New power-aware routing protocol for mobile ad hoc networks

Djamel Djenouri*

Basic Software Laboratory,
CERIST Center of Research,
Algiers, Algeria
E-mail: ddjenouri@mail.cerist.dz
*Corresponding author

Nadjib Badache

LSI, Computer Science Department,
USTHB University,
Algiers, Algeria
E-mail: badache@lsi-usthb.dz

Abstract: Since devices used in wireless mobile ad hoc networks are generally supplied with limited autonomous resources, energy conservation is one of the most significant aspects in these networks. Recent studies show that the energy consumed for routing data-packets in mobile ad hoc networks can be significantly reduced compared with the min-hop full-power routing protocols. One of the promising mechanisms proposed in literature to reduce the energy consumption is the transmission power control. In this paper, we define new routing metrics to strike a balance between the required power minimisation and batteries freshness consideration. We also define a new technique which allows the distribution of the routing task over nodes. Using these metrics and techniques we derive from DSR [2] a new power-aware and power-efficient routing protocol, whose performance is analysed by simulation in different situations of mobility and network load.

Keywords: wireless mobile ad hoc network; routing protocol; power awareness; energy efficiency; GloMoSim simulation.

Reference to this paper should be made as follows: Djenouri, D. and Badache, N. (2006) 'New power-aware routing protocol for mobile ad hoc networks', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 1, No. 3, pp.126–136.

Biographical notes: Djamel Djenouri obtained his Engineering Degree in Computer Science and his Master's Degree in Computer Science from the University of Science and Technology USTHB (Algiers), respectively in 2001 and 2003. He is currently a PhD student at USTHB and a Research Assistant at the CERIST centre of research in Algiers. He works mainly on ad hoc networking, especially on the following topics: Security, Power Management, Routing Protocols, MAC Protocols.

Nadjib Badache obtained the Engineer Degree in computer science from University of Constantine, Algeria in 1978, and the Master Degree from USTHB University in 1982. In 1995, he joined The ADP research group at the IRISA institute in France, where he prepared a PhD Thesis on causal ordering and fault tolerance in mobile environment. He obtained his PhD from USTHB in 1998. He is currently a Professor at the computer science department of USTHB, where he is also the head of the LSI laboratory. He is author of many papers, and he supervised many thesis and research projects. His research interests are distributed mobile systems, mobile ad hoc networks and security.

1 Introduction

For mobility and portability constraints, devices used in mobile ad hoc networks are generally supplied with lightweight, limited batteries. Moreover, each node acts as a router and relays packets initiated from other nodes, which adds extra tasks to these devices in addition to the executions of users' applications. Therefore, power consumption becomes a very important issue in this new environment.

Recent studies show that energy efficient routing protocols for ad hoc networks may be designed from the existing protocols, by adding new mechanisms and techniques (Doshi and Brown, 2002b; Singh et al., 1998), and that the reactive approach is more adaptive than the proactive one (Brown et al., 2001; Badache et al., 2003).

One of the promising mechanisms proposed for reducing the energy consumption is the transmission power control, which consists of using adaptive transmission powers according to the distance separating the transmitter and the receiver, instead of using a fixed full power. Doshi and Brown (2002a) have proposed an implementation of this mechanism, where they have modified DSR (Dynamic Source Routing) (Johnson and Maltz, 1996) to obtain new power efficient versions. In Singh et al. (1998), Singh et al. have proposed general battery-aware metrics which can be used by routing protocols. However, all these metrics do not take advantage of the power control usage. On the other hand, all DSR versions proposed in Doshi and Brown (2002a) do not consider nodes' batteries states. Instead, routes are chosen by minimising the required energy on the available routes, regardless of either batteries' states of nodes on the routes or states of the communication channels, which may lead to an overuse of a nodes subset. Consequently, batteries of these nodes will lose rapidly their capacities before the others, resulting in a possible network partition. To ensure a long life to all nodes and to avoid the network partition, we should take the batteries states into account. We should also distribute the routing task over nodes, to avoid overusing some of them.

In this paper, we define new metrics aiming at resolving the tradeoff between the batteries' freshness and the total required power minimisation when selecting routes and defining their optimality. We also define a new technique which allows us to take advantage of all available routes, and disperses the routing task over several nodes. Using these metrics and this technique, we derive from DSR a new power-aware and efficient protocol. The remainder of this paper is organised as follows: we overview in the Section 2 the current route selection strategies, focusing on their drawbacks, followed in the Section 3 by a presentation of the new metrics and the new technique we propose. Section 4 is a presentation of our protocol derived from DSR. Sections 5 and 6 are devoted to our protocol's performance evaluation. In the former, we describe the simulation environment and in the latter, the results of our simulation are presented. Finally, Section 7 concludes the paper and summarises the future work.

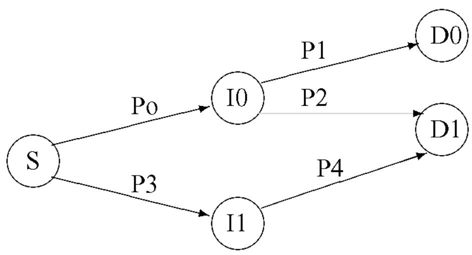
2 Current route selection strategies

2.1 Minimising the total required power

This strategy needs the power control technique to be employed, its purpose is to minimise the total required power when sending each packet. When using the power control technique, a cost may be associated to each link as the power required on it; the optimal route is the one that minimises the sum of these costs. The problem with this strategy is that it does not take into account batteries' states, which might result in an overuse of a nodes' subset as shown in the following example:

We consider the stationary network represented in Figure 1, where weights represent the powers required on the links. We assume there is a session between node s and node D_0, and another between s and D_1. And we assume that $P_0 + P_2 < P_3 + P_A$. When using this strategy, node I_0 will be used to route packets both for D_0 and D_1, while node I_1 will not be used at all. This way, node I_0 may lose its battery capacity before the others; then, a soon network partition can take place (node D_0 will not be reachable anymore). This partition would be avoided if node I_1 was used.

