Power-Aware Localized Routing in Wireless Networks

Ivan Stojmenovic and Xu Lin

Abstract—Recently, a cost aware metric for wireless networks based on remaining battery power at nodes was proposed for shortest-cost routing algorithms, assuming constant transmission power. Power-aware metrics, where transmission power depends on distance between nodes and corresponding shortest power algorithms were also recently proposed. We define a new power-cost metric based on the combination of both node's lifetime and distance-based power metrics. We investigate some properties of power adjusted transmissions and show that, if additional nodes can be placed at desired locations between two nodes at distance d, the transmission power can be made linear in d as opposed to d^α dependence for $\alpha \geq 2$. This provides basis for power, cost, and power-cost localized routing algorithms where nodes make routing decisions solely on the basis of location of their neighbors and destination. The power-aware routing algorithm attempts to minimize the total power needed to route a message between a source and a destination. The cost-aware routing algorithm is aimed at extending the battery's worst-case lifetime at each node. The combined power-cost localized routing algorithm attempts to minimize the total power needed and to avoid nodes with a short battery's remaining lifetime. We prove that the proposed localized (where each node makes routing decisions based solely on the location of itself, its neighbors, and destination) power, cost, and power-cost efficient routing algorithms are loop-free and show their efficiency by experiments.

Index Terms—Routing, wireless networks, distributed algorithms, power management.

◆

1 INTRODUCTION

In this paper, we consider the routing task, in which a message is to be sent from a source node to a destination node (in a sensor or ad hoc wireless network). Due to propagation path loss, the transmission radii are limited. Thus, routes between two hosts in the network may consist of hops through other hosts in the network. The nodes in the network may be static (e.g., thrown from an aircraft to a remote terrain or a toxic environment), static most of the time (e.g., books, projectors, furniture), or moving (vehicles, people, small robotic devices). Wireless networks of sensors are likely to be widely deployed in the near future because they greatly extend our ability to monitor and control the physical environment from remote locations and improve our accuracy of information obtained via collaboration among sensor nodes and online information processing at those nodes. Networking these sensors (empowering them with the ability to coordinate among themselves on a larger sensing task) will revolutionize information gathering and processing in many situations. Sensor networks have been recently studied in [9], [12], [13], [16]. A similar wireless network that received significant attention in recent years is ad hoc network [15], [20]. Mobile ad hoc networks consist of wireless hosts that communicate with each other in the absence of a fixed infrastructure. Some examples of the possible uses of ad hoc networking include soldiers on the battlefield, emergency disaster relief personnel, and networks of laptops.

Macker and Corson [20] listed qualitative and quantitative independent metrics for judging the performance of routing protocols. Desirable qualitative properties [20] include: distributed operation, loop-freedom, demand-based operation, and "sleep" period operation, while hop count and delivery rates are among quantitative metrics. We shall further elaborate on these properties and metrics in order to address the issue of routing in wireless networks while trying to minimize the energy consumption and/or reduce the demands on nodes that have significantly depleted batteries. This is an important problem since battery power at each node is limited. Our final goal is to design routing protocols with the following properties:

1. *Minimize energy required per routing task*. Hop count was traditionally used to measure energy requirement of a routing task, thus using constant metric per hop. However, if nodes can adjust their transmission power (knowing the location of their neighbors), then the constant metric can be replaced by a power metric that depends on distance between nodes [8], [22], [12]. The distance between neighboring nodes can be estimated on the basis of incoming signal strengths (if some control messages are sent using fixed power). Relative coordinates of neighboring nodes can be obtained by exchanging such information between neighbors [4]. Alternatively, the location of nodes may be available directly by communicating with a satellite, using GPS (Global Positioning System), if nodes are equipped with a small low power GPS receiver. We will use location information in making routing decisions as well to minimize energy per routing task.

- *I. Stojmenovic is with SITE at the University of Ottawa, Ontario, K1N 6N5, Canada. E-mail: ivan@site.uottowa.ca.*
- *X.Lin is with Cognos Inc., 3755 Riverside Drive, P.O. Box 9707, St. T, Ottawa, ON, K1G 4K9, Canada. E-mail: xu.lin@cognos.com.*

Manuscript received 7 Feb. 2000; revised 2 Dec. 2000; accepted 26 Apr. 2001. For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 111419.

1045-9219/01/$10.00

2. *Loop-freedom*. The proposed routing protocols should be inherently loop-free to avoid timeout or memorizing past traffic as cumbersome exit strategies.
3. *Maximize the number of routing tasks that a network can perform*. Some nodes participate in routing packets for many source destination pairs and the increased energy consumption may result in their failure. Thus, the pure power consumption metric may be misguided in the long term [26]. A longer path that passes through nodes that has plenty of energy may be a better solution [26]. Alternatively, some nodes in the sensor or ad hoc network may be temporarily inactive and the power consumption metric may be applied on active nodes.
4. *Minimize communication overhead.* Due to limited battery power, the communication overhead must be minimized if the number of routing tasks is to be maximized. Proactive methods that maintain routing tables with up-to-date routing information or global network information at each node are certainly unsatisfactory solutions, especially when node mobility is high with respect to data traffic. For instance, shortest-path-based solutions are too sensitive to small changes in local topology and activity status (the later even does not involve node movement).
5. *Avoid memorizing past traffic or route*. Solutions that require nodes to memorize route or past traffic are sensitive to node queue size, changes in node activity, and node mobility while routing is ongoing (e.g., monitoring environment). Flexibility in selecting routes is thus preferred.
6. *Localized algorithms*. Localized algorithms [9] are distributed algorithms that resemble greedy algorithms, where simple local behavior achieves a desired global objective. In a localized routing algorithm, each node makes a decision to which neighbor to forward the message based solely on the location of itself, its neighboring nodes, and destination. While neighboring nodes may update every other location whenever an edge is broken or created, the accuracy of the destination location is a serious problem. In some cases, such as monitoring environment by sensor networks, the destination is a fixed node known to all nodes (i.e., monitoring center). Our proposed algorithms are directly applicable in such environments. All nonlocalized routing algorithms proposed in literature are variations of shortest weighted path algorithm (e.g., [5], [19], [22], [26]).
7. *Single-path routing algorithms.* The task of finding and maintaining routes in mobile networks is nontrivial since host mobility causes frequent unpredictable topological changes. Most previously proposed position-based routing algorithms (e.g., [1], [18]) for wireless ad hoc networks were based on forwarding the actual message along multiple paths toward an area where destination is hopefully located, thus achieving robustness. However, we have shown, in our previous work, that single-path strategies may be even more robust (for instance, they can guarantee delivery [3]) and with less communication overhead. The significant communication overhead can be avoided if a variant of source-initiated on-demand routing strategy [2], [24] is applied. In the strategy, the source node issues several search "tickets" (each ticket is a "short" message containing sender's *id* and location, destination's *id* and best known location and time when that location was reported, and constant amount of additional information) that will look for the exact position of destination node. When the first ticket arrives at the destination node D, D will report back to the source with a brief message containing its exact location and possibly create a route for the source. The source node then sends a full data message ("long" message) toward the exact location of destination. The efficiency of destination search depends on the corresponding location update scheme. A quorum-based location update scheme is being developed in [32]. Other schemes may be used with various trade-offs between the success and flooding rates (including an occasional flooding). If the routing problem is divided as described, the mobility issue is algorithmically separated from the routing issue, which allows us to consider (in this paper) only the case of static networks with known destination in our algorithms and experiments. The choice is justified whenever the destination does not move significantly between its detection and message delivery and information about neighboring nodes is regularly maintained. Yet another routing method may forward a message toward an imprecise destination location, hoping that closer nodes will locate a destination more accurately.
8. *Maximize delivery rate.* Our localized algorithms achieve very high delivery rates for dense networks while further improvements are needed for sparse networks. We have designed power, cost, and power-cost routing algorithms that guarantee delivery, which is an extension to be reported elsewhere [28].

