Ad Hoc Networks 8 (2010) 15–29

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

# Power-aware semi-beaconless 3D georouting algorithms using adjustable transmission ranges for wireless ad hoc and sensor networks

A.E. Abdallah [a,*], T. Fevens [a], J. Opatrny [a], I. Stojmenovic [b]

[a] *Department of Computer Science and Software Engineering, Sir George Williams Campus, Concordia University, 1515 St. Catherine W, EV008.173, Montréal, Québec, Canada H3G 2W1*
[b] *SITE, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5*

ARTICLE INFO

*Article history:*
Received 27 September 2007
Received in revised form 17 February 2009
Accepted 5 March 2009
Available online 17 March 2009

*Keywords:*
Ad hoc networks
Localized routing in 3D
Power management

ABSTRACT

Due to the limited lifetime of the nodes in ad hoc network, energy efficiency needs to be an important design consideration in any routing algorithm for ad hoc and sensor networks. In most of the existing position-based routing algorithms the nodes use the maximum transmission power to discover neighbors, which may cause excessive power consumption. This paper presents several localized power-aware 3D position-based routing algorithms that increase the lifetime of a network by maximizing the average lifetime of its nodes. New algorithms are semi-beaconless, using for neighbor discovery an optimal transmission range (OR) for control packets, and, if needed, maximal transmission range (MR) during routing process, and using adjusted transmission radius for message transmission. PAGR algorithm selects neighbor closest to destination among those within OR if any exists providing progress, or otherwise among those within MR. If greedy progress is not possible, *PAGR*:*CFace*(1) variant resorts to face routing on projected network in coordinate plane until recovery is possible, at which point PAGR algorithm resumes. We evaluate our algorithms and compare their power savings with the current power-aware routing algorithms. The simulation results show a significant improvement in the overall network lifetime.

## 1. Introduction

A wireless sensor or (mobile) ad hoc network (MANET) consists of wireless hosts that communicate with each other in the absence of a fixed infrastructure. For each ordered pair of nodes $(u, v)$, there is an associated transmission power threshold, denoted by $P(u, v)$, which indicates the transmission power needed by $u$ so that its signal can be received by $v$. The transmission power threshold for a pair of nodes depends on a number of factors including the distance between the transceivers, interference, noise, environment, etc. [33]. A node in a network can communicate directly only with its neighbors (the nodes within its transmission range). To communicate with nodes outside its maximum transmission range, multi-hop routing is used utilizing intermediate communicating nodes.

A crucial problem in multi-hop routing is to find an efficient and correct route between a source and a destination; however for many networks, a more important problem is providing an energy efficient routing protocol because of the limited battery life of the wireless nodes. Transmission power management which selects the optimized power level of nodes is one of the primary means of increasing the lifetime of the nodes. The power consumption at each node in an ad hoc network can be divided into three phases, according to functionality [4]: the power consumed for transmitting the message, the power consumed during message reception, and the power consumed while the node is idle.

The design of power efficient position-based routing algorithms in wireless ad hoc networks is difficult for the

---

* Corresponding author. +1 514 562 2567.
*E-mail addresses:* ae_abdal@cse.concordia.ca (A.E. Abdallah), fevens@cse.concordia.ca (T. Fevens), opatrny@cse.concordia.ca (J. Opatrny), ivan@site.uottawa.ca (I. Stojmenovic).

1570-8705/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2009.03.001

following reasons. First, the nodes may not have a fixed location, thus the wireless links on the network may not be known in advance. Next, interference and noise affect the signal reception power which determines the number of neighbors. Finally, global information is not available to the nodes.

In previous work [34,39,41,42], two main metrics have been used to optimize power routing for a sequence of messages. The first metric, called the power metric, tries to minimize the energy consumed for each message. If the transmission range is fixed for all the nodes then the number of nodes in the route path is used as the energy required for the routing task. This metric can be optimized if the nodes can adjust their transmission range. Then the constant metric can be replaced by a power metric that depends on the distances between nodes. In formal terms [20], let $e_m$ be the energy required by the packet $m$ to traverse a sequence of nodes $n_1, n_2, \ldots, n_k$, where $n_1$ is the source and $n_k$ is the destination. If $p(n_i, n_{i+1})$ is the power needed to forward $m$ over one hop from $n_i$ to $n_{i+1}$, then the aim of the power metric is to minimize $e_m = \sum_{i=1}^{k-1} p(n_i, n_{i+1})$. A drawback of the power metric is that some nodes may be repeatedly chosen over many routes, which quickly leads to their failure. In many cases this may result in the loss of network connectivity.