Figure 1 Stationary network



2.2 Routes containing the freshest batteries

The aim of this strategy is to select routes that contain the freshest batteries. For this purpose, a cost representing the battery state is associated with each node and the route that minimises the total sum of these costs is considered as the most optimal. In Singh et al. (1998), a definition of the cost function proportional to the battery voltage has been proposed. The cost of node i is given by:

$F_i(z_i) = 1/(1 - g(z_i))$. Where z_i denotes the measured voltage (that gives a good indication of the energy used thus far) and $g(z_i)$ $(0 < g(z_i) < 1)$ is the normalised power (in percentage) consumed for the voltage z_i. For a given value of z_i, $g(z_i)$ can be obtained from the discharge curve that features the battery (Gold, 1997).

The major drawback of such a strategy is that it does not consider links' costs, especially when batteries states are close to each other. In this case, the shortest routes are selected regardless of their costs, causing important waste of energy. This metric minimises the difference in energy consumption between nodes, but does not ensure long life for the batteries. In other words, it ensures that nodes remain alive together, but not for the longest period.

3 New power-aware routing strategy

3.1 Metrics

As illustrated in the previous section, each of the two power-aware strategies has a drawback. We think that both links' costs and batteries' states should be taken into account when selecting routes. Therefore, a tradeoff between the batteries' freshness and the required power minimisation should be overcome when selecting routes. For this purpose, we define new metrics on which the routing is based. First, we assume without loss of generality that the network is stationary, and we try to resolve the problem in a centralised manner at a given time *t*.

Let (S, L, P, E) be a quadruple such as:

S: Vertices set

L: Arcs set, $L = \{(s_i, s_j)/s_i \in S \text{ and } s_j \in S\}$, such that (S, L) is the graph representing the network topology at the time *t*.

P: A function which associates to each element (s_i, s_j) of L, a value in R (the real numbers set), called the *weight* of (s_i, s_j).

E: A function which associates to each element s from S, a value in R, called the *state* of s. On a given route $C = I_0, I_1, \dots, I_{n-1}$ we propose the following metric:

$$M_c = \sum_{i=0}^{n-2} \alpha \times E(I_i) + \beta \times P(I_i, I_{i+1}). \tag{1}$$

where α is the states *rate*, β is the links *rate* (α, $\beta \in [0,1]$), such that $\alpha + \beta = 1$.

The aim is to choose the route C which minimises M_c.

The function E should reflect batteries states, while the function P should reflect transmission power costs over links. Let E be the following function:

$$E(s_i) = 1/(1 - \text{eng}(s_i)), \tag{2}$$

where *energy* $(s_i) \in [0,1]$ is the rate of the energy consumed by node s_j. This function is monotone in the interval [0,1], i.e., more the node's consumption rate increases, the more its state value increases.

For P, we propose the following function:

$$P(s_i, s_j) = 1/(1 - \text{PowRate}(s_i, s_j)). \tag{3}$$

Such that,

$$\text{PowRate}(s_i, s_j) = \frac{\text{Pow}(s_i, s_j)}{\text{MaxPow}(s_i)},$$

where $\text{Pow}(s_i, s_j) \in [0,1]$ is the required power over the link (s_i, s_j), $\text{MaxPow}(s_i)$ is the full power of the node s_i (the power that allows it to cover its power range). Note that P increases with *PowerRate.*

When using this metric (M_c), the more α increases, the more the routes containing fresh batteries are favoured, and the more it decreases, the more routes that minimise the total transmission energy cost are favoured. Hence, we propose the following strategy. The more the difference between the nodes' batteries states increases, the more α should be increased. And the more the nodes' batteries states become closer, the more it should be decreased. This way, batteries' states are constantly well balanced while considering the total power when transmitting data packets.

Definition 1: We define the difference between batteries states at the time *t* by:

$$\text{EngDiff} = \max_S(eng(s_i)) - \min_S(eng(s_i)).$$

It is the difference between the maximum rate of the energy consumed by the network's nodes and the minimum one, note that $\text{Engdiff} \in [0,1]$.

It remains to find an increasing function for α: $\alpha(\text{EngDiff}):[0,1] \rightarrow [0,1]$.

We propose the following functions, where α_0 is an initial value:

$$F1 = (1 - \alpha_0) \times EngDiff + \alpha_0$$

$$F2 = \sqrt{(1 - \alpha_0)^2 \times EngDiff + \alpha_0}.$$

The two functions are monotone increasing in the interval [0,1], but they differ from each other in their way of increasing. The latter is more increasing in the interval [0,1] than the first one. What is the way of increasing that gives more efficiency? Evaluating the performance of these two functions in a further section may provide an answer to this question. Using equations (1–3), we rewrite the metric Opt_c representing the optimality of the route $C = I_0, I_1, \dots, I_{n-1}$ connecting node I_0 to node I_{n-1} as follows:

$$\text{Opt}_c = \sum_{i=0}^{n-2} \frac{\alpha}{1 - \text{eng}(I_i)} + \frac{1 - \alpha}{1 - \text{Pow}(I_i, I_{i+1}) / \text{MaxPow}(I_i)}, \tag{4}$$

where: $\alpha = F1$ or $\alpha = F2$.

The optimal path corresponds to the minimum value of Opt_c. At time *t*, $\text{Pow}(s_i, s_j)$ value is known for each link, the value of $\text{eng}(s_i)$ is known for each node, thus both EngDiff and α can be computed. Thus, the optimal path connecting any pair of nodes at the instant *t* can be computed (using Djikstra algorithm for instance). In time, values of $\text{eng}(s_i)$ change, the computation of parameters and routes must consequently be done again. Using this approach, node I_0 of the previous example will not be used all along the sessions between node s and nodes D_0 and D_1. Because when I_0 starts consuming its battery capacity for routing packets, $E(I_0)$ increases, and the route (s, I_0, D_1) becomes less optimal than (s, I_1, D_1), hence the last one will be used for routing packets to D_1. This way, a soon network partition is avoided.

Thus far we have proposed new metrics that have been explained using a centralised solution. It is obvious that in practice, no node can have an accurate view of the network topology. Still, these metrics can be employed by any distributed routing protocol, as we will see in the next section.

3.2 New data dispersal technique

Many reactive routing protocols, such as DSR (Johnson and Maltz, 1996), are multiroutes, i.e., after a route discovery, many routes may be found. But in DSR, just one route (considered to be the optimal) among the available ones is chosen. All the others are considered only as alternatives, and they are not used until the optimal route fails.