The final important goal of a routing algorithm is to handle node mobility with proper *location update schemes.* This issue seems to be the most complex of all discussed here, as argued in an upcoming report [27]. In a source-initiated routing strategy, the source broadcasts short destination search messages. The destination responds with the short routing message (knowing the location of sender) indicating its position. Finally, the source routes full message to the known location of destination. If this strategy is adopted, routing and mobility issues can be separated since the message speed is significantly greater than the mobility rate. Each node is only required to maintain the location of its neighbors as the only communication overhead.

Ad hoc and sensor networks are best modeled by *minpower* graphs constructed in the following way. Each node A has its transmission range $t(A)$. Two nodes A and B

in the network are neighbors (and, thus, joined by an edge) if the Euclidean distance between their coordinates in the network is less than the minimum between their transmission radii (i.e., $d(A,B) < min\{t(A), t(B)\}$). If all transmission ranges are equal, the corresponding graph is known as the *unit graph.* The minpower and unit graphs are valid models when there are no obstacles in the signal path. Ad hoc and sensor networks with obstacles can be modeled by subgraphs of minpower or unit graphs. The properties of power metrics, the proposed algorithms, and their loop free properties in this paper are valid for arbitrary graphs. We have used, however, only unit graphs in our experiments.

A number of protocols for achieving efficient routing has been recently proposed. They differ in the approach used for searching a new route and/or modifying a known route when hosts move. The surveys of these protocols that do not use geographic location in the routing decisions are given in [2], [15], [23], [24]. The power awareness in these protocols is limited to the amount of control messages sent and the degree of message flooding. While the computational power of the devices used in the network is rapidly increasing, the lifetime of batteries is not expected to improve much in the future. We see a clear need for improvement in power consumption in existing MAC protocols and routing algorithms [26].

In the next section, we shall review known power-aware metrics and routing algorithms. In Section 3, existing routing protocols that use geographic location or consider power in their route decisions are reviewed. Section 4 discusses the effect of power awareness on the routing decisions in GPS based algorithms. Section 5 proposes three distributed (localized) algorithms aimed at extending node and/or network life. In Section 6, we prove that these algorithms are loop-free. Their performance evaluation is given in Sections 7 and 8.

2 EXISTING POWER-AWARE METRICS AND ROUTING ALGORITHMS

In most of routing protocols, the paths are computed based on minimizing hop count or delay. When transmission power of nodes is adjustable, hop count may be replaced by a power consumption metric. Some nodes participate in routing packets for many source-destination pairs and the increased energy consumption may result in their failure. A longer path that passes through nodes that have plenty of energy may be a better solution [26].

The algorithm [26] proposed to use a function $f(A)$ to denote node A's reluctance to forward packets and to choose a path that minimizes the sum of $f(A)$ for nodes on the path. This routing protocol [26] addresses the issue of energy critical nodes. As a particular choice for *f,* [26] proposes $f(A) = 1/g(A)$ where $g(A)$ denotes the remaining lifetime ($g(A)$ is normalized to be in the interval [0,1]). Thus, reluctance grows significantly when lifetime approaches 0. The other metric used in [26] is aimed at minimizing the total energy consumed per packet. However, [26] merely observes that the routes selected when using this metric will be identical to routes selected by shortest hop count routing since the energy consumed in transmitting (and receiving) one packet over one hop is considered *constant.* For each of the two proposed power consumption metrics (cost and hop count), [26] assigns weights to nodes or edges and then refers to nonlocalized Dijkstra's algorithm for computing shortest weighted path between source and destination. We also observed that the validation of power-aware metrics in [26] was done on random graphs where each pair of nodes is joined by an edge with a fixed probability *p.*

Rodoplu and Meng [22] proposed a general model in which the power consumption between two nodes at distance d is $u(d) = d^{\alpha} + c$ for some constants α and c and describe several properties of power transmission that are used to find neighbors for which direct transmission is the best choice in terms of power consumption. In their experiments, they adopted the model with $u(d) = d^4 + 2 * 10^8$, which will be referred to as *RM-model.* They discuss that large-scale variations (modeled by lognormal shadowing model) can be incorporated into the path loss formula and that small-scale variations (modeled by a Rayleigh distribution) may be handled by diversity techniques and combiners at the physical layer. Rodoplu and Meng [22] described a power-aware routing algorithm which runs in two phases. In the first phase, each node searches for its neighbors and selects these neighbors for which direct transmission requires less power than if an intermediate node is used to retransmit the message. This defines so-called *enclosure* graph. In the second phase, each possible destination runs a distributed loop-free variant of nonlocalized Bellman-Ford shortest path algorithm and computes the shortest path for each possible source. The same algorithm is run from each possible destination. The algorithm is thus proactive, resulting in significant overhead for low data traffic volumes. We observe that, since the energy required to transmit from node A to node B is the same as energy needed to transmit from node B to node A, the same algorithm may be applied from each possible source and used to discover the best possible route to each destination node. Alternatively, it may be used to find the location of destination and the best route to it. Such an on-demand variant is a competitive routing protocol, but it requires path memorization and may not be energy efficient since a single transmission at larger radius may reach more nodes at once.

Ettus [8] showed that minimum consumed energy routing reduces latency and power consumption for wireless networks utilizing CDMA, compared to minimum transmitted energy algorithm (shortest path algorithm was used in experiments). Heizelman et al. [12] used signal attenuation to design an energy efficient routing protocol for wireless microsensor networks, where destination is fixed and known to all nodes. They propose to utilize a 2-level hierarchy of forwarding nodes, where sensors form clusters and elect a random clusterhead. The clusterhead forwards transmissions from each sensor within its own cluster. This scheme is shown to save energy under some conditions. However, clustering requires significant communication overhead, therouting algorithm is not localized, and the destination is not necessarily fixed.

Nevertheless, their simple radio model and metric is adopted in our paper as follows:

In the simple radio model [12], radio dissipates $E_{elec} = 50\ \mathrm{nJ/bit}$ to run the transceiver circuitry. Both sender and receiver node consume E_{elec} to transmit one bit between them. Assuming d^2 energy loss, where d is the distance between nodes, transmit amplifier at the sender node consumes $E_{amp}d^2$ further, where $E_{amp} = 100\ \mathrm{pJ/bit/m^2}$. Thus, to transmit one bit message at distance d, the radio expends $E_{elec} + E_{amp}d^2$ and, to receive the message, the radio expends E_{elec}. In order to normalize the constants, divide both expressions by E_{amp}, so that radio expands $T = E + d^2$ for transmission and $P = E$ for reception, where $E = E_{elec}/E_{amp} = 500\mathrm{m}^2$. Note that $T/P = 1 + d^2/E$ and $T + P = 2E + d^2$. Therefore, in this model, referred to as *HCB-model*, the power needed for transmission and reception at distance d is $u(d) = 2E + d^2$.

Preliminary versions of this paper were published as technical reports [31] in 1998 and presented at a conference [30]. Chang and Tassiulas [6], [7] independently proposed (in 1999) combining power and cost into a single metric. In [6], they experimented with metric $c_{ij} = (\underline{E}_i - e_{ij}\lambda)^{-1}$, where $e_{ij} = u(d) = d^4 + c$ is energy for transmission on link ij with length d, λ and c are small constants, and $\underline{E}_i$ is remaining battery power at node i. In [7], they proposed a general metric $c_{ij} = e^a_{ij}\underline{E}_i^{-b}E_i^c$, where E_i is initial energy at node i, and a, b, and c are constants. They consider routing tasks with fixed source-destination pairs, one-to-one [6], and one-to-many [7] cases. The power needed for reception is not considered. Distributed non-localized Bellman-Ford shortest weighted path algorithm is used. Their experiments indicate $(a, b, c) = (1, 50, 50)$ as values that are close to an optimal one. Network lifetime is maximized when traffic is balanced among the nodes in proportion to their energy reserves instead of routing to minimize the absolute consumed power [6], [7]. A similar power-cost metric was proposed also in [11] (in 1999), followed by a multipath route-redirect algorithm, where messages are redirected through any intermediate node that saves power or reduces cost. However, multipath transmission, in effect, increases the power and cost, contrary to the design goals.