The second metric, called network survivability [36,11], or cost (reluctance) metric [39], tries to maximize the lifetime of the nodes. Given alternative routing paths, select the one that will contribute to the longest network operation time. One way of optimizing this metric is by choosing the nodes with plenty of energy as relaying nodes. We will focus on this metric for our routing algorithms.

Mauve et al. [30] classified the routing algorithms in MANETs as being of two basic types: topology-based [37,35,34,6,9] and position-based [30,5,19]. Topology-based routing algorithms use the information about the links in the network to perform packet forwarding. It can be classified as *proactive* or *reactive*. In proactive protocol, i.e. DSDV [31] or OLSR [13], a routing table is used at each node which has to be updated when any node moves. The main drawback of these approaches is the unacceptable overhead when data traffic is lower than the mobility rate. In reactive approach, i.e. DSR [23] or AODV [32], the nodes maintain only the routes that are currently in use. In this strategy, the current node holding a packet issues a destination search request, if the route is not available. The destination search is performed by flooding a short control message. Although reactive protocols discover the routes only when they are needed, they may still generate a huge amount of traffic when the network changes frequently [12]. Topology-based routing can usually find the optimal power path. However, they consume a lot of power for the nodes which are not part of the route which makes these algorithms the worst in terms of network survivability.

Position-based routing algorithms use position information to forward the packet in the geographical direction of the destination. In this type of routing, the node forwards the message based on the position of the node itself, the position of the destination and the position of the nodes to which it can communicate directly. Position-based routing is scalable to a large number of network nodes. Since there is no flooding it is the best choice to increase the network lifetime. Recent research in position-based routing usually addresses such routing algorithms in two-dimensional space (2D) [30,28,38,39]. However, in real applications, nodes may be distributed in 3D space such as city landscape, hilly terrain and airborne.

In this paper, we propose several 3D power-aware position-based routing algorithms which try to maximize the delivery rate and maximize the network connectivity time (the number of messages that can be sent by the whole network). The new algorithms are based on the idea of replacing the constant transmission power of the node with an adjusted transmission power during two stages: while discovering the neighboring nodes and during the routing process. Our simulation experiments, using the model of [21], show that the delivery rates of the new algorithms are very high and there is significant improvement on the network lifetime, up to twice the lifetime, compared with its associated power and none power-aware routing algorithms (Greedy and greedy-cost).

The rest of this paper is organized as follows. Section 2 is devoted to the background material. In Section 3, we briefly review some related works that are closely related to our proposed schemes. In Section 4, we give a detailed description of the new routing algorithms. Experimental results to demonstrate the much improved performance of the proposed methods in comparison with existing techniques are presented in Section 5. Finally, Section 6 draws the conclusions of the paper.

## 2. Background

### 2.1. Network model

We assume that the set of $n$ wireless hosts is represented by a point set $V$ in the 3D space, and all the network hosts have the same communication range $R$, which is represented as a sphere volume of radius $R$. Let $dist(u,v)$ be the Euclidean distance between the nodes $u$ and $v$: $dist(u,v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}$. Two nodes are connected by an edge if the Euclidean distance between them is at most $R$. The resulting graph is called a unit disk graph (UDG). For node $u$, we denote the set of its neighbors by $N(u)$. A path from $S$ to $D$ is a sequence of nodes $S = v_1, v_2, \ldots, v_k = D$, such that $v_i$ and $v_{i+1}$ are neighbors.

The Gabriel Graph (GG) [18,24] is a subgraph of a UDG graph $G$ that can be constructed locally as follows: given any two adjacent nodes $u$ and $v$ in $G$, the undirected edge $(u,v)$ belongs to GG if, and only if, no other node $w \in G$ is located in the sphere with diameter $uv$. It is known that if UDG is two-dimensional then the Gabriel subgraph is planar, where the edges intersect only at their common end-vertices, and that if the UDG is connected, then its Gabriel subgraph is also connected [8]. It has been proven that GG is a $2\pi\sqrt{2n-4}/3$-spanner of $G$ [7].

### 2.2. Routing problem

Position-based routing protocols assume that any node knows:

- The coordinates $(x, y, z)$ of its position, which can be obtained using a method like a global positioning system, or another such mechanism [26,10].
- The location of its neighbors using a periodical exchange of control messages.
- The location of the destination, e.g., by using a location service [22,29].

