Reducing the Energy Drain in Multihop Ad Hoc Networks

Géraud Allard
INRIA
Domaine de Voluceau
Rocquencourt,
France.
Email: geraud.allard@inria.fr

Bernard Mans
Dept. of Computing
Macquarie University
Sydney, NSW 2109,
Australia.
Email: bmans@ics.mq.edu.au

Abstract—Numerous studies on energy-efficient routing for Multihop Ad Hoc Networks (MANET) look at extending battery life by minimizing the cost at the transmitting node. In this paper, we study the complexity of energy-efficient routing when the energy cost of receiving packets is also considered.

We first prove that, surprisingly, even when all nodes transmit at the same power, finding a simple unicast path that guarantees enough remaining energy locally at each node in the network then becomes an NP-complete problem. Hence we propose several heuristics based on Dijkstra’s shortest path algorithm in which we integrate the notion of remaining energy in order to satisfy flow requirements. We show with several simulations that these heuristics not only allow the computation of routes that save energy of nodes with low battery but also allow the network to handle more flows.

I. INTRODUCTION

The proliferation of wireless communicating devices has created a wealth of opportunities for the field of mobile computing. In the extreme case of self-organizing networks such as Multihop Ad Hoc Networks (MANET), it is still challenging to guarantee efficient communications when mobile nodes roam at will. As the topology changes arbitrarily and rapidly, the communication protocol must obtain dynamically the adequate routing whilst dealing with scarce resources and packet losses due to transmission errors, noise and interferences that characterize wireless networks.

Multihop Ad Hoc Networks (MANET) are unlike the well-studied cellular wireless systems that rely heavily on the robust structure of the physically connected stations. They are self-organizing entities that must distributedly choose how to interconnect in order to facilitate the communication within the network. This feature makes them attractive but increases the difficulty of the routing. A mobile node has to cooperate with other hosts to find routes and relay messages.

In addition, mobile nodes have a limited energy source (i.e., battery) and communication range (each message may “hop” several times from node to node before reaching its destination).

Here we are concerned with the energy consumed by the MANET protocols as a performance metric. We are interested in characterizing some of the energy-efficient features required by MANET protocols to route with minimum energy cost and extend the life of the energy source of each mobile node.

We focus our study on the fact that wireless traffic carried by neighboring nodes within $H_I$ hops, when $H_I > 0$, may interfere (contrarily to wireline networks for which each link is physically isolated from the other links even if they are attached to the same node, i.e., this case corresponds to $H_I = 0$). In this paper, we address the problem of energy-efficiency in MANET when the energy cost of packet receptions is also taken into consideration. We show that, even with simple assumptions such as a fixed or common transmission power, more problems become NP-hard, and some cannot be approximated in polynomial time within $n^\varepsilon$, $\varepsilon > 0$, provided that $P \neq NP$.

It is already known that the energy-optimal broadcast is an NP-hard problem when nodes may choose among different transmission powers (e.g., [2], [4]). The first novel element brought in Section III is the fact that finding a simple unicast path that guarantees enough remaining energy locally at each node is then an NP-complete problem (even when all nodes transmit at the same power).

Hence, in Section IV, we propose several heuristics based on Dijkstra’s shortest path algorithm in which
we integrate the notion of remaining energy in order to satisfy flow requirements. We show with several simulations that these heuristics not only allow computation of routes that save energy of nodes with low battery but also allow the network to handle more flows.

We introduce our motivations, model and notations in the following section.

II. ENERGY EFFICIENCY IN MANET

A. Motivation

The untethered, structureless and mobile nature of MANET nodes drastically limits their battery lifespan. Energy-efficient Multihop Ad Hoc Networks (MANET) look at extending battery life by minimizing the energy cost. The difficulty of such a problem lies in the fact that it involves several network layers. Three main approaches to alleviate this problem have been taken:

1) Modify the MAC layer (e.g., for IEEE 802.11);
2) Use an energy metric (e.g., the energy consumed per packet or per node) and design schemes that minimize such a metric (at the physical or link layer);
3) Find an optimal transmit power locally at each node to control some global properties (e.g., connectivity [2], [4], cone-based geometrical cover [13], constant stretch-factor of the shortest path).