3 EXISTING GPS-BASED ROUTING METHODS

Most existing routing algorithms do not consider the power consumption in their routing decisions. They are reviewed here in order to experimentally compare their power savings performance with newly proposed algorithms. All described routing algorithms are localized, demand-based, and adapt well to "sleep" period operation. Several GPS-based methods were proposed in 1984-1986 by using the notion of progress. Define progress as the distance between the transmitting node and receiving node projected onto a line drawn from transmitter toward the final destination. A neighbor is in forward direction if the progress is positive; otherwise, it is said to be in backward direction. In the random progress method [21], packets destined toward D are routed with equal probability towards one neighboring node that has positive progress. In the *NFP* method [14], a packet is sent to the nearest neighboring node with forward progress. Takagi and Kleinrock [33] proposed *MFR* (most forward within radius) routing algorithm, where the packet is sent to the neighbor with the greatest progress. The method is modified in [14] by proposing to adjust the transmission power to the distance between the two nodes. Finn [10] proposed a variant of random progress method, called Cartesian routing, which "allows choosing any successor node which makes progress toward the packet's destination" [10]. The best choice depends on the complete topological knowledge. Finn [10] adopted the greedy principle in his simulation: Choose the successor node that makes the best progress toward the destination. When no node is closer to the destination than current node C, the algorithm performs a sophisticated procedure that does not guaranty delivery.

Recently, three articles [1], [18], [17] independently reported variations of fully distributed routing protocols based on direction of destination. In the *compass routing* (or *DIR*) method proposed by Kranakis et al. [17], the source or intermediate node A uses the location information for the destination D to calculate its direction. The location of one hop neighbors of A is used to determine for which of them, say C, is the AC direction closest to the direction of AD. The message m is forwarded to C. The process repeats until the destination is hopefully reached. A counterexample showing that undetected loops can be created in directional-based methods is given in [29]. The method is therefore not loop-free.

GEDIR routing algorithm [29] is a variant of the greedy routing algorithm [10] with a "delayed" failure criterion. *GEDIR*, *MFR*, and compass routing algorithms fail to deliver message if the best choice for a node currently holding message is to return it to the previous node [29]. Such criterion reduced the failure rate and provided a fair comparison in our experiments. *GEDIR* and *MFR* algorithms are inherently loop-free [29]. The proofs are based on the observation that distances (dot products, respectively) of nodes toward destination are decreasing. A routing algorithm that guarantees delivery by finding a simple path between source and destination is described in [3].

The 2-hop variants of three basic routing algorithms were proposed in [29]. The delivery rate of *GEDIR*, compass routing (or *DIR*), or *MFR* algorithms can be improved if each node is aware of its 2-hop neighbors (neighbors of its neighbors). The node A currently holding the message may then choose the node closest to the destination among all 1-hop and 2-hop neighbors and forward the message to its neighbor that is connected to the choice. In case of ties (that is, more than one neighbor connected to the closest 2-hop neighbor), choose the one that is closest to the destination.

This review did not include various flooding-based or multiple paths routing algorithms or methods for sending control messages to update positions [1], [18], [32]. Our primary interest in this paper is to examine power consumption under the assumption that nodes have accurate information about the location of their neighbors and destination node (e.g., static networks, source-initiated on-demand routing, or networks with superb location update scheme).

4 SOME PROPERTIES OF POWER ADJUSTED TRANSMISSIONS

In this section, we shall study the optimality of power adjusted transmissions using a simple and general radio model. We shall further generalize the model of [22] (by adding a linear factor) and assume that the power needed for the transmission and reception of a signal is $u(d) = ad^\alpha + c$. Constant factor *c* in this expression for total energy consumption may also include the energy consumed in computer processing and encoding/decoding at each station. Next, the leading coefficient *a* can be adjusted to the physical environment, unit of length considered, unit size of a signal (a bit, byte, or frame, for example), etc. In the *RM-model*, $\alpha = 4$, $a = 1$, $c = 2^*10^8$ while in the *HCB-model* $\alpha = 2$, $a = 1$, $c = 2\ E = 1{,}000$. These two models were used in our experiments.

Suppose that the sender *S* is able to transmit the packet directly to the destination *D*. Let us examine whether energy can be saved by sending the packet to an intermediate node *A* between the nodes and forwarding the packet from *A* to *D*. Let $|SD| = d$, $|SA| = x$, and $|AD| = d - x$. Here, we assume that additional nodes can be placed at arbitrary positions and the results shall, therefore, be treated as lower bounds on energy consumption for transmission.

Lemma 1. *If* $d > (c/(a(1-2^{1-\alpha})))^{1/\alpha}$, *then there exists intermediate node A between source S and destination D so that the retransmission will save the energy. The greatest power saving is obtained when A is in the middle of SD.*

Proof. The power needed to send the message directly from *S* to *D* is $u(d) = ad^\alpha + c$ while the power needed to send it via *A* is

$$(ax^\alpha + c) + (a(d-x)^\alpha + c) = a(x^\alpha + (d-x)^\alpha) + 2c.$$

$a(x^\alpha + (d-x)^\alpha) + 2c < ad^\alpha + c$ is satisfied for

$$g(x) = a(x^\alpha + (d-x)^\alpha - d^\alpha) + c < 0.$$

The minimum for $g(x)$ is obtained for $g'(x) = 0$, i.e., $a(\alpha x^{\alpha-1} - \alpha(d-x)^{\alpha-1}) = 0$. Thus, $x^{\alpha-1} = (d-x)^{\alpha-1}$, $x = d - x$, or $x = d/2$. The minimum is < 0 if $g(d/2) < 0$, i.e., $a((d/2)^\alpha + (d/2)^\alpha - d^\alpha) + c < 0$ or

$$ad^\alpha(2^{1-\alpha} - 1) + c < 0,$$

which is satisfied for $c < ad^\alpha(1 - 2^{1-\alpha})$, or

$$d^\alpha > c/(a(1-2^{1-\alpha}))$$

and the lemma follows. Note that this inequality has a solution in *d* if and only if $\alpha > 1$. □

Lemma 2. *If* $d > (c/(a(1-2^{1-\alpha})))^{1/\alpha}$, *then the greatest power savings are obtained when the interval SD is divided into* $n > 1$ *equal subintervals, where n is the nearest integer to* $d(a(\alpha-1)/c)^{1/\alpha}$. *The minimal power is then*

$$dc(a(\alpha-1)/c)^{1/\alpha} + da(a(\alpha-1)/c)^{(1-\alpha)/\alpha}.$$

Proof. Let *SD* be divided into intervals of lengths $x_1, x_2, \ldots, x_n$ such that $d = x_1 + \ldots + x_n$. The energy needed for transmissions using these intervals is

$$(ax_1^\alpha + bx_1 + c) + \ldots + (ax_n^\alpha + bx_n + c) = \\ nc + a(x_1^\alpha + \ldots + x_n^\alpha).$$

For fixed $x_i + x_j$, the expression $x_i^\alpha + x_j^\alpha$ is minimal when $x_i = x_j$ (see Lemma 1). Therefore, the energy is minimal when $x_1 = x_2 = \ldots = x_n = d/n$ and is equal to

$$f(n) = cn + an(d/n)^\alpha = nc + an^{1-\alpha}d^\alpha.$$

This expression has the minimum when $f'(n) = 0$ or

$$c + a(1-\alpha)n^{-\alpha}d^\alpha = 0,$$

i.e., $c = a(\alpha-1)n^{-\alpha}d^\alpha$, $n^\alpha = a(\alpha-1)d^\alpha/c$,

$$n = d(a(\alpha-1)/c)^{1/\alpha}$$

(*n* is rounded to the nearest integer). □

For *HCB*- and *RM*-models, optimal values of *n* are $0.031d$ and $0.011d$, respectively. Assuming that we can set additional nodes in arbitrary positions between the source and destination, the following theorem gives power optimal packet transmissions.