The position-based routing task is to find a path from a source node to a destination node. It uses the local information at each node to determine how to route the packet. We are interested in the following performance measures for routing algorithms: the delivery rate, which is the percentage of times that the algorithm succeeds in delivering its packet, and the network survivability, which can be measured by the remaining power in the maximally used node during a set of consecutive routing messages.

### 2.3. Power consumption wireless model

A general wireless model has been proposed in [33] in which the power consumption between two nodes at distance $d$ is proportional to $u(d) = d^{\alpha} + c$, where $\alpha (2 \leqslant \alpha \leqslant 6)$ is the path loss exponent in the power consumption model and $c$ is a constant which represents the energy consumed in computer processing and encoding–decoding at both transmitter and receiver. It has been proven in [39] that direct transmission is power optimal if $d \leqslant (c/(1 - 2^{1-\alpha}))^{1/\alpha}$. Otherwise, the greatest power savings are obtained when $d$ is divided into $n > 1$ equal length sub-intervals of size $(c/(\alpha - 1))^{1/\alpha}$.

## 3. Related position-based routing algorithms

In this section, we review some representative position-based routing algorithms that are closely related to our proposed approach. We briefly discuss their algorithmic methodologies as well as their limitations.

*Greedy routing* [16,24]: the current node $X$ forwards the packet to the neighboring node $A$ that minimizes the remaining distance to the destination $D$. Formally, $gdy(X, N(X), D) = A \in N(X) : dist(A, D) \leqslant dist(W, D)$ for all $W \in N(X)$. The same procedure is repeated until the destination node is reached or no such $A$ exists. This routing method suffers from the so-called *local minimum* phenomenon, in which a packet may get stuck at a node that does not have a neighbor that makes a progress to the destination, even though the source and the destination are connected by a path in the network.

Fin [16] proposes to flood all $\ell$-hop neighbors (nodes at distance at most $\ell$ hops from current node, where $\ell$ is network dependent parameter) until a node closer to destination than the current node $C$ is found. Takagi and Kleinrock [40] propose another way to counter this problem by forwarding the packet to the least backward (negative) progress. However, this raises the problem of looping packets.

*2D face routing* [8]: the routing takes place in a planar geometric subgraph of a network. The packets are routed over the faces of planar subgraph which are intersected by the line between the source $S$ and the destination $D$, called $SD$, using the right hand rule. That is, the boundary of $f$ is traversed in the counterclockwise direction, until the current edge crosses $SD$ at an intersection point closer to the destination than any previously discovered intersection point. In this case, the algorithm switches to the face containing destination and continues with the right hand rule. This algorithm is repeated until the node arrives at the destination. Face routing algorithm guarantees delivery only in a 2D planar geometric graph [17].

Face routing may be used in practice in combination with algorithms that usually find shorter routes, such as the *Greedy* algorithm. The idea is to use *Greedy* routing until a local minimum is reached, whereupon the algorithm enters into recovery mode by switching to face routing. When the local minimum is bypassed, the algorithm switches back to *Greedy* routing, and so on. This algorithm is termed *GFG* (*Greedy–Face–Greedy*) [8]. In 3D geometric graphs extracting a straight line planar graph is not possible since the notions of planarity and routing about the perimeters of faces do not exist. Recently, Durocher et al. [14] show that there can be no local position-based routing algorithm that guarantees delivery in 3D space if the thickness of one of three dimensions is more than $r/\sqrt{2}$, where $r$ is the transmission range.

Abdallah et al. [1] proposed $CFace(3)$ (Coordinate Face), a heuristic using a projective approach to adapt face routing to 3D graphs, see Fig. 1. The algorithm may be summarized as follows. The 3D nodes are first projected onto the $xy$ plane. Then face routing is performed on this projected graph. If the routing fails, i.e., a loop is detected where the route path is longer than a threshold, the nodes are then reprojected onto the second plane, the $yz$ plane. Then face routing is performed again. If the routing again fails, the nodes are projected onto the third plane, the $xz$ plane. Face routing is again performed. A simplified version of $CFace(3)$, called $CFace(1)$, attempts face routing with the