As some devices have the ability to adjust power, most of the theoretical work has been achieved for the last item and has led to many NP-complete problems (e.g., energy-optimal broadcast [2], [4]) and approximation heuristics (e.g., broadcasting increment power [1]). There exist some implemented protocols that adaptively adjust power to optimize the network performance, by taking the smallest common power level which results in connectivity of the (thus bidirectional) network [11].

Several studies have also suggested energy dependent participation protocols where nodes can evaluate and elect their involvement in the network based on the remaining level of their battery (e.g., [12]).

To our knowledge, almost all studies have focused on transmitting energy as the sole energy cost. However, as pointed out in Feeney and Nilsson’s seminal work [6] regarding the energy consumption of MANET current interfaces, there are other substantial costs. These are mainly due to the differences between MANET mode and base station mode. In MANET, there is no infrastructure and the nodes cannot go into an energy-saving (“sleep”) mode as the intrinsic ad hoc nature of the network forces them to remain in a “ready to receive” state (i.e., the so-called “idle” state). Receiving any packet is costly because the conservation of energy occurs only after discerning that an incoming packet is not intended for a node. The node is still very much active not idle. (For example, with IEEE 802.11b, the energy consumption of the promiscuous receive mode and the discard mode is almost the same).

For instance, in [6], the Lucent IEEE 802.11 2Mbps WaveLAN PC Card power consumption characters were measured as follow:
- Idle power at 843mW;
- Receive power at 967mW;
- Transmit power at 1327mW.

As overall consumption is dependent of the mobile hardware, we restrict our study to considering “unit” of transmission and reception energy costs of the wireless network interface.

There is a large body of work to reduce substantially this “idle state” cost (e.g., at the MAC layer) as its rate of energy consumption is high and only slightly lower than the reception cost. Also, there has been work on how to schedule sleep periods among nodes that have limited battery level so that the network can still function and yet allow nodes to disconnect to save energy (e.g., [3], [14]). It is expected that the difference of cost between reception and idle states will rapidly increase.

However the tenet that the reception cost is larger than a substantial fraction of the transmission cost is likely to remain. That is, defining \( t_i, r_i \) and \( \rho_i \) as the idle state, the transmission and reception interface energy costs for a node \( i \), respectively, we can define \( t_i(=\tau_i-\ell) \) and \( r_i(=\rho_i-\ell) \) as the transmission and reception interface’s energy net costs for a node \( i \), respectively. Thus, we may assume that for some small integer constant \( k \):

\[
\frac{t_i}{k} > r_i \quad (1)
\]

If \( k \) is smaller than the number of neighbors of a transmitting node, the passive receptions from non-targeted neighbors may cost more than the transmission from the source and the reception from the targeted neighbors. For example, with the Lucent specifications above, for which the net energy costs are \( t_i = 484mW \) and \( r_i = 124mW \) respectively, the network with an average degree of five neighbors will consume much more energy through receptions than through transmissions. It can lead to a detrimental drain of neighboring nodes with low battery even if they did not need to be involved in the transmission.
Here, we study the impact of link interferences on energy-efficient routing in MANET by considering the energy cost of transmitting and receiving packets (even when all nodes transmit at the same power). In the rest of the paper, we will assume that \( t_i \) and \( r_i \) are fixed at each node \( i \).

### B. Model and notations

For clarity, and without loss of generality, we introduce a simple graph model (which does not impact on our complexity results). We introduce several notations. For other basic graph-theoretical definitions, see Diestel [5].

We consider an undirected graph \( G(V, E) \) modeling a wireless network. Link \((i, j)\) means that nodes \( i \) and \( j \) can communicate. We assume that all links are bidirectional. Node \( i \) has remaining energy capacity \( e_i \) and can send a message to its neighbors at transmitting interface’s energy cost \( t_i \) if \( t_i \leq e_i \). Note that this assumption is not a convenient modeling simplification but rather a technical requirement as bidirectional links are used to achieve unicast transmission with 802.11b.