Theorem 1. *Let d be the distance between the source and the destination. The power needed for direct transmission is* $u(d) = ad^\alpha + c$ *which is optimal if* $d \leq (c/(a(1-2^{1-\alpha})))^{1/\alpha}$. *Otherwise (that is, when* $d > (c/(a(1-2^{1-\alpha})))^{1/\alpha}$*),* $n - 1$ *equally spaced nodes can be selected for retransmissions, where* $n = d(a(\alpha-1)/c)^{1/\alpha}$ *(rounded to the nearest integer), producing minimal power consumption of about*

$$v(d) = dc(a(\alpha-1)/c)^{1/\alpha} + da(a(-1)/c)^{(1-\alpha)/\alpha}.$$

Corollary 1. *Let* $\alpha = 2$. *The power needed for direct transmission is* $u(d) = ad^2 + c$ *which is optimal if* $d \leq (2c/a)^{1/2}$. *Otherwise (that is, when* $d > (2c/a)^{1/2}$*),* $n - 1$ *equally spaced nodes can be selected for retransmissions, where* $n = d(a/c)^{1/2}$ *(rounded to the nearest integer), producing minimal power consumption of about* $v(d) = 2d(ac)^{1/2}$.

Theorem 1 announces the possibility of converting polynomial function in *d* (with exponent *α*) for power consumption (in case of direct transmission from sender to destination) to linear function in *d* by retransmitting the packet via some intermediate nodes that might be available.

5 POWER SAVING ROUTING ALGORITHMS

If nodes have information about the position and activity of all other nodes, then the optimal power saving algorithm that will minimize the total energy per packet can be obtained by applying Dijkstra's single source shortest weighted path algorithm, where each edge has weight $u(d) = ad^\alpha + c$ and where *d* is the length of the edge. This will be referred to as the *SP-power algorithm.*

We shall now describe a corresponding localized routing algorithm. The source (or an intermediate node) *B* should select one of its neighbors to forward the packet toward the destination with the goal of reducing the total power needed for the packet transmission. Let A be a neighbor of B and let $r = |AB|$, $d = |BD|$, and $s = |AD|$. The power needed for transmission from B to A is $u(r) = ar^\alpha + c$, while the power needed for the rest of the routing algorithm is not known. Assuming a uniformly distributed network, we shall make a fair assumption that the power consumption for the rest of routing algorithm is equal to the optimal one (see Theorem 1). That is, the power needed for transmitting a message from A to D is estimated to be

$$v(s) = sc(a(\alpha - 1)/c)^{1/\alpha} + sa(a(\alpha - 1)/c)^{(1-\alpha)/\alpha}.$$

For $\alpha = 2$, $v(s) = 2s(ac)^{1/2}$. This is, of course, an unrealistic assumption. However, it is fair to all nodes. A more realistic assumption might be to multiply the optimal power consumption by factor t, which is a constant that depends on the network. The proposed algorithms are looking for existing nodes in the network that are closest to the optimal desirable position, hoping that this will produce energy savings. The validity of the approach has been confirmed by our experiments.

The localized *power efficient routing algorithm* can be described as follows: Each node B (source or intermediate node) will select one of its neighbors A which will minimize

$$\begin{aligned} p(B, A) &= u(r) + v(s) \\ &= ar^\alpha + c + sc(a(\alpha - 1)/c)^{1/\alpha} + sa(a(\alpha - 1)/c)^{(1-\alpha)/\alpha}. \end{aligned}$$

For $\alpha = 2$, it becomes $u(r) + v(s) = ar^2 + c + 2s(ac)^{1/2}$. If destination D is a neighbor of B, then compare the expression with the corresponding one, $u(d) = ad^\alpha + c$, needed for direct transmission ($s = 0$ for D and D can be treated as any other neighbor). The algorithm proceeds until the destination is reached, if possible. A generalized power efficient routing algorithm may attempt to minimize $p(B, A) = u(r) + tv(s)$, where t is a network parameter.

In the basic (experimental) version of the algorithms, the transmission stops if the message is to be returned to a neighbor it came from (otherwise, a detectable loop is created). The power-efficient routing algorithm may be formalized as follows:

```
Power-routing(S,D);
  A:=S;
  Repeat
        B:=A;
        Let A be neighbor of B that minimizes
            $p(B, A) = u(r) + tv(s)$;
        Send message to A
  until A=D(* destination reached *) or A=B (* delivery failed *)
```

Let us now consider the second metric proposed in [26], measuring the nodes' lifetime. Recall that the cost of each node is equal to $f(A) = 1/g(A)$, where $g(A)$ denotes the remaining lifetime ($g(A)$ is normalized to be in the interval [0,1]). Reference [26] proposed a shortest weighted path algorithm based on this node cost. It is referred to as the *SP-cost* algorithm in experimental data in Table 2. The algorithm uses the cost to select the path, but the actual power is charged to nodes.

The localized version of this algorithm, assuming constant power for each transmission, can be designed as follows: The cost $c(A)$ of a route from B to D via neighboring node A is the sum of the cost $f(A) = 1/g(A)$ of node A and the estimated cost of route from A to D. The cost $f(A)$ of each neighbor A of node B currently holding the packet is known to B. What is the cost of other nodes on the remaining path? We assume that this cost is proportional to the number of hops between A and D. The number of hops, in turn, is proportional to the distance $s = |AD|$ and inversely proportional to radius R. Thus, the cost is ts/R, where factor t is to be investigated separately. Its best choice might even be determined by experiments. We have considered the following choices for factor t:

1. t is a constant number, which may depend on network conditions.
2. $t = f(A)$ (that is, assuming that remaining nodes have equal cost as A itself).
3. $t = f'(A)$, where $f'(A)$ is the average value of $f(X)$ for A and all neighbors X of A.
4. $t = 1/g'(A)$, where $g'(A)$ is the average value of $g(X)$ for A and all neighbors X of A.

Note that $t = t(A)$ depends on A. The cost $c(A)$ of a route from S to D via neighboring node A is estimated to be $c(A) = f(A) + ts/R$ for the appropriate choice of t. We also suggest to investigate the product of two contributing elements instead of their sum, that is, the cost definition $c(A) = f(A)ts/R$.

The localized *cost efficient routing algorithm* can be described as follows: If the destination is one of node B's neighbors of currently holding the packet, then the packet will be delivered to D. Otherwise, B will select one of its neighbors A which will minimize $c(A)$. The algorithm proceeds until the destination is reached, if possible, or until a node selects the neighbor the message came from as its best option to forward the message. The algorithm can be coded as follows:

```
Cost-routing(S,D);
  A:=S;
  Repeat
        B:=A;
        Let A be neighbor of B that minimizes c(A);
        If D is neighbor of B
              then send to D else send to A
  until D is reached or A=B;
```

The versions of this cost routing algorithm that use choices 2) and 3) for t ($t = f(A)$ and $t = f'(A)$, respectively), will be referred to as *cost-ii* and *cost-iii* algorithms in our experiments.