Data dispersal had been first proposed in distributed systems for security and load balancing aims by Rabin (1989). In Rabin's algorithm, largely used in literature, additional redundancy is added to data packets and the same amount of data is sent on each route.

We think that this strategy could be projected to our context, and employed in order to exploit a maximum number of the available routes, thereby dispersing data load over a maximum number of nodes. But route optimality should be considered in such a way to transmit more packets on the most optimal routes. To apply data dispersal while considering routes optimality, we suggest a mechanism analogous to the one used by schedulers of operating systems (such as UNIX). Routes are analogous to queues, packets to processes, and the routing protocol to the scheduler.

Let *nbrpaths* be the number of routes available that relay a source node *s* to a destination *D*. We assume that these routes are ordered according to their optimality (route 0 is the most optimal one). And let *nbrpackets* be the number of packets to be sent, stored in a buffer.

In order to use the maximum number of routes and to use optimal routes more than the less optimal ones, we propose to transmit on each route *i*, the following number of packets:

$$nbrpaquets_i = \left\lceil \frac{2^{nbrpaths-i-1} \times nbrpaquets}{2^{nbrpaths} - 1} \right\rceil, \qquad (5)$$

where [] denotes the upper integer part. Thereby, on the first route, node *s* sends the:

$$\left\lceil \frac{2^{nbrpaths-1} \times nbrpaquets}{2^{nbrpaths} - 1} \right\rceil$$

first packets. On the next, it sends the following:

$$\left\lceil \frac{2^{nbrpaths-2} \times nbrpaquets}{2^{nbrpaths} - 1} \right\rceil$$

and so on, till all the packets are transmitted. This way almost all the routes will be used and the number of packets sent on each route *i* is approximately twice as much as the number of packets sent on the next route ($i + 1$).

4 The protocol

The protocol we propose is derived from DSR which is a distributed reactive protocol based on the source routing concept. In this section, we first give an overview on DSR; then we will present the main modifications that we have added to this protocol. In the third subsection, we will present our protocol in more detail and discuss the operation that we have added.

4.1 DSR (Dynamic Source Routing)

DSR (Johnson and Maltz, 1996) is a reactive protocol based on the source route approach. The principle of this approach is that the whole route is chosen by the source, and is put within the header of each data packet transmitted, in the IP options. Each node keeps in its cache, the source routes that it learns. When it needs to send a packet, it first checks its cache; if it finds a route to the corresponding destination then it uses it, otherwise it launches a route discovery by broadcasting a request (RREQ) packet through the network. When receiving the RREQ, a node seeks a route in its cache for the RREQ's destination; if it finds such a route, it sends a route reply (RREP) packet to the source, and if no appropriate route exists, then it adds its address to the request packet and continues the broadcasting. When a node detects a route failure, it sends a route error (RER) packet to the source that uses this link, then this one will either choose another route or apply again, the route discovery process if it has no other route to the destination in its cache. A detailed presentation of DSR is available in Johnson and Maltz (1996).

4.2 Main modifications

We mainly add to DSR, the following:

- *Implementation of the power control technique*. The required powers are computed on each link during the route discovery procedure, as in Doshi and Brown (2002a). On each link, the transmitter of a RREQ puts the transmission power in the RREQ's header, then the receiver retrieves this value that it uses along with the reception power which it gets from its radio, and the appropriate environment's propagation model, to compute the required power.
- *Consider energy state*. For this purpose, we assume that each node is able to determine its consumed energy rate. This can be achieved by implementing a bottom level mechanism, allowing nodes to determine their battery voltage. When getting its battery voltage, the node can deduce its energy consumed rate from the battery discharge curve (Brown et al., 2001; Gold, 1997). Moreover, each node should have a view on the energy states of the other nodes. To ensure this, we suggest that it should piggyback the energy state information into control packets, especially into reply packets and E_state packets that we present hereafter.
- *Adding E_state packet.* Each node has a global view of other nodes' energy states. To get a better view (especially about neighbours), we propose that each node broadcasts in its neighbourhood, a special packet called E_state, which carries the current energy state of the sender. This small packet is sent each time the energy consumed rate changes with a certain rate. For instance, if we fix this rate at 25%, each node sends 3 packets; the first upon consuming 25%, the second upon consuming 50%, and the last one upon consuming 75% of the total battery capacity. This adds an overhead of complexity $3 \times n$ packets transmission during the whole network life, where n is the nodes number. Generally speaking, if the rate is $1/x$ then the cost is $(x - 1) \times n$ packets transmission. We note that it

may be interesting to increase this rate when batteries capacities decrease, because the energy state information becomes more and more important when the batteries lose their capacities. But increasing this rate, however, requires more overhead and imposes a tradeoff between the information freshness and the overhead.

- Routes are stored in each node's cache according to our metric Opt_c presented previously (formula 4).
- *Packets format modification*. In order to implement our technique and metrics, we add some modifications to packets format. Mainly, each field of the record part should contain, in addition to the IP address of a node i, the current energy state of the node i (E_i), and the power to use on the link relaying it to the next hope.
- *Adding the possibility to discover more routes*. When a node receives a request that it has already received, contrary to DSR, it may continue relaying it if this request has followed a different route from the one followed by the first received request. This depends on the ALLROUTES flag that we add to the request. Although this technique allows discovering much more routes than the standard DSR, it results in more overhead, especially when connectivity increases. We will test this technique's performance in the next section.
- *Route maintenance modification*. Mobility may make a power used on a link insufficient, without causing the link failure. This occurs when nodes forming the link move away from each other, while keeping the distance between them less than their power range. Therefore, a mechanism allowing discovery of new suitable power is required. We suggest proceeding as follows: When a packet fails to pass through a link and the MAC layer protocol of the sender turns it back to the routing protocol, the last protocol tries to resend the packet using the full power, by paging it back to what we call *a directed request*, a special request packet which differs from the ordinary request in the fact that it is sent to a unique node (the upstream node of the appropriate link), instead of being propagated. Moreover, the sender updates its variables in order to temporally avoid sending packets via this link. If the upstream receives the packet, it replies with a maintenance reply packet. Otherwise, after a given timeout, the sender's MAC protocol turns back the directed request to its routing protocol, then this latter detects the link failure.
- To distribute the routing load over a maximum number of nodes, and to benefit from the available routes in the cache, we propose to use our data dispersal technique presented previously.