Let \( N(i) \) be the neighbors of node \( i \). Denote by \( N_i(h) \) the set of nodes that are at most \( h \) hops away from node \( i \) (including node \( i \)), e.g., \( N_i(1) = \{i\} \cup N(i) \).

A transmission interferes with nodes that are within \( H_I \) hops, from the transmitter, depending on the signal to noise ratio. When node \( i \) transmits a packet to its neighbor \( j \), all nodes in \( N_i(H_I) \) can receive the packet and may need to process it.

Consuming \( t_i \) units of interface energy (at node \( i \)) for transmission from \( i \) to node \( j \) induces the consumption of \( r_k \) units of interface energy at each receiving node \( k \) within \( H_I \) hops from \( i \) (including \( j \)).

In this context, it is easy to compute the worst-case per-packet cost for passing a packet along a path \( \Pi \) from a source \( s \) to a destination \( d \):

\[
\sum_{i \in \Pi - \{d\}} (t_i + \sum_{j \in N_i(H_I)} r_j)
\]

(2)

Let \( K_{\Pi}(j) = \{i \in \Pi - \{d\} | j \in N_i(H_I)\} \), that is, the nodes on the path \( \Pi \) that are within \( H_I \) hops from a node \( i \) of the path \( \Pi \). Let \( k_{\Pi}(j) = |K_{\Pi}(j)| \). The per-node cost at a node \( j \) (for passing a packet along a path \( \Pi \) from a source \( s \) to a destination \( d \), possibly not including \( j \)) is defined as:

\[
\begin{align*}
r_j \quad & k_{\Pi}(j) + t_j & \text{if } j \in \Pi, \\
r_j \quad & k_{\Pi}(j) & \text{otherwise.}
\end{align*}
\]

(3)

The second case formalizes the fact that the battery of a node in the vicinity of some transmissions may be drained rapidly without transmitting once.

This also emphasizes the fact that the per-packet cost and per-node cost lead to two distinct problems. When all the nodes transmit at the same energy cost, it is easy to find a unicast path that minimizes the per-packet cost (e.g., by using a Dijkstra’s shortest path algorithm running in polynomial time). However, minimizing the per-node cost is not trivial and becomes NP-complete if some minimum remaining energy is required, as we will prove in the next section.

### III. Per-node energy efficiency and NP-completeness

Several studies have suggested energy-dependent participation protocols where nodes can evaluate and elect their involvement in the network based on the remaining level of their battery (e.g., [12]).

This yields the Remaining Energy problem.

#### A. Local minimum remaining energy

Within our model, this raises the natural question of avoiding the energy drain of nodes that are not necessary for the success of the multihop communication.

In particular, we may want to ensure that a flow is not detrimental to the remaining energy of some neighboring low-level nodes. We say that the flow path is tolerable if this operation leaves the remaining energy \( e_j \) of all nodes \( j \) in the network above a minimum tolerable capacity level \( c_j \). (For the critical functioning of each individual node, one may require that the remaining energy is always above a non-zero fraction of the nominal capacity."

Assume that node \( i \) has minimum tolerable capacity \( c_i \). We must have

\[
c_i \leq e_i, \quad \forall i \in V
\]

(4)

We define the energy-efficient “Path with Remaining Energy problem” as follows.

**Definition 1:** The Path with Remaining Energy problem (RE) is defined as:

**Instance:** A Graph \( G = (V, E) \), two vertices \( s \) and \( t \) from \( V \), the remaining energy \( e_i \) and a minimum tolerable capacity \( c_i \), both in \( \mathbb{R} \), for each vertex \( i \) from \( V \).

**Question:** Is there a simple path from \( s \) to \( t \) in \( G \) that satisfies the constraint \( c_i \leq e_i, \quad \forall i \in V \)?

We now prove that, because of the basic constraint (4), this problem is NP-complete.
B. NP-completeness proof

We first recall the definition of the Path with Forbidden Pairs problem (PFP).

**Definition 2:** The Path with Forbidden Pairs problem (PFP) (see GT54 in [7]) is defined as:

**Instance:** A Graph \( G = (V,E) \), two vertices \( s \) and \( t \) from \( V \), and a collection \( C = \{(a_1,b_1),\ldots,(a_m,b_m)\} \) of pairs of vertices from \( V \).