We may incorporate both power and cost considerations into a single routing algorithm. A new power-cost metrics is first introduced here. What is the power-cost of sending a message from node B to neighboring node A? We propose two different ways to combine power and cost metrics into a single power-cost metric, based on the product and sum of two metrics, respectively. If the product is used, then the power-cost of sending message from B to a neighbor A is equal to $power - cost(B, A) = f(A)u(r)$ (where $|AB| = r$).

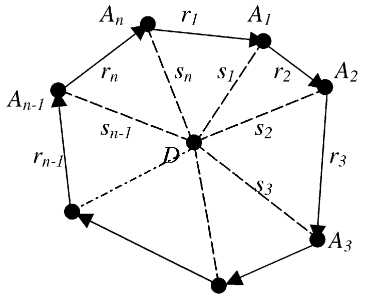

Fig. 1. Power efficient routing algorithm is loop-free.

The sum, on the other hand, leads to a new metric $power - cost(B, A) = \alpha u(r) + \beta f(A)$ for suitably selected values of α and β. For example, sender node S may fix $\alpha = f'(S)$ and $\beta = u(r')$, where r' is the average length of all edges going out of S. The values α and β are (in this version) determined by S and used, without change, by other node Bs on the same route. The corresponding shortest path algorithms can find the optimal power-cost by applying single source shortest weighted path Dijkstra's algorithm (the node cost is transferred to the edge leading to the node). The algorithm will be referred to as the $SP - Power^{*}Cost$ and $SP - Power + Cost$ algorithms, respectively, in Table 2.

The *power-cost efficient routing* algorithm may be described as follows: Let A be the neighbor of B (node currently holding the message) that minimizes

$$pc(B, A) = power - cost(B, A) + v(s)f'(A)$$

(where $s = 0$ for D, if D is a neighbor of B). The algorithm is named $power - cost0$ in Table 2 when

$$power - cost(B, A) = f(A)u(r)$$

and $power - cost1$ when

$$power - cost(B, A) = f'(S)u(r) + u(r')f(A).$$

The packet is delivered to A. Thus, the packet is not necessarily delivered to D when D is a neighbor of B. The algorithm proceeds until the destination is reached, if possible, and may be coded as follows:

```
Power-cost-routing(S,D);
    A:=S;
    Repeat
        B:=A;
        Let A be neighbor of B that minimizes
            $pc(B, A) = power - cost(B, A) + v(s)f'(A)$;
        Send message to A
    until A=D (* destination reached *)
        or A=B (* delivery failed *);
```

The algorithm may be modified in several ways. The second term may be multiplied by a factor that depends on network conditions. We also tested the version, called $power - cost2$, that minimizes $pc(B, A) = f(A)(u(r) + v(s))$ and an algorithm, called $power - costP$, that switches selection criteria from power-cost to power metric only whenever destination D is a neighbor of current node A.

6 LOOP-FREE PROPERTY

Theorem 2. *The localized power efficient routing algorithm is loop-free.*

Proof. Suppose that, on the contrary, there exists a loop in the algorithm. Let $A_1, A_2, \ldots A_n$ be the nodes in the loop so that A_1 sends the message to A_2, A_2 sends the message to $A_3, \ldots$, A_{n-1} sends the message to A_n, and A_n sends the message to A_1 (see Fig. 1). Let $s_1, s_2, \ldots, s_n$ be the distances of $A_1, A_2, \ldots A_n$ from D, respectively, and let

$$|A_nA_1| = r_1, |A_1A_2| = r_2, |A_2A_3| = r_3, \ldots, |A_{n-1}A_n| = r_n.$$

Let $u(r) = ar^{\alpha} + br + c$ and

$$v(s) = sc(a(\alpha - 1)/c)^{1/\alpha} + sa(a(\alpha - 1)/c)^{(1-\alpha)/\alpha}$$

(for $\alpha = 2$, $v(s) = 2s(ac)^{1/2}$). According to the choice of neighbors, it follows that $u(r_1) + v(s_1) < u(r_n) + v(s_{n-1})$ since the node A_n selects A_1, not A_{n-1}, to forward the message. Similarly $u(r_2) + v(s_2) < u(r_1) + v(s_n)$ since A_1 selects A_2 rather than A_n. Next,

$$u(r_3) + v(s_3)u(r_2) + v(s_1), \ldots, u(r_n) + v(s_n) < u(r_{n-1}) + v(s_{n-2}).$$

By adding the left and right sides, we obtain

$$\begin{aligned} &u(r_1) + u(r_2) + \ldots + u(r_n) + v(s_1) + v(s_2) \\ &\quad + \ldots + v(s_n) \\ &< u(r_n) + u(r_1) + \ldots + u(r_{n-1}) + v(s_{n-1}) + v(s_n) \\ &\quad + \ldots + v(s_{n-2}), \end{aligned}$$

which is a contradiction since both sides contain the same elements. Thus, the algorithm is loop-free. □

In order to provide for loop-free method, we assume that (for this and other mentioned methods below), in case of ties for the choice of neighbors, if one of the choices is the previous node, the algorithm will select that node (that is, it will stop or flood the message). Note that the above proof may be applied (by replacing "+" with "*") to an algorithm that will minimize $p(A) = u(r)tv(s)$.

Theorem 3. *Localized cost efficient algorithms are loop-free.*

Proof. Note that the cost $c(A)$ of sending a message from B to A is only the function of A (that is, $t = t(A)$) and is independent of B. In the previous proof, assume $u(r_i) = 0$ for all nodes and let $v(s_i) = c(A_i)$ for each i. The proof then becomes the same as in the previous theorem. The proof is valid for both formulas

$$c(A) = f(A) + ts/R \text{ and } c(A) = f(A)ts/R.$$

Note that the proof assumes that the cost of each node is not updated (that is, communicated to the neighbors) while the routing algorithm is in progress. It is possible to show that, on the other hand, if nodes inform their neighbors about new cost after every transmitted message, a loop (e.g., triangle) can be formed. □

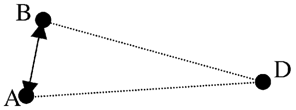

Fig. 2. *NFP* method fails.

Theorem 4. *Localized power-cost efficient algorithms are loop-free for the metrics* $power - cost(B, A) = \alpha u(r) + \beta f(A)$ *(where* α *and* β *are arbitrary constants) and*

$$pc(B, A) = power - cost(B, A) + v(s)t(A)$$

(where $t(A)$ *is determined by one of formulas 1-4).*

Proof. The proof is again by contradiction, similar to the proof of previous theorems. Suppose that there exists a loop $A1, A2, \dots An$ in the algorithm (see Fig. 4). Let $pc(A_n, A_1), pc(A_1, A_2), \dots pc(A_{n-1}, A_n)$ be the power-costs of sending message to nodes $A_1, A_2, \dots A_n$, respectively, from the previous node in the loop. According to the choice of neighbors in Fig. 1 it follows that $pc(A_n, A_1) < pc(A_n, A_{n-1})$ since the node A_n selects A_1, not A_{n-1}, to forward the message. Similarly,

$$\begin{aligned} pc(A_1, A_2) < pc(A_1, A_n), pc(A_2, A_3) \\ < pc(A_2, A_1), \dots, pc(A_{n-1}, A_n) \\ < pc(A_{n-1}, A_{n-2}). \end{aligned}$$

By adding the left and right sides we obtain

$$\begin{aligned} pc(A_n, A_1) + pc(A_1, A_2) + pc(A_2, A_3) + \dots + pc(A_{n-1}, A_n) \\ < pc(A_n, A_{n-1}) + pc(A_1, A_n) + \dots + pc(A_{n-1}, A_{n-2}). \end{aligned}$$