4.3 Protocol description and verification

In the following, we describe the algorithm executed by each node, and we analyse and verify the operations added to DSR, relying on the correctness and the stability of DSR. Our protocol can be summarised by the following events:

Having packets to send

This local event is triggered when a node has one or more data packets to send to any destination. If the node has any routes to the destination then it can send the packets, using our data dispersal technique previously presented. However, if the node has no route to the appropriate destination, then it launches a route request (if no request to the destination has recently been launched).

When the node finds no route to the destination, it reacts exactly as in DSR. The only difference appears when the node has more than one route to the destination. In this case it uses almost all these routes, contrary to DSR where just the shortest one is used. This is equivalent to the data dispersal technique (Rabin, 1989) already proposed for security and load balancing as in distributed systems, but our protocol, however, does not add any redundancy, and considers routes' optimality when selecting the number of packets to be transmitted on each route. The stability of this operation depends on the stability of each transmission on each route, which is identical to DSR. Note that dispersing data packets causes no problem, since data in packets based networks (to which ad hoc networks belong) usually follow different routes and the upper protocols (TCP/UDP) reorder them.

Receiving a route request packet (RREQ)

When a node receives a request packet, and if it is a mater of a directed request, then it computes the new power required and sends a reply. Otherwise, if the request is an ordinary one and the node is not its final destination, then it computes the required power on the previous link, adds this computed value along with its current battery state to the RREQ, and continues its broadcast as in DSR. If the node is the RREQ's destination, it sends back a RREP exactly like in DSR; it just adds to each field of the source route discovered, the power-aware information computed and collected along the route discovery process. The directed request does not exist in the standard DSR; we have added it in order to maintain unbroken links on which the power used becomes insufficient because of nodes' mobility. This operation does not affect the protocol correctness; when sending a directed.

RREQ, if the destination is still in the sender's power range, then it replies and provides the new required power, thereby the forwarding continues like in DSR. On the other hand, if the link is broken then the sender will not receive any reply, and after a timeout it will react for the link broken detection as in DSR. However, the impact of these operations is the possible rise of the latency that will be investigated later in the simulation. The power-aware information addition into RREQs increases the overhead, which causes more energy consumption, but provides important energy gain later when transmitting data packets. The simulation results which we present in the next section will check whether the gain is more important than the cost.

Receiving a route reply packet (RREP)

When a node receives a maintenance reply packet, it updates its cache, resends awaiting packets for this maintenance, and sends a route error back to each node which is using the maintained link. Otherwise, if the reply type is not maintenance, the receiver reacts as in DSR.

Resending awaiting packets is analogous to salvaging packets after a link break in the standard DSR (Johnson and Maltz, 1996). The only difference is that when salvaging after a link break, the packets will be transmitted on a new discovered route, but in our case they will be transmitted on the same route using a new up to date power on the current hop. Sending a route error packet to nodes currently using the maintained link has the purpose of informing these nodes about the actual required power on the link in question, which would be useful at these node for routes selection. The only impact of this procedure is the increasing overhead.

Receiving a data packet to forward

This event is triggered when a node receives a data packet for which it is not its final destination but an intermediate router. If the next hop of the routing header was recently broken then the node sends an error packet to the source. If the link is waiting for maintenance then the node puts the packet in a waiting queue. Otherwise, if the link is supposed to be reliable, the router continues forwarding the packet like in DSR, but using the power specified in the source header instead of using the full power. The only considerable difference from DSR is when the link is waiting for maintenance; in this case packets are delayed. All the possible subsequent operations in this case have been already discussed.

Receiving a link error message

It is a local event triggered by the MAC protocol when the node fails to relay a packet to its next hop. If the packet returned is a directed request, then the node detects a link failure. Otherwise, it supposes that the power used over the link is not sufficient anymore, so it tries to maintain this link by sending a directed request.

Receiving a route error packet (RER)

Error packets can be launched either because of a link failure or because the power required on a link changes, according to the cost field value, a field we add to the RER packet. If the cost field value is ∞, then it is a matter of a link failure and the node reacts as in DSR. Otherwise, it just updates its cache by taking into consideration the new power value and it normally continues forwarding the RER packet.

Significant change of energy state

This local event takes place each time the energy state of the node changes *significantly* with a certain threshold, as described before. When the event is triggered, the node broadcasts to its neighbours, an E_state packet, in order to inform them about its new energy state. This adds overhead but, note that this additional overhead is limited, since the E_state packet is of moderate size, and it is not forwarded beyond the transmitter's neighbourhood.

As illustrated before, the threshold energy change configured for E_state transmission affects the overhead, as well as the power-aware information freshness essential for selecting routes. In our simulation, we fixed this value at 10%. That is, each node sends an E_state packet each time it consumes 10% of its battery power.

Receiving E_state

When a node receives an E_state packet, it updates the parameters used to compute routes, and its cache as well.

5 Performance evaluation

To evaluate our protocol's performance, we drive a simulation study using GloMoSim (Zeng et al., 1998), to which we add many extensions, such as the implementation of our protocol. From our protocol we derive four versions by varying the parameter ALL_ROUTES and the function α; then we compare them to the standard DSR available in GloMoSim which we consider as a benchmark. As it will be illustrated later in the next section, our protocol shows important improvements in power consumption while keeping acceptable delays. We note our protocol DSRPA as DSR Power-Aware, and its versions as follows:

DSRPA0: ALL_ROUTES parameter is disabled, and $\alpha = F1$

DSRPA1: ALL_ROUTES parameter is enabled, and $\alpha = F1$

DSRPA2: ALL_ROUTES parameter is disabled, and $\alpha = F2$

DSRPA3: ALL_ROUTES parameter is enabled, and $\alpha = F2$.

5.1 Simulation environment

We simulate a network of ten nodes moving around in a 330×250 m^2 area during 900 seconds. Each node has a power range of 150 m, and moves according to the random way point model (Badache et al., 2003; Broch et al., 1998). Table 1 summarises the simulation set up.