**Question:** Is there a simple path from \( s \) to \( t \) in \( G \) that contains at most one vertex from each pair in \( C \)?

The PFP problem is known to be NP-complete (see GT54 in [7]).

**Theorem 3:** The Path with Remaining Energy problem (RE) is NP-complete.

**Proof:** For clarity, we first prove the theorem in the case \( H_1 = 1 \) (i.e., only neighbors of the node are receiving the packets transmitted by this node).

It is easy to see that \( RE \in NP \), because a nondeterministic algorithm needs only to guess a path and check in polynomial time that the constraint \( c_i \leq e_i \), \( i \in V \), is valid. (Of course, checking the constraint for the vertices of the path and for their neighbors is sufficient.)

We now give a polynomial reduction from this problem to the Path with Forbidden Pairs problem (PFP).

Without loss of generality, we assume that for any node, a reception costs one unit of energy (i.e., \( r_i = 1 \) for any node \( i \)). We first transform an instance \( (G = (V,E), s,t,C) \) of the PFP problem in an instance \( (G' = (V',E'), s,t,p,c) \) of the Remaining Energy problem by formally defining:

- \( V' = V \cup \{v_{xy} | (x,y) \in C \} \),
- \( E' = E \cup \{(x,v_{xy}), (y,v_{xy}) | (x,y) \in C \} \),
- \( s \) and \( t \) are unchanged,
- \( c_i \), the minimum tolerable capacity at any node \( i \) is set to an arbitrary non-negative value,
- \( e_i \), the remaining energy set to:

  \[(i) \quad e_i = c_i, \quad \text{if } i \in \{v_{xy} | (x,y) \in C \} \quad \text{and} \quad (x = t \text{ or } y = t) \},
  \[(ii) \quad e_i = c_i + 1, \quad \text{if } i \in \{v_{xy} | (x,y) \in C \} \quad \text{and} \quad (x \neq t \text{ and } y \neq t) \},
  \[(iii) \quad e_i = c_i + |V|, \quad \text{otherwise.} \]

Informally, \( G' \) contains all the vertices of \( G \) as well as \( m \) new vertices, each representing a forbidden pair. Let us define \( F \) as the set of \( m \) vertices in \( G' \) representing each of the forbidden pairs of \( C \). Each vertex of \( F \) is only connected to its two respective “forbidden” vertices and is assigned an actual energy equal to 1 more than the capacity (i.e., only one “forbidden” node can transmit) or to the capacity (if the destination is part of the forbidden pair). All other vertices are assigned the remaining energy equal to the capacity plus \(|V|\), that is the remaining energy is larger than any possible energy cost induced by a given established path. Of course, without loss of generality, we assume that \((s,t)\) is not a forbidden pair.

We now prove that a solution of this instance of the \( RE \) problem is a solution if and only if it is a solution for the original instance of the PFP problem. It is easy to see that a solution path \( \Pi \) from \( s \) to \( t \) for the \( RE \) problem in \( (G' = (V',E'), s,t,p,c) \) does not include any of the vertices of \( F \), as each vertex \( v \) of the path (except the destination \( t \)) requires to decrement \( e_v \) by at least 2. Hence this path \( \Pi \) is also a path \( \Pi' \) in \( G \). Furthermore, none of the forbidden pairs are included in the solution path. Otherwise there would exist such a pair of vertices \( a \) and \( b \) (with \( (a,b) \in C \) that both belong to the solution path. This would imply that the vertex \( x_{ab} \) receives more than 2 transmissions along the path within 1 hop and that the basic energy constraint for the vertex \( x_{ab} \) is not valid as it will decrement \( e_{x_{ab}} \) by 2 and make it (see (ii)) equal to an intolerable \( e_{x_{ab}} - 1 \), thus leading to a contradiction. In the case that one node of the forbidden pairs is the destination, the other cannot transmit without decreasing the remaining energy \( e_{x_{ab}} \) by 1 which is then smaller than \( c_{x_{ab}} \) (by (i)), also leading to a contradiction. Hence a solution for the \( RE \) problem in \( (G' = (V',E'), s,t,p,c) \) is a solution for the instance \( (G = (V,E), s,t,C) \) of the PFP problem.