This inequality is equivalent to

$$\begin{aligned} &[\alpha u(r_n) + \beta f(A_1) + v(s_1)t(A_1)] + [u(r_1) + f(A_2) + v(s_2)t(A_2)] \\ &\quad + \dots + [\alpha u(r_{n-1}) + \beta f(A_n) + v(s_n)t(A_n)] \\ &\quad < [\alpha u(r_n) + \beta f(A_{n-1}) + v(s_{n-1})t(A_{n-1})] \\ &\quad + [\alpha u(r_1) + \beta f(A_n) + v(s_n)t(A_n)] \\ &\quad + \dots + [\alpha u(r_{n-1}) + \beta f(A_{n-2}) + v(s_{n-2})t(A_{n-2})], \end{aligned}$$

which is a contradiction since both sides contain the same elements. Thus, the algorithm is loop-free. Note that the proof also assumes that the cost of each node is not updated (that is, communicated to the neighbors) while the routing algorithm is in progress. Note that this proof does not work for the formula $power - cost(B, A) = f(A)u(r)$, which does not mean that the corresponding power-cost routing algorithm is not loop-free. □

7 PERFORMANCE EVALUATION OF POWER EFFICIENT ROUTING ALGORITHM

The experiments are carried using (static) random unit graphs. Each of n nodes is chosen by selecting its x and y coordinates at random in the interval $[0, m)$. In order to control the average node degree k (that is, the average number of neighbors), we sort all $n(n-1)/2$ (potential) edges in the network by their length in increasing order. The radius R that corresponds to the chosen value of k is equal to the length of $nk/2\text{th}$ edge in the sorted order. Generated graphs which were disconnected are ignored. We have fixed the number of nodes to $n = 100$ and average node degree k to 10. We have selected higher connectivity for our experiments in order to provide for better delivery rates and hop counts and concentrate our study on power conserving effects.

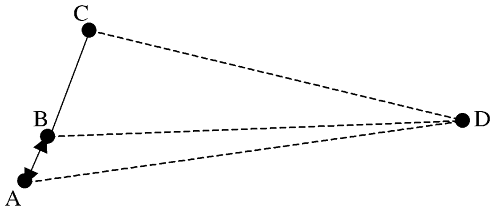

Fig. 3. $Power1$ frequently fails.

The choice of route for *DIR* (compass routing), *MFR*, and *GEDIR* methods in [29] and their mutual comparison did not depend on size *m* of the square containing all the points. However, in case of power consumption, the actual distances greatly impact the behavior of algorithms. More precisely, the path selection (and the energy for routing) in our power saving algorithm depends on the actual size of the square. We compared all methods for squares of sizes $m = 10$, 100, 200, 500, 1,000, 2,000, 5,000 for both *HCB-* and *RM-models*. The results are averages over 20 graphs with 100 routing pairs in each chosen at random.

In our comparisons, the power consumption (cost and power-cost, respectively) in all compared methods was measured by assigning the appropriate weights to each edge. Our comparison for the category of power (only) consumption involved the following GPS-based distributed algorithms: *NFP*, random progress, *MFR*, *DIR*, *GEDIR*, *NC*, the proposed localized power efficient routing algorithm (with $t = 1$), and the benchmark shortest (weighted) path algorithm (*SP*).

We have introduced a new routing method, called *NC* (nearest closer), in which node *A*, currently holding the message, forwards it to its nearest node among neighboring nodes, which are closer to destination *D* than *A*. This method is an alternative to the *NFP* method which was experimentally observed to have a very low success rate (under 15 percent in our case). The reason for the low success rate seems to be the existence of many acute triangles *ABD* (see Fig. 2) so that *A* and *B* are closest to each other and, therefore, selected by *NFP* method which then fails at such nodes.

The proposed power efficient method, which will be referred to as $power1$ method, was also experimentally shown to have very low success rate for large *m*. The power efficient algorithm is, therefore, modified to increase its success rate. Only neighbors that are closer to destination than the current node are considered and this variant will be called the *power* method. The success rates of *power* and $power1$ methods are almost the same for $m \leq 200$. While the success rate of *power* method remains at 95 percent level, the success rate for *power1* drops to 59 percent, 11 percent, 4 percent, and 2 percent only for remaining sizes of *m* (numbers refer to *HCB-model* and are similar for the other model). Consider a scenario in which $power1$ fails (see Fig. 3, where $|AD| < |BD| < |CD|$). Node *A* sends a message to the closest neighbor *B*. Since *A* is very close to

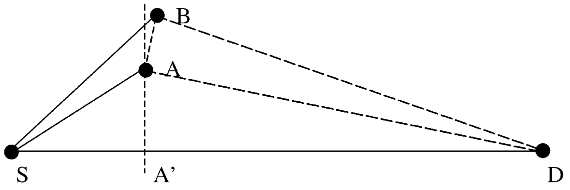

Fig. 4. GEDIR consumes less power than MFR.

B, but C is not, the power formula applied at B selects A to send the message back and a loop is created.

We included 2-hop *GEDIR, DIR,* and *MFR* methods in our experiments. The delivery rates for *1-GEDIR, 1-DIR,* and *1-MFR* methods in our experiments were about 97 percent, *1-NC* had 95 percent success, *2-GEDIR* (that is, *2-hop GEDIR*) and *2-MFR* had about 99 percent, *2-DIR* had about 91 percent, random method had about 98 percent, and power method had a 95 percent success rate (for both *HCB-* and *RM-models).* While all other methods choose the same path independently on m and power formula applied, power method does not and an almost constant and good delivery rate for it is a very encouraging result. The hop counts for nonpower-based methods were 3.8, 4.2, 3.9, 9.8, 3.8, 3.9, 4.1, and 6.4, respectively (in above order). Hop counts for power method were 3.8, 3.8, 3.8, 3.8, 6.3, 9.0, and 9.7 for *RM-model* and 3.8, 3.8, 4.0, 6.6, 8.3, 9.1, and 9.8 for *HCB-model,* in respective order of m. Clearly, with increased energy consumption per distance, the power method reacted by choosing closer neighbors, resulting in higher hop counts. For small and large values of m, hop counts of *GEDIR* and *NC* were limiting factors while, for $m = 500$, $m = 1,000$, the *RM-model* has significantly lower hop count than HCB-model.

Let us show the average case superiority of the *GEDIR* method over the *MFR* method and superiority of *DIR* routing over the random progress method. Let A and B be the nodes selected by the *GEDIR* and *MFR* methods, respectively, when the packet is to be forwarded from node S (see Fig. 4). Suppose that B is different from A (otherwise, the energy consumption at that step is the same). Therefore, $|AD| < |BD|$. Node B cannot be selected within triangle SAA', where A' is the projection of node A on direction *SD*, since B has more progress than A (here, we assume, for simplicity, that A and B are on the same side of line *SD*). However, the angle *SAB* is then obtuse and $|SB| > |SA|$. From $|SB| > |SA|$ and $|BD| > |AD|$, it follows that the packet requires more energy if forwarded to B instead of A.

Suppose now that A and B are selected neighbors in case of *DIR* and random progress routing algorithms (we shall use the same Fig. 4). Since the lengths $|SA|$ or $|SB|$ are not considered when selecting the neighbors, on the average, we may assume that $|SA| = |SB|$. However, the direction of A is closer to the direction of destination (that is, the angle *ASD* is smaller than the angle *BAD*) and, thus, A is closer to D than B.

Table 1 shows average power assumption (rounded to the nearest integers) per routing tasks that were successful by *all* methods, which occurs in about 85 percent of cases. It is calculated as the ratio of total power consumption (for each method) for these tasks over the total number of such deliveries. The quadratic *HCB-model* formula is used (the results for the *RM-model* were similar).