Table 1 Simulation set up

No. of nodes	10
Simulation time	900 s
Terrain	330 m × 250 m
Power range	150 m
Mobility pattern	Random way point
Propagation pattern	Free space
MAC protocol	IEEE802.11
Application protocol	CBR

Table 1 Simulation set up (continued)

Received request list entry time out	30 s
Replaying from cache	No
Minimum time separating two requests	10 ms
Time separating two transmissions of a request	500 ms
Estate sending fraction	10%
Failed links list entry time out	1 s

5.2 *Metrics of comparison*

Our purpose is to minimise the energy consumption and maximise the battery life time of all nodes, so that the network partition will be avoided as long as possible. Intuitively, this means maximising the average battery life time and minimising the battery life time difference between nodes. A protocol is considered more power-efficient than another if it causes less energy consumption, more average battery life time, and less battery life time difference. By introducing new routing metrics, our protocol adds more communication and computation overhead, thus more energy consumption. The measurements of the energy consumption allow us to check whether the energy gain provided from our protocol is higher than the energy consumption caused by its overhead. Moreover, this overhead may increase the latency. To investigate this impact, the end to end delay is also included as a metric of comparison in our simulation. Our simulation study includes the following metrics of comparison:

Consumed energy

The energy computation in GloMoSim is based on NCR Wavelan radio model (Lucent technologies, 2000). Node i's consumed energy ($Power\_consumed_i$) is computed using the following formula:

$$PC_i = \sum_{Rx} TD \times (RRR - RSR) + \sum_{Tx} TD \times (RTR \times (p/p_{\max}) - RSR) + RSR \times (R_{off} - R_{on}) \quad (6)$$

where

TD: packet size/bandwidth + ST.
ST: 192 micro second.
RTR(RadioTransmissionRate) = 3/second,
RRR(RadioReceptionRate) = 1.48/second,
RSR(RadioSleepRate) = 0.18/second.
R_{on}: the radio turning on time (simulation start time).
R_{off}: the radio turning off time (simulation end time).
P: the transmission power.
$p_{\max}$: the maximum full power, it is 72,321 mWhr in our simulations, which is the power allowing to cover the 150 m power range.
Tx: The set of transmitted packets.
Rx: The set of received packets.

In other words, PC_i is the sum of: the power consumed for transmitting all packets, the power consumed for receiving all packets, and the power consumed during the time when the radio is in the idle mode. Since the last part of formula 7 ($RSR \times (R_{off} - R_{on})$) is equivalent to the power consumed in the idle mode for the whole simulation time, the equivalent consumed energy in the idle mode during receptions and transmissions has been subtracted in the first two sums $\left(\sum_{Rx} TD \times RSR \text{ and } \sum_{Tx} TD \times RSR\right)$. Thus, the formula (6) can be rewritten as:

$$PC_i = \sum_{Rx} TD \times RRR + \sum_{Tx} TD \times RTR \times p/p\max + RSR \times (R_{off} - R_{on}) - \sum_{Tx} TD \times RSR - \sum_{Rx} TD \times RSR. \quad (7)$$

The first two sums represent, together, the energy consumed for communication; the former represents the total energy consumed in reception, whereas the latter is the total energy consumed in transmission. The remainder of the formula is the energy consumed by the radio when it is in the idle mode, which is unaffected by the routing protocol but can be reduced if some radio management technique is employed by the MAC protocol (Rakhmatov et al., 2002). In our case, however, the power consumed in the idle mode is always the same regardless of the protocols; hence we do not consider it.

Our first metric of comparison is the consumed energy for communication, it is the two first sums of the previous formula averaged to make an average metric on nodes number (m), and is given by:

$$E_{\text{com}} = \sum_{I=1}^{m} \frac{\sum_{Rx} TD \times RRR + \sum_{Tx} TD \times RTR \times p/p_{\max}}{m}. \quad (8)$$

This metric can be divided into two parts: the reception average energy, and the transmission average energy; they are respectively given by:

$$E_{RX} = \sum_{I=1}^{m} \sum_{Rx} \frac{TD \times RRR}{m} \quad (9)$$

$$E_{TX} = \sum_{I=1}^{m} \sum_{Tx} \frac{TD \times RTR \times p / p_{\max}}{m}. \quad (10)$$

Average battery life time

It is the average battery discharge time, that is, the average time when nodes' batteries are discharged; it is given by:

$$BL_{avg} = \sum_{I=1}^{m} BL_i / m. \quad (11)$$

BL_i: is the discharge time of the battery of node i, we also call it the battery life time of node i.

Battery life time difference i

It is the difference between the maximum battery life time and the minimum one, formally speaking:

$$BL_{\text{dif}} = \max_{i=1\ldots m}(BL_i) - \min_{i=1\ldots m}(BL_i). \tag{12}$$

The performance regarding this metric will be achieved by minimising it. Thus, a protocol is more energy efficient than another if it increases the average battery life time and decreases this metric.

End to end delay

This well-known metric is the average time separating the data packets transmission from source nodes and their arriving at destinations. If we note this metric by *delay*, then:

$$\text{delay} = \sum_{i \in pr} \frac{\text{delay}_i}{\|pr\|}$$

such that: pr is the set of packets received by all the destination nodes, $\|pr\|$ is the number of the received packets and delay_i is the transfer delay of packet i, where: delay_i = packet i arrival time – packet i transmission time.

We will investigate the impact of our protocol's computation and communication overhead on this metric.