Conversely, given a solution path \( \Pi' \) for the instance \( (G = (V,E), s,t,C) \) of the PFP problem, we can verify that the path \( \Pi' \) is a feasible solution path for the \( RE \) problem in \( (G' = (V',E'), s,t,p,c) \). Obviously, \( \Pi' \) is a simple path of \( G' \) as \( G \) is a subgraph of \( G' \). Hence, we just need to verify that the remaining energy induced by the path respect the energy constraints. It is clear from the reduction that only the nodes of \( F \) may jeopardize the feasibility of the solution, as they may not have sufficient energy. Again, none of them belong to the path. A vertex of \( F \) can be a neighbor of a vertex of \( \Pi' \), incurring an energy cost of at least 1 (unless it is a neighbor of the destination). However, a vertex of \( F \) cannot be a neighbor of two nodes in the paths as it can only be neighbor of its forbidden pair which would contradict the feasibility of \( \Pi' \) for the instance \( (G = (V,E), s,t,C) \) of the PFP problem.

As the proof follows a polynomial reduction from this problem to the Path with Forbidden Pairs problem (PFP), the Path with Remaining Energy problem (RE) is NP-complete with \( H_I \geq 1 \).
This theorem proves that looking for “any” path that satisfies the basic constraint 4 is not tractable. For other considerations, we may also request that each path reserved is a shortest path. Using the result of Kann [9] on the shortest feasible PFP, it is easy to deduce the following corollary.

**Corollary 4:** Shortest Path with Remaining Energy problem (SPRE) remains NP-complete and is also hard to approximate (i.e., NPO PB-complete).

**Proof:** Variants of PFP problem exist where a measure is the length of the path (i.e., the number of edges in the path): the shortest feasible PFP and the longest feasible PFP problems. Not surprisingly both variants are NP-complete and also NPO PB-complete [9], i.e., in polynomial time, they cannot be approximated within $n^\varepsilon$ for some $\varepsilon > 0$, where $n$ is the size of the input, provided that $P \neq NP$.

We could introduce some variants of these optimization problems while respecting the basic constraint (e.g., Least Remaining Energy, Total Consumed Remaining Energy), but all are NP-complete and one need to introduce some heuristics. It is worth emphasizing that, even with a more detailed reception model, with mobility or with localization, the NP-completeness of each problem still holds.

### IV. Heuristics

In order to cope with the NP-completeness of the Path with Remaining Energy problem, we propose, in this section, two heuristics based on Dijkstra’s shortest path algorithm in which the node remaining energy is introduced.

The main aim of these heuristics is to compute routes that save energy of low remaining energy nodes. In other words, when computing a route we want to choose forwarding nodes that have a high battery level according to the other routers. Moreover, as we discussed in section II, neighbors of a forwarding node will also consume energy to receive packets even though they are not the destination node. The second heuristic we will introduce in this paper focuses on this particular point.

Next it is important to note that routes computed with these heuristics may be longer than those derived from Dijkstra’s shortest path algorithm. Longer paths can actually be used to bypass some routers that have low remaining energy. On the other hand, routes should not be too long: the overall energy consumption of a packet routed through a long path will be higher as the number of intermediate transmissions will grow. Thus there must be a trade off between energy saving and routes length. For this reason, we designed our heuristics using Dijkstra’s shortest path algorithm.

Lastly we will see that using such heuristics allow the network to handle more flows than the common Dijkstra’s algorithm. In other words, these heuristics not only compute routes that satisfy energy requirements of forwarding nodes but also choose paths in such a way that more connections can be accepted.

#### A. Model

In the following model we consider that each node knows: the topology of the ad-hoc network and the remaining energy of the other nodes. This can be easily carried out if we suppose that the underlying routing protocol works on a proactive basis. As proactive routing protocols use link state information, nodes actually have knowledge of the topology of the network. Moreover, if a proactive protocol (e.g., from IETF Manet group [8]) is used, one can easily add new packets whereby nodes periodically announce their local remaining energy. Thus, if these messages are broadcasted within the network, each node should know the remaining energy of the other routers. This can be done by an admission control mechanism set at each node to handle applications requests. One can also take advantage of multipoint relaying mechanism to reduce the flooding overhead of control messages.