The power consumption for *GEDIR* algorithm is smaller than the one for *DIR* routing method for small values of square size m. The reason is that the smaller hop count is decisive when no retransmission is desirable. However, for larger m, *DIR* routing performs better since the greatest advance is not necessarily the best choice, and the closer direction, possibly with smaller advance, is advantageous. The *NC* method is inferior to *GEDIR* or *DIR* for smaller values of m because the greatest possible advance is the better choice for neighbor than the nearest node closer to destination. However, for larger values of m, *NC* outperforms both significantly since it simulates retransmissions in the best possible way. 2-hop methods failed to produce power savings over corresponding 1-hop methods and were eliminated in our further investigations.

As expected, the proposed distributed power efficient routing algorithm outperforms all known GPS-based algorithms for all ranges of m. For small m, it is a minor improvement over *GEDIR* or *DIR* algorithms. However, for large m, the difference becomes significant since the nearest rather than the furthest progress neighbors are preferred. For large m, the only competitor is *NC* algorithm.

The overhead (percentage of additional energy per routing task) of the power efficient algorithm, with respect to optimal *SP-power* one, is 1.2 percent, 2.3 percent,

TABLE 1
Power Consumption of Routing Algorithms

method/size	10	100	200	500	1000	2000	5000
SP-Power	3577	4356	6772	20256	62972	229455	1404710
SP	3578	4452	7170	25561	92438	358094	2236727
Power	3619	4457	6951	21331	69187	261832	1647964
GEDIR	3619	4460	7076	24823	89120	344792	2152891
DIR	3928	4681	7046	23033	81001	311743	1942952
MFR	3644	4523	7264	25845	93150	361021	2254566
NC	7604	8271	10523	25465	80136	297580	1833993
Random	5962	7099	10626	34382	121002	465574	2896988
2-GEDIR	3587	4452	7148	25399	91570	354980	2216528
2-DIR	3937	4764	7386	25109	89371	344644	2148913
2-MFR	3603	4478	7208	25738	92816	359491	2248876

TABLE 2
Number of Iterations Before One Node in Each Method Dies

method/trial count	10	100	200	500	1000	2000	5000
SP	289	713	1412	668	647	454	275
SP-Power	342	865	1710	983	1114	796	482
SP-Cost	674	1703	3540	1686	1590	1066	646
SP-Power*Cost	674	1697	3530	1776	1838	1230	728
SPPower+Cost	647	1668	3495	1725	1688	1124	682
Power	379	954	1843	1009	1162	789	469
Cost-iii	624	1630	3255	1594	1479	988	601
Cost-ii	637	1616	3304	1651	1494	991	602
PowerCostP	671	1616	3127	1522	1522	1053	600
Power*Cost	662	1609	3118	1513	1528	1056	617
Power+Cost	660	1611	3180	1664	1757	1179	712
PowerCost2	631	1537	3211	1676	1716	1152	686
1-GEDIR	373	941	1814	832	849	548	318
1-DIR	345	921	1741	831	902	603	355
1-MFR	375	909	1775	800	797	525	316
1-NC	204	551	1268	809	931	668	414
Random	201	481	889	546	512	312	202

2.6 percent, 5.3 percent, 9.9 percent, 14.1 percent, and 17.3 percent for the considered values of m, respectively. Therefore, the localized power efficient routing algorithm, when successful, closely matches the performance of the nonlocalized shortest-power path algorithm. We have experimented also with different values of parameter t, and concluded that $t = 1$ was the best value.

8 PERFORMANCE EVALUATION OF COST AND POWER-COST EFFICIENT ROUTING ALGORITHMS

The experiments that evaluate cost and power-cost routing algorithms are designed as follows: Random unit connected graphs are generated as in the previous section. An iteration is a routing task specified by the random choice of source and destination nodes. A power failure occurs if a node has insufficient remaining power to send a message according to a given method. Iterations are run until the first power failure at a node occurs (at which point the corresponding method "dies"). Each node is initially assigned an energy level at random in the interval [*minpow, maxpow*], where parameters depend on m. After sending a message from node A to node B, the energy that remained at A (B) is reduced by the power needed to transmit (receive) the message, respectively. The experiment is performed on 20 graphs for each method, for each of *HCB-* and *RM-model* formulas.

The success rates for unrestricted versions of cost and power-cost algorithms (where all neighbors were considered) was again low in our experiments. For example, the success rate of *cost-iii* method drops from 64 percent to 55 percent with increasing m while *power*cost* method drops from 77 percent to 14 percent (data for other variants are similar; *HCB-model* is again used while the other model had very similar data). Consequently, these methods were deemed not viable. The success rate for restricted versions (only closer neighbors considered) was in the range 92-95 percent for all cost and power-cost methods discussed here, both models and all sizes m. The number of iterations for *HCB-model* before each method dies is given in Table 2 (data refer to restricted versions). The *RM-method* gave similar results. The cost and power-cost methods are defined in Section 5.

The intervals *[minpow, maxpow]* were set as follows: [80K, 90K], [200K, 300K], [500K, 1M], [750K, 1.5M], [3M, 4M], [8M, 10M], [30M, 40M], for given respective sizes of m, where $K = 1,000$ and $M = 1,000,000$. Our experiments confirmed the expectations on producing power savings in the network and/or extending nodes lifetime. Both cost methods and all four power-cost methods gave very close trial numbers and, thus, it is not possible to choose the best method based on trial number alone. However, all proposed localized cost and power-cost methods performed equally well as the corresponding nonlocalized shortest path-cost and power-cost algorithms (the number of trials is sometimes even higher, due to occasional delivery failures which save power). It is also clear that cost and power-cost routing algorithms last longer than the power algorithm.

Table 3 shows the average remaining power at each node after the network dies for the most competitive methods. Cost methods have more remaining power only for the smallest size $m = 10$ when the power formula reduces to the constant function. For larger sizes of m, two better power-cost formulas leave about 15 percent more power at nodes than the cost method.

SP-cost, *cost-iii*, and *cost-ii* methods have hop counts approximately 4.0, 4.5, and 4.9 for *HCB-model* and all values of m. Four power-cost methods have similar hop counts, 5.8, 4.7, 5.0, 6.7, 8.4, 9.1, and 9.6, respectively, for sizes of m. Two SP-power-cost methods do not have similar hop counts. *SP-Power*Cost* method has hop counts 4.0, 4.1, 4.3, 6.3, 7.8, 8.3, and 8.7, while *SP-Power+Cost* method has hop counts between 4.0 and 4.6.

We have also experimented with moving nodes, using source initiated routing strategy. The communication overhead due to mobility depends on mobility rate with no significant impact on the success rate or power savings of routing task. The reason is that the location update between neighbors is only a very simple and power efficient strategy.

TABLE 3
Average Remaining Power Level at Each Node

method/power	10	100	200	500	1000	2000	5000
SP-Cost	44381	133245	395592	618640	1857188	4819903	19238265
SP-Power*Cost	44437	133591	396031	642748	2067025	5686092	23187052
SPPower+Cost	46338	136490	406887	646583	1972185	5252813	21081420
Cost-iii	43996	129608	410610	656349	2053190	5370162	21338314
Cost-ii	39831	120785	377549	619221	2022771	5335936	21233992
PowerCostP	30561	127819	421927	712958	2299590	6058424	24782129
Power*Cost	27434	126066	416889	712033	2286840	6030614	24419832
Power+Cost	27520	126201	409208	666907	2091211	5658144	22622947
PowerCost2	33563	131804	401174	652199	2078140	5684752	23136193

9 CONCLUSION

This paper described several localized routing algorithms that try to minimize the total energy per packet and/or lifetime of each node. The proposed routing algorithms are all demand-based and can be augmented with some of the proactive or reactive methods reported in literature to produce the actual protocol. These methods use control messages to update positions of all nodes to maintain efficiency of routing algorithms. However, these control messages also consume power and the best trade-off for moving nodes is to be established. Therefore, further research is needed to select the best protocols. Our primary interest in this paper was to examine power consumption in case of static networks and provide basis for further study. Our method was tested only on networks with high connectivity and their performance on lower degree networks remains to be investigated. Based on experience with basic methods like GEDIR [29], improvements in the power routing scheme to increase delivery rates or even to guaranty delivery [3], [28] are necessary before experiments with moving nodes are justified. Power efficient methods tend to select well positioned neighboring nodes in forwarding the message while the cost efficient method favors nodes with more remaining power. The node movement, in this respect, will certainly assist power aspect of the formula since the movement will cause the change in relative node positioning. This will further emphasize the advantage of power-cost over power only or cost only methods.