5.3 Simulation stages

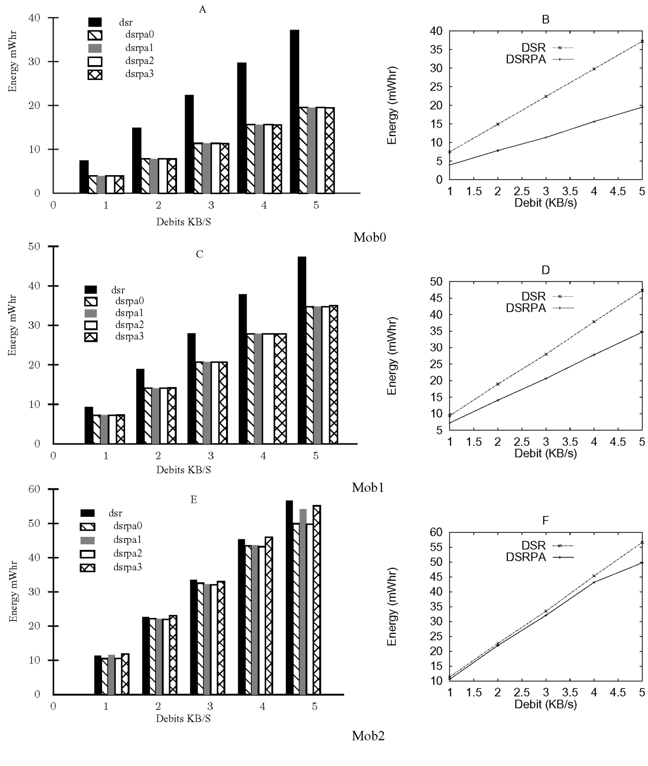
We evaluate our protocol's performance by comparing it to DSR in different network loads, batteries charge, and mobility situations. In order to investigate the mobility impact on our solution, we use along our simulation, three kinds of mobility: mob0 which is a stationary situation (no mobility), mob1 which is a medium mobility, the nodes' average speed is 0.5 m/s resulting in an average link change (Badache et al., 2003) of 8.00, and finally the high mobility mob2, resulted from an average speed 1 m/s which is important vis a vis the simulation scenarios, since it causes an average link change of 14.00. The network load is generated by two CBR sessions that last for the whole simulation time, one between nodes 0 and 9 and the other between 6 and 7. The packet size is maintained at 1 kb all along our simulation, and we vary the sender's CBR load (throughput) by changing the time separating two transmissions; the loads used are 1 kb/s, 2 kb/s, 3 kb/s, 4 kb/s and 5 kb/s. We assume that all the nodes have the same initial battery charge. During the performance evaluation vs. the load, it is fixed to 300 mWhr when measuring the energy and the end to end delay (this value avoids any discharge during the 10 minutes of simulation), and this parameter is fixed at 160 mWhr when measuring the other metrics to cause discharge during the simulation time. During the measurements vs. the battery capacity, the batteries charge is changed from 40 mWhr to 200 mWhr. However, these values are far from the real batteries' charges (Rakhmatov et al., 2002) which require much more simulation time and thus, very powerful equipment. These last measurements (vs. the battery charge) illustrate the impact of the battery charge on the protocol performance. All the metrics presented in the previous sections are measured in the different situations of mobility, network load, and battery capacity, resulting in 225 scenarios (executions).

6 Simulation results

6.1 Consumed energy

The histograms A, C, and E of Figure 2 show the DSR consumed energy and those of each DSRPA versions vs. the throughput in the different mobilities (mob0, mob1 and mob2). The other diagrams; B, D and F represent the energy consumption of DSR and that of the best version of our protocol; they clearly illustrate the difference between DSR and our new protocols. We remark from the diagram A that all the versions of DSRPA consume far less energy than DSR, and they have almost the same consumption. But for mobility mob2, the versions that use the ALLROUTES option (DSRPA1 and DSRPA3) consume slightly more energy than the others. However, all DSRPA's versions outperform DSR in all situations. From the diagrams B, D and F, we can see that the difference between DSR and DSRPA decreases with mobility, but it increases with the throughput even for high mobility (mob2). DSRPA ensures an important gain in the energy consumption, especially when the throughput (the network load) increases. In the following, we will investigate this gain by analysing the two parts of this metric (transmission and reception energy).

Figure 2 Consumed communication energy

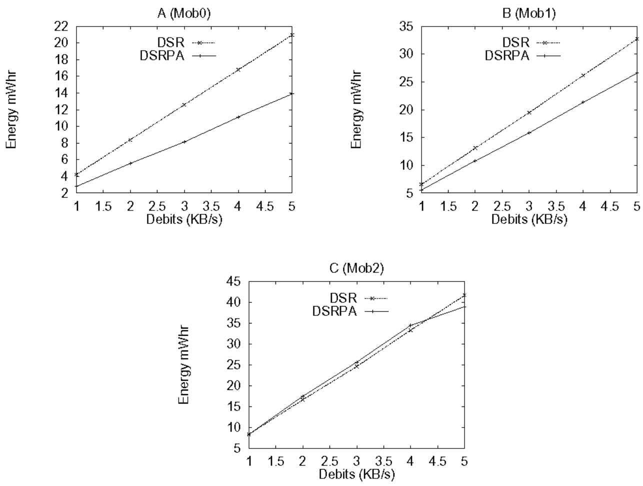


Transmission energy. Diagrams representing the transmission energy have almost the same forms as the previous ones, thus they are omitted due to space limitation. However, we remarked some differences:

- the difference between DSR and DSRPA is more important especially for mob2
- the versions that use ALL_ROUTES option do not consume more energy for transmissions than the others.

Reception energy. We remark that the plots in diagrams A and B of Figure 3 have almost the same forms as the corresponding ones of Figure 2. Nevertheless, for the mobility mob2, and contrary to the previous plots, we remark that before the debit 4 kb/s, DSR consumes slightly less energy than DSRPA, and DSRPA becomes a little bit more efficient for the highest load (5 kb/s). On the one hand, unlike DSRPA, DSR uses the promiscuous mode; thereby nodes receive all packets that they overhear when they are in the ideal mode, which explains the DSRPA's gain. On the other hand, mobility causes more overhead for maintaining routes in DSRPA that uses adaptable powers. As we have said, nodes' mobility may renders powers used on links insufficient without breaking these links down, which causes extra maintenance only to DSRPA but not to DSR. This overhead for maintenance explains the difference in favour of DSR in the first part of the diagram C. We point out that because the extra maintenance overhead has been largely compensated by the gain provided from the power control technique and from the metric used to select routes this DSR's outperforming was not observed either in the transmission energy part or in the total communication energy. The histograms are not represented, since they have quite the same forms as those of Figure 2. The minor extra consumption of versions that use the ALLROUTES option is due to the wide propagation of requests.