An admission control protocol can be set as follows:

- whenever a node $a$ requires a route for a destination node $b$, $a$ computes a path $\Pi$ using one of the heuristics presented in the next subsections;
- an admission control is performed to verify that $\Pi$ meets the energy requirements of the network;
- if the admission control succeed, packets can be send through this path;
- otherwise the flow is rejected.

Notice that the routing decision is taken at the originator node. A routing mechanism similar to source routing must be set to ensure that packets are transmitted through the correct path. For example, if two paths of the same length exist between two nodes, hop-by-hop routing may lead to a wrong routing decision and then energy requirements may not be guaranteed any more. Moreover when topology or battery level changes are detected in the network, the source should guarantee that energy requirements are still met for this new configuration and then a new admission control should be performed.
B. Heuristic based on forwarding nodes remaining energy

Denote by \( V \) the set of nodes (with \( |V| = n \)) and let \( r(i) \) be the transmission range of node \( i \). Let \( d(i, j) \) be the distance between nodes \( i \) and \( j \). Nodes \( i \) and \( j \) can hear each other if and only if \( d(i, j) \leq \min(r(i), r(j)) \). To make our presentation clearer, we will consider, in the following of this study, that \( r(i) = r, \forall i \) (i.e., all links are symmetric links). Let \( e(i) \) be the remaining energy of node \( i \) and denote by \( W(i) \) the weight associated with node \( i \). The aim of the common Dijkstra’s algorithm is to find a path \( \Pi \) from a source \( s \) to a destination \( d \), that minimizes \( W_\Pi = \sum_{i \in \Pi - d} W(i) \). Notice that, in the common Dijkstra’s algorithm, weights are usually associated with links but since battery level is a node-related characteristic, we no longer associate weights with links but with nodes.

The main idea for this heuristic, named \( H_1 \), is to find a route that minimizes the energy drain within forwarding nodes. In other words we want to maximize the remaining energy in the forwarding nodes. If we associate each node \( i \) with a weight equal to the inverse of its remaining energy \( W(i) = \frac{1}{e(i)} \), finding a path \( \Pi \) that maximizes remaining energy of forwarding nodes amounts to using Dijkstra’s shortest path algorithm with weights \( W(n) \).

C. Heuristic based on forwarding nodes neighborhood remaining energy

Heuristic \( H_1 \) is only concerned with nodes that are located on the path, but it is important to notice that, as we discussed in Section II, neighbors of a forwarding node must also consume energy to receive packets and analyze frames header.

The idea for this second heuristic, named \( HN_1 \), is to integrate the values of neighbors remaining energy in the weight of forwarding nodes. Denote by \( \mathcal{N}(i) = \{ j \in V \mid d(i, j) \leq r \} \) the neighbor set of node \( i \). Let us define a new weight \( W(i) = \sum_{j \in \mathcal{N}(i), e(i) \neq 0} \frac{1}{e(j)} \). If a node \( i \) has a low battery level, weights \( W(j) \) of nodes located in \( \mathcal{N}(i) \) will have an immediate impact since \( W(j) \) will tend to infinity when \( e(i) \) decreases. Although this heuristic is a slightly more time-consuming, especially for dense networks where the average degree is large, we will see in the next section that it leads to better performances.

D. Simulation results

We conducted several simulations to compare Dijkstra’s algorithm to \( H_1 \) and \( HN_1 \) heuristics. We randomly placed \( N \) \( (1000 \leq N \leq 1500) \) nodes within a \( 1000m \times 1000m \) flat area. Different transmission ranges between 50m and 300m have been considered. Each node has 200,000 units of energy at the beginning of the simulation. For each simulation we try to insert 1000 flows as follows:

1) a source and a destination are randomly chosen,
2) a route is computed for this flow,
3) an admission control is performed along the path to check whether energy requirements are satisfied.

The flow is then accepted or rejected accordingly.

This procedure is reiterated for each flow insertion.