The formulas for power, cost, and power-cost methods may also need some improvements. Our experiments do not give an ultimate answer on even the selection of approach that would give the most prolonged life to each node in the network. We will investigate this question further in our future work [28] which will consider a number of metrics including a generalized one $f(A)^a u(r)^b$, which is similar to one proposed in [7].

ACKNOWLEDGMENTS

The authors are grateful to the referees for their careful reading of this manuscript which has improved its readability. This research is partially supported by National Science and Engineering Research Council of Canada (NSERC) and La Red de Desarollo e Investigación en Informática (REDII) grants.

REFERENCES

[1] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *Proc. Conf. Mobile Computing, MOBICOM,* pp. 76-84, 1998.

[2] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. Conf. Mobile Computing, MOBICOM,* pp. 85-97, 1998.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm.,* pp. 48-55, Aug. 1999.

[4] S. Capkun, M. Hamdi, and J.P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," *Proc. Hawaii Int'l Conf. System Sciences,* Jan. 2001.

[5] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks," *IEEE J. Selected Areas in Comm.,* vol. 17, no. 8, pp. 1488-1505, Aug. 1999.

[6] J.H. Chang and L. Tassiulas, "Routing for Maximum System Lifetime in Wireless Ad-Hoc Networks," *Proc. 37th Ann. Allerton Conf. Comm., Control, and Computing,* Sept. 1999.

[7] J.H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-Hoc Networks," *Proc. 19th Ann. Joint Conf. IEEE Computer and Comm. Societies, INFOCOM,* Mar. 2000.

[8] M. Ettus, "System Capacity, Latency, and Power Consumption in Multihop-Routed SS-CDMA Wireless Networks," *Proc. IEEE Radio and Wireless Conf.,* Aug. 1998.

[9] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. Conf. Mobile Computing, MOBICOM,* pp. 263-270, 1999.

[10] G.G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," Research Report ISU/RR-87-180, Inst. for Scientific Information, Mar. 1987.

[11] J. Gomez, A.T. Campbell, M. Naghshineh, and C. Bisdikian, "Power-Aware Routing in Wireless Packet Networks," *Proc. IEEE Int'l Workshop Mobile Multimedia Comm.,* Nov. 1999.

[12] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Routing Protocols for Wireless Microsensor Networks," *Proc. Hawaii Int'l Conf. System Sciences,* Jan. 2000.

[13] W.R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proc. Conf. Mobile Computing, MOBICOM,* pp. 174-185, 1999.

[14] T.C. Hou and V.O.K. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Trans. Comm.,* vol. 34, no. 1, pp. 38-44, 1986.

[15] Internet Engineering Task Force (IETF) internet draft, http://www.ietf.org/html.charters/manet-charter.html, 1998.

[16] J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Next Century Challenges: Mobile Networking for 'Smart Dust'," *Proc. Conf. Mobile Computing, MOBICOM,* pp. 271-278, 1999.

[17] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," *Proc. 11th Canadian Conf. Computational Geometry,* Aug. 1999.

[18] Y.B. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Proc. Conf. Mobile Computing, MOBICOM,* pp. 66-75, 1998.

[19] C.R. Lin and J.S. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE J. Selected Areas in Comm.,* vol. 17, no. 8, pp. 1426-1438, Aug. 1999.

[20] J.P. Macker and M.S. Corson, "Mobile Ad Hoc Networking and the IETF," *Mobile Computing and Comm. Rev.,* vol. 2, no. 1, pp. 9-14, 1998.

[21] R. Nelson and L. Kleinrock, "The Spatial Capacity of a Slotted ALOHA Multihop Packet Radio Network with Capture," *IEEE Trans. Comm.,* vol. 32, no. 6, pp. 684-694, 1984.

[22] V. Rodoplu and T.H. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE J. Selected Areas in Comm.,* vol. 17, no. 8, pp. 1333-1344, Aug. 1999.

[23] S. Ramanathan and M. Steenstrup, "A Survey of Routing Techniques for Mobile Communication Networks," *Mobile Networks and Applications,* vol. 1, no. 2, pp. 89-104, 1996.

[24] E.M. Royer and C.K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Comm.,* pp. 46-55, Apr. 1999.

[25] S. Singh and C.S. Raghavendra, "PAMAS—Power Aware Multi-Access Protocol with Signaling for Ad Hoc Networks," *ACM Computer Comm. Rev.,* pp. 5-26, July 1998.

[26] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proc. Conf. Mobile Computing, MOBICOM,* pp. 181-190, 1998.

[27] I. Stojmenovic, "Location Updates for Efficient Routing in Wireless Networks," *Wireless Networks and Mobile Computing,* 2001.

[28] I. Stojmenovic and S. Datta, *Power Aware Routing Algorithms with Guaranteed Delivery in Wireless Networks,* 2001.

[29] I. Stojmenovic and X. Lin, "GEDIR: Loop-Free Location Based Routing in Wireless Networks," *Int'l Assoc. Science and Technology for Development (IASTED) Conf. Parallel and Distributed Computing and Systems,* pp. 1025-1028, Nov. 1999.

[30] I. Stojmenovic and X. Lin, "Power-Aware Localized Routing in Wireless Networks," *IEEE Int'l Parallel and Distributed Processing Symp., IPDPS,* pp. 371-376, May 2000.

[31] I. Stojmenovic and X. Lin, "Power Aware Distributed Routing in Ad Hoc Wireless Networks," Technical Report TR-98-11, Computer Science, SITE, Univ. of Ottawa, Dec. 1998.

[32] I. Stojmenovic, "A Routing Strategy and Quorum Based Location Update Scheme for Ad Hoc Wireless Networks," Technical Report TR-99-09, Computer Science, SITE, University of Ottawa, Sept. 1999.

[33] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Trans. Comm.,* vol. 32, no. 3, pp. 246-257, 1984.

Ivan Stojmenovic received the BS and MS degrees in 1979 and 1983, respectively, from the University of Novi Sad and the PhD degree in mathematics in 1985 from the University of Zagreb, Croatia. In 1980, he joined the Institute of Mathematics, University of Novi Sad (Yugoslavia). In Fall 1988, he joined the faculty in the Computer Science Department at the University of Ottawa (Canada), where he is currently a full professor in SITE. Since June 2000, he has been a researcher at DISCA, IIMAS, Universidad Nacional Autonoma de Mexico. He has published three books and more than 150 different papers in journals and conferences. His research interests are wireless networks, parallel computing, multiple-valued logic, evolutionary computing, neural networks, combinatorial algorithms, computational geometry, and graph theory. He is currently the managing editor of *Multiple-Valued Logic*, an international journal, and an editor of the following journals: *Parallel Processing Letters*, *IASTED International Journal of Parallel and Distributed Systems, Discrete Mathematics and Theoretical Computer Science*, *Parallel Algorithms and Applications*, and *Tangenta*.

Xu Lin received the BCS degree (Dean's Honors List) from McGill University in 1997. In 1997, he started working for Canada Custom and Revenue Agency, during which he received the MCS degree from the University of Ottawa, 2000. He won a second prize at Quebec Mathematics Contest in 1990 and was ranked 23rd in 1994 Descartes Physics Contest. Currently, he is working as a full time software engineer in Cognos Inc.