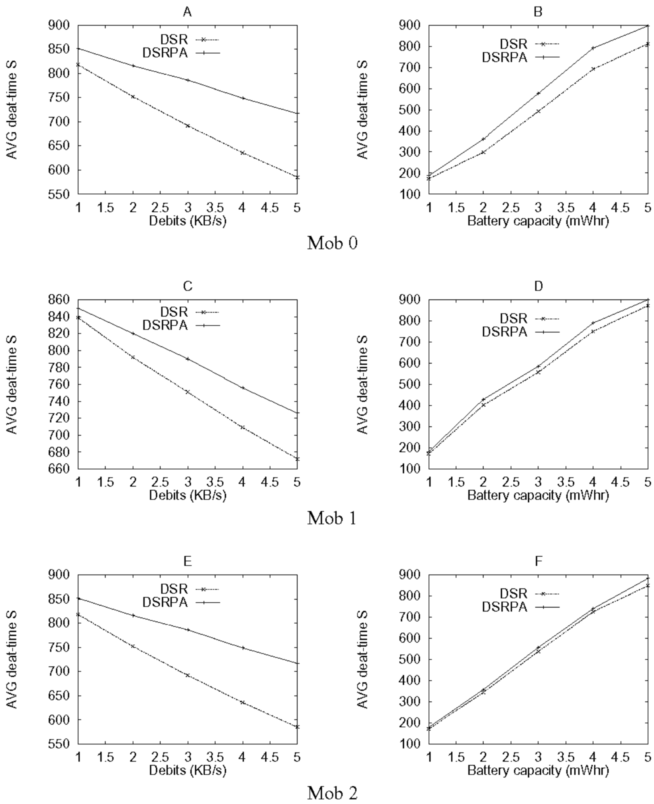
Figure 3 Consumed reception energy



6.2 Average battery life time

On diagrams A, C, and E of Figure 4, we present the average battery life time vs. the throughput. The same metric is represented vs. the battery initial charge on the other diagrams. The histograms representing the DSRPA's versions are omitted since all these versions have all but the same values. We can clearly see that DSRPA is always more efficient than DSR, and that the difference between DSR and DSRPA increases with the network load and the battery charge, but it slightly decreases with the mobility. The DSRPA's gain is due to the energy consumption gain analysed previously.

Figure 4 Average life time



6.3 Battery life time difference

Diagrams B, D and F of Figure 5 illustrate how the difference between DSR and DSRPA with respect to the battery life time difference is important. It increases with the load and also with the mobility, contrary to the other metrics. This performance is mainly due to our metrics that are battery state aware, and also to our new technique that allows us to disperse the network charge over the available routes. From diagrams A, C and E, we can see that DSRPA3 has often the best performance (the minimum difference); the other versions are close to each other. Hence, the function F2 and the ALL_ROUTES option minimise the battery life time difference while keeping the average battery life time good enough. But note that the choice of the ALL_ROUTES option must depend on the network connectivity. When the connectivity is high, this option causes a lot of requests broadcasting, resulting in more energy consumption.

Figure 5 Battery life time difference vs. throughput

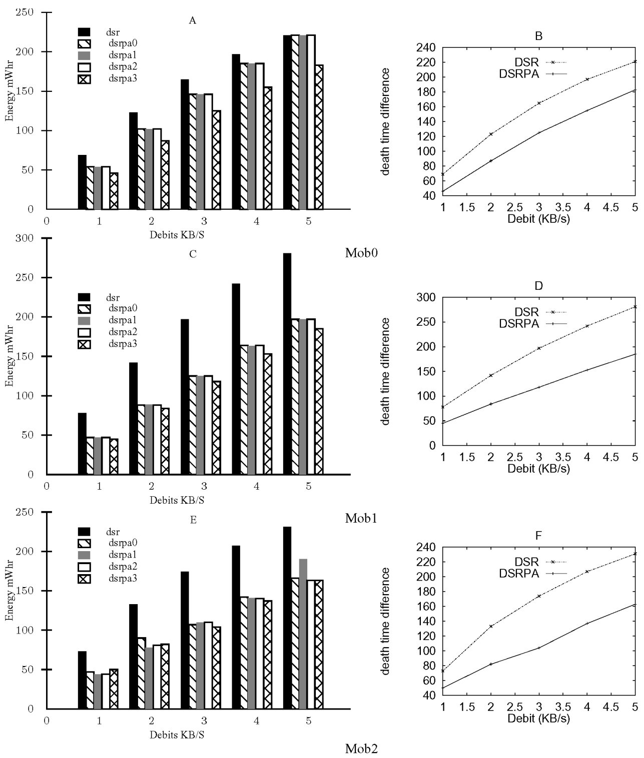


Figure 6 shows the difference between DSRPA and DSR on the battery life time difference vs. the initial battery charge. The plot A is monotone, increasing upon the charge 80 mWhr, where B is increasing up to the charge 120, then it becomes stable. The last one shows an important increasing from charges 80 to 160, and it is nearly stable out of this interval. Generally speaking, the difference between DSRPA and DSR with respect to the battery life time difference increases with the initial battery charge, and it is very important for the high charges (160 mWhr an 200 mWhr) compared with the low ones (40 mWhr and 80 mWhr).

6.4 *End to end delay*

Figure 7 shows the end to end delay of data packets vs. the network load in different mobilities. We remarked that all the versions of our protocol have always almost the same values, thus we omit histograms representing these versions.

As illustrated in plots of diagrams A, B, and C of Figure 7, representing respectively, mobilities mob0, mob1 and mob2, the delay of our protocol is too close to that of DSR in all situations. Nevertheless, the DSR's delay is a bit lower than that of DSRPA. This is mainly due to the computation and communication overhead added by our protocol to exchange power-aware information, and to the link maintenance procedure due to nodes' mobility which affects only DSRPA due to the power control use. However, the difference is minor and unaffected by the mobility or the load increase. We also omit representing the delay vs. the initial battery capacity, since we remarked that the delay was uninfluenced by this parameter.