Figure 1 shows results of flows insertion for \( n = 1300 \) and \( r = 100 \) for Dijkstra’s shortest path algorithm, \( H_1 \) and \( HN_1 \). All routes are accepted until flow 450 and then Dijkstra’s algorithm starts to reject routes whereas \( H_1 \) and \( HN_1 \) accept all routes until flow 700 and flow 850 respectively. At the end of the simulation, the following admission rate results were observed:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Admission rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>( HN_1 )</td>
<td>90%</td>
</tr>
<tr>
<td>( H_1 )</td>
<td>85%</td>
</tr>
<tr>
<td>Dijkstra</td>
<td>65%</td>
</tr>
</tbody>
</table>

Figures 2, 3 and 4 represent the distribution of remaining energy as a function of inserted flows. We can see that Dijkstra’s algorithm tends to quickly spread the distribution over all values of energy. Some nodes with low remaining energy becomes unable to handle any traffic. On the contrary, \( H_1 \) and \( HN_1 \) tends to concentrate the distribution around a medium value. This behavior can be clearly seen on Figure 4. This can be explained by the fact that \( H_1 \) and \( HN_1 \) always try to choose nodes with the highest battery level.
The previous results were related to a specific network configuration where \( n = 1300 \) and \( r = 100 \), but the behavior we have depicted is similar for the other network configurations. Figure 5 shows the total number of inserted flows as a function of a density parameter named \( \Delta \) defined as follows: \( \Delta = \frac{\delta}{|V|} \) where \( \delta \) is the average degree of the network and \( |V| \) the number of nodes. In other words \( \Delta \) represents the fraction of nodes that are located in the neighborhood of a node \( n \). Notice that \( \Delta \) tends to 1 when the density of the network increases. Figure 5 shows that heuristics \( H_1 \) and \( HN_1 \) allow the network to handle more traffic than Dijkstra’s algorithm for every network configurations. It is also important to notice that \( HN_1 \) delivers better performance than \( H_1 \) since it performs a more accurate evaluation of the remaining energy in the network. Lastly we can make the following remark: when the network density grows (i.e., \( \Delta \) tends to 1), each packet transmission tends to be heard by every nodes and then the behavior of \( H_1 \) and \( HN_1 \) tends to be equivalent to Dijkstra’s algorithm. This can be verified in Figure 5.

Finally, as we pointed out at the beginning of this section, routes computed with heuristics \( H_1 \) and \( HN_1 \) should not be too long in order to avoid too much retransmissions which could lead to increasing energy drain of the nodes. Figures 6 and 7 show, for each flow accepted by the three algorithms, the length of the routes computed by \( H_1 \) and \( HN_1 \) compared with the Dijkstra’s shortest paths. We can see that some routes computed with \( H_1 \) and \( HN_1 \) are somewhat longer. Actually route lengths sometimes need to be increased to bypass nodes with low battery level. This behavior can be especially noticed for \( HN_1 \) since it tends to bypass...
the whole neighborhood of a low battery level node and attempts to distribute the overall energy costs fairly over the network. It is worth emphasizing that the previous simulations results have shown that although routes are slightly longer than those derived from Dijkstra’s algorithm, heuristics $H_1$ and $HN_1$ allow more routes admissions (i.e., consume less overall).

V. CONCLUDING REMARKS

In this paper, we first proved that when a realistic model of reception energy costs is considered, problems such as finding a simple unicast path that guarantees enough remaining energy locally at each node in the network or finding a virtual backbone that minimizes the overall energy cost during broadcast then become NP-complete problems. This, surprisingly, remains true even when all nodes transmit at the same transmission power.

We proposed several heuristics based on Dijkstra’s shortest path algorithm in which we integrate the notion of remaining energy in order to satisfy flow requirements. We showed with several simulations that these heuristics not only allow the computation of routes that save energy of nodes with low battery but also allow the network to handle more flows.

Acknowledgments. Part of this work [10] was done while Bernard Mans was the INRIA-Hitachi 2003 Chair financed by Hitachi and hosted at INRIA-Rocquencourt, Domaine de Voluceau, 78153 Le Chesnay Cédex, France.

REFERENCES