations Management. *MedHocNet*, June 2004.

[4] T. He et al. Energy-Efficient Surveillance Systems Using Wireless Sensor Networks. *Proceeding of the ACM MobiSys*, 2004.

[5] Ember Corporation. EM250 Single-Chip ZigBee/802.15.4 Solution, Data Sheet . 2005.

[6] J. Polastre, R. Szewczyk, D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. *Proceedings of IPSN/SPOTS*, pages 302–11, Los Angeles, CA, April 2005.

[7] Freescale Semiconductor. MC13192/MC13193 2.4 GHz Low Power Transceiver for the IEEE 802.15.4 Standard. *Rev. 2.8*, 04/2005.

[8] Freescale Semiconductor. HSC08 Microcontrollers, MC9S08GB60/D Data Sheet. *Rev. 2.3*, 12/2004.

[9] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley edition, 2005.

[10] J. N. Al-Karaki, A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, Dec 2004.

[11] Volkan Rodoplu and Teresa H.Y. Meng. Minimum energy mobile wireless networks. *IEEE JSAC*, 8(17):1333–44, August 1999.

[12] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. *Proceedings of the IEEE Infocom*, pages 1388–97, Anchorage, AL, April 2001.

[13] J. Wu, F. Dai, M. Gao, I. Stojmenovic. On calculating power aware connected dominating sets for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communications and Networks*, 1(4):59–70, March 2002.

[14] C. Intagonwiwat et al. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. on Networking*, 11(1):2–16, February 2003.

[15] J. Heidemann, F. Silva, and D. Estrin. Matching Data Dissemination Algorithms to Application Requirements. *Proceedings of the ACM Sensys*, pages 218–29, Los Angeles, CA, November 2003.

[16] A. Bachir, D. Barthel, M. Heusse, and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. *submitted for publication*.

[17] Toh, C.-K., Cobb, H. and Scott, D.A. Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks. *Proceedings of the IEEE ICC*, pages 2824–9, Helsinki, Finland, June 2001.

[18] S. Banerjee, A. Misra. Energy Efficient Reliable Communication for Multi-hop Wireless Networks. *Accepted for publication to Journal of Wireless Networks (WINET)*.

[19] A. Misra and S. Banerjee. MRPC: Maximizing network lifetime for reliable routing in wireless environments. *Proceedings of the IEEE WCNC*, Orlando, FL, March 2002.

[20] Qun Li and Javed A. Aslam and Daniela Rus. Online power-aware routing in wireless Ad-hoc networks. *Proceedings of the ACM Mobicom*, pages 97–107, Rome, Italy, July 2001.

[21] Q. Li and J. Aslam and D. Rus. Distributed Energy-Conserving Routing Protocols for Sensor Networks. *Proceedings of the IEEE HICSS*, Hawaii, January 2003.

[22] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *Proceeding of the IEEE WCNC*, 2002.

[23] S.Y. Ni et al. The Broadcast Storm Problem in Mobile Ad Hoc Network. *Proceedings of the IEEE/ACM Mobicom*, 1999.

[24] T. J. Kwon and M. Gerla. Efficient flooding with Passive Clustering (PC) in ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 32(1):44–56, January 2002.

[25] Xiaoxing Guo. Broadcasting for network lifetime maximization in wireless sensor setworks. *Proceedings of the IEEE SECON*, Santa Clara, CA, October 2004.

Appendix

A. Problem definition

We call ζ_{ik} the relative residual energy of node N_{ik}. Values ζ_{ik} are normalized in $[0,1]$, hence $0 \leq \zeta_{ik} \leq 1$ for all nodes.

Let us call ζ_k^- the node with least amount of residual energy on route R_k. Then,

$$\zeta_k^- = \min_{1 \leq i \leq |R_k|} \{\zeta_{ik}\} \tag{20}$$

The max-min residual energy routing selects the route with the largest ζ_k^-. Then, max-min residual energy routing selects the route R that satisfies:

$$R = \operatorname*{argmax}_{R_k \in \mathcal{R}} \{\zeta_k^-\} \tag{21}$$

Combining Eq. 20 and Eq. 21, we get

$$R = \operatorname*{argmax}_{R_k \in \mathcal{R}} \left\{ \min_{1 \leq i \leq |R_k|} \{\zeta_{ik}\} \right\} \tag{22}$$

Let us now examine min-delay routing. We call δ_{ik} the delay introduced by each node N_{ik} on route R_k. Route R_k experiences the total delay of δ_k,

$$\delta_k = \sum_{i=1}^{|R_k|} \delta_{ik} \tag{23}$$

Therefore, min-delay routing selects the route with minimum δ_i. The selected route, denoted by R', satisfies:

$$R' = \underset{R_k \in \mathcal{R}}{\operatorname{argmin}} \{d_k\} \tag{24}$$

Combining Eq. 23 and Eq. 24, we get

$$R' = \underset{R_k \in \mathcal{R}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{|R_k|} \delta_{ik} \right\} \tag{25}$$

Our goal is to make the min-delay routing select the route that satisfies the max-min residual energy metric, *i.e.* route R' matches route R. To make this possible, we propose to use function f to map the residual energies of nodes into an intentional delay. Our goal is to solve Eq. 22 by solving Eq. 25 on a suitable set of

$$\delta_{ik} = f(\zeta_{ik}) \tag{26}$$

B. Approximate Solution: Heuristic Functions

By choosing f to be strictly decreasing, we can rewrite Eq. 22 as:

$$R = \underset{R_k \in \mathcal{R}}{\operatorname{argmin}} \left\{ f\left(\min_{1 \le i \le |R_k|} \{\zeta_{ik}\} \right) \right\} \tag{27}$$

Matching Eq. 27 with Eq. 25 and replacing δ_{ik} by its values calculated in Eq. 26, we conclude that function f such that for all i in $1, \cdots, |R_k|$,

$$\sum_{i=1}^{|R_k|} f\left(\zeta_{ik}\right) = f\left(\min_{1 \le i \le |R_k|} \{\zeta_{ik}\} \right) \tag{28}$$

would meet our goal.

An approximate solution is obtained with f being a convex function $[0,1] \to [0,1]$. Indeed, if f is convex and decreasing, the minimal ζ_k^- along route R_k makes a dominant contribution to the sum to the left of Eq. 28, i.e. we have

$$f(\zeta_k^-) \gg \left(\sum_{i=1}^{|R_k|} f(\zeta_{ik}) - f(\zeta_k^-) \right), \tag{29}$$

and therefore

$$\sum_{i=1}^{|R_k|} f(\zeta_{ik}) \approx f\left(\min_{1 \le i \le |R_k|} \{\zeta_{ik}\} \right). \tag{30}$$