Figure 6 Battery life time difference vs. Battery's charge

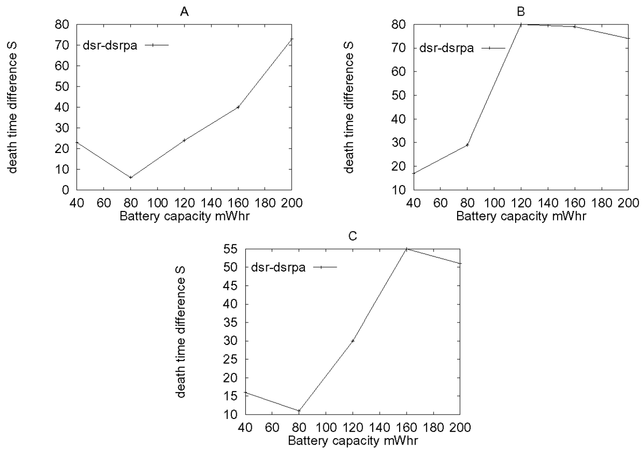
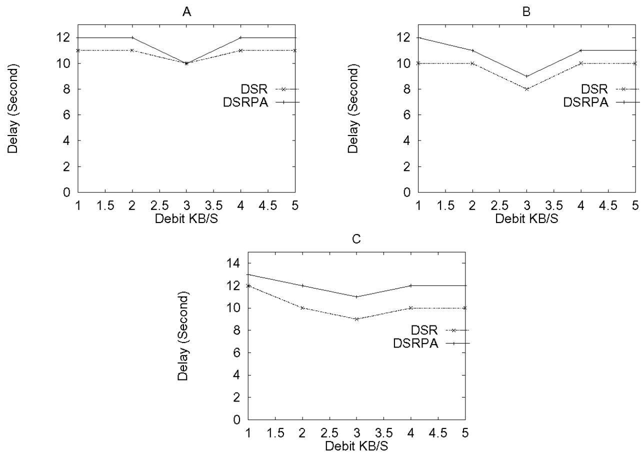


Figure 7 End to end delay vs. load



7 Conclusions and future work

One of the promising techniques proposed in literature to reduce the energy consumption in ad hoc networks is the power transmission control. When using this technique, the obvious metric for route selection is to minimise the total power required, but this choice may cause an overuse of a nodes subset and resulting in a network partition. To avoid such a problem, batteries' states should be considered when selecting routes.

In this paper, we have defined new power-aware metrics and a new technique that can be implemented with any routing protocol. The proposed metrics represent both powers required on links and batteries' states of nodes. The technique we have defined is analogous to a policy used by operating system schedulers; it aims to disperse the network information load over as many nodes as possible, while considering their optimality. Basing on these metrics, techniques, and DSR, we have presented a new power-aware routing protocol, whose purpose is to allow nodes to stay alive together as long as possible.

Our protocol's performance has been evaluated by simulation in different situations of mobility and network load. We have defined two kinds of comparison metrics; the first represents the energy consumption, whereas the second represents the battery life time. The last kind includes two metrics: the average battery life time and the battery life time difference. This last one is not related to the consumed energy (the fist metric) at all. We have also measured the end to end delay, and investigated the impact of the overhead introduced by our protocol's operations on this important QoS (Quality of Service) metric. An efficient protocol must both increase the average battery life time (by decreasing the consumed energy), and decrease the battery life time difference, without affecting the end to end delay.

The results of our simulation show important improvements compared with the DSR benchmark, especially for high load situations, while keeping the end to end delay too close to the one of DSR. The second function (F2) for the parameter α shows slightly better efficiency as well as the ALL_ROUTES parameter. But as for the last one, it must be treated carefully in high connected networks.

The proposed protocol relies on the cooperation and the wellbehaving of nodes. However, in some applications, namely the self-organised ad hoc networks applications where nodes do not belong to a single authority and do not pursue a common goal, a node is unwilling to spend its energy on routing packets for other nodes, and may tend to be *selfish.* As a perspective, we plan to study this emergent problem.

Acknowledgements

Many thanks are due to the anonymous reviewers for their comments that have considerably helped us improving the first version of the paper.

References

Badache, N., Djenouri, D. and Dehab, A. (2003) 'Mobility impact on mobile ad hoc networks', *ACS/IEEE Conference Proceeding*, Tunis, Tunisia.

Broch, J., Maltz, D.A., Hu, Y-C. and Jetcheva, J. (1998) 'A performance comparison of multi-hop wireless ad hoc network routing protocols', *The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, TX, USA, pp.85–97.

Brown, T., Doshi, S. and Zhang, Q. (2001) 'Optimal power aware routing in a wireless ad hoc network', *IEEE LANMAN 2001 Workshop Proceedings*, pp.102–105.

Doshi, S. and Brown, T. (2002a) 'Minimum energy routing schemes for a wireless ad hoc network', *The 21st IEEE Annual Joint Conference on Computer Communications and Networking (INFOCOM'02)*, New York, USA.

Doshi, S. and Brown, T. (2002b) 'Design considerations for an on-demand minimum energy routing protocol for a wireless ad hoc network', *International Conference on Communications (ICC'02)*, New York, USA.

Gold, S. (1997) 'A PSPICE macromodel for lithium-ion batteries', *The 12th Annual Battery Conference on Applications and Advances*, California State University, Long Beach, CA, USA.

Johnson, D.B. and Maltz, D.A. (1996) 'Dynamic source routing in ad hoc wireless networks', in Imielinskiand, T. and Korth, H. (Eds.): *Mobile Computing*, Kluwer Academic Publishers, Vol. 353, Chapter 5, pp.153–181.

Lucent technologies (2000) 'WAVELAN product specification', www.wavelan.com

Rabin, M.O. (1989) 'Efficient dispersal of information for security, load balancing, and fault tolerance', *Journal of ACM*, Vol. 36, No. 2, pp.335–348.

Rakhmatov, D., Vrudhula, S. and Wallach, D.A. (2002) 'Battery lifetime prediction for energy-aware computing', *International Symposium on Low Power Electrics and Design*, Monterey, California, USA, pp.154–159.

Singh, S., Woo, M. and Raghavendra, C.S. (1998) 'Power-aware routing in mobile ad hoc networks', *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, Texas, USA, pp.187–190.

Zeng, X., Bagrodia, R. and Gerla, M. (1998) 'GloMoSim: a library for the parallel simulation of large-scale wireless networks', *The 12th Workshop on Parallel and distributed Simulation, PADS'98*, Banff, Alberta, Canada, pp.154–161.