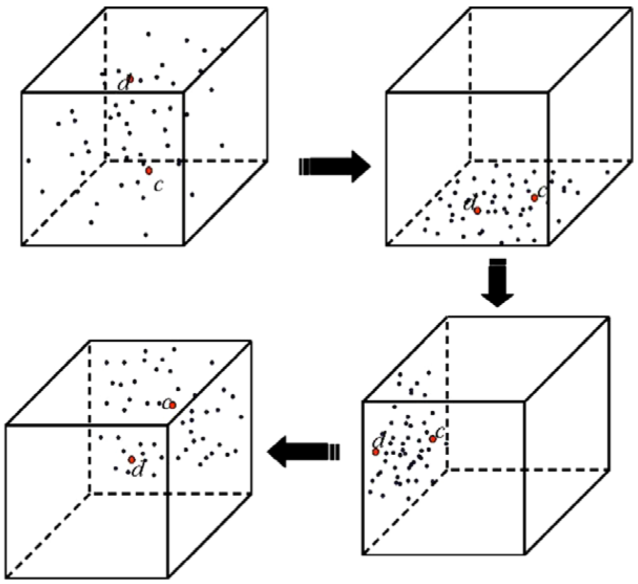

**Fig. 1.** $CFace(3)$ routing algorithm. The algorithm attempts 2D face routing, cyclically until success, with the nodes projected onto the $xy$ plane, the $yz$ plane, and then the $xz$ plane.

nodes projected once only onto one of the *xy*, *yz*, or *xz* planes, randomly chosen. A more elaborate version of projection-based face routing was proposed by Kao et al. [24,25]. In [1], the authors also proposed a combination between *CFace*(3) and randomized progress-based algorithms.

It is worth to state here that a projected graph is not UDGs, so any of the planarization algorithms on the projected graph may not give a planar graph and may not even give a connected graph [14,1].

## 4. Existing power-aware routing algorithms

Several power-aware routing algorithms, that try to minimize the total energy consumed by the packet and also increase the average network lifetime, have been proposed [28,38,39]. Let the current node be $X$, $A$ is a neighbor of $X$, and the destination is $D$. Let $h = dist(X,A), p = dist(X,D)$ and $q = dist(A,D)$, where $q < p$. Let the cost of transmitting a packet between two nodes at distance $d$ be $ad^{\alpha} + c$, where $a, \alpha, c$ are constants that depend on the wireless model. Let $\bar{h}$ be the average length of all edges out from the source $S$. Let $f(A) = \frac{1}{g(A)}$, where $g(A)$ is the remaining lifetime for the node $A$. Also, $\bar{f}(A)$ is the average value of $f(l)$ for $A$ and all neighbors $l$ of $A$, $\bar{g}(A)$ is the average value of $g(l)$ for $A$ and all neighbors $l$ of $A$, $R$ is the transmission radius. The algorithms are summarized as follows [39]:

*Power algorithm:* This algorithm tries to minimize the total energy consumed by the packet in the routing process, regardless of the available energy at the nodes. With the Power algorithm, the current node $X$ chooses as a next node $A$ which minimizes the expression: $P(X,A) + P(A,D)$, where $Q(X,A) = ah^{\alpha} + c$ is the cost to reach $A$ and $Q(A,D) = qc(a(\alpha - 1)/c)^{1/\alpha} + qa(a(\alpha - 1)/c)^{(1-\alpha)/\alpha}$, which is an assumption that the rest of the routing process is cost optimal.

*Cost-i algorithm*: This algorithm uses the cost metric. It tries to maximize the network lifetime by carefully choosing the next node from the set of neighbors with plenty of energy. In this algorithm the current node chooses a next node A which minimizes the equation: $cost(A) = f(A) * t/R$, where $t = \bar{f}(A)$.

*Cost-ii algorithm*: Since the factor $t$ is network dependent, there are different versions of the previous algorithm. One of those algorithms is called *Cost-ii*. In this algorithm, the current node chooses one of its neighbors, say $A$, which minimizes the equation: $cost(A) = f(A) * t/R$, where $t = 1/\bar{g}(A)$.

There are two ways to combine power and cost metrics into a single metric, based on the product or sum of the two metrics.

*Power * Cost*: In this algorithm, the current node chooses one of its neighbors, say $A$, which minimizes the following equation: $Power * Cost(A) = Power * Cost(X,A) + Power * Cost(A,D)$, where $Power * Cost(X,A) = f(A) * (ah^{\alpha} + c)$ and $Power * Cost(A,D) = (qc(a(\alpha - 1)/c)^{1/\alpha} + qa(a(\alpha - 1)/c)^{(1-\alpha)/\alpha}) * \bar{f}(A)$.

*Power + Cost*: In this algorithm the current node chooses one of its neighbors $A$ which minimizes the equation: $Power + Cost(A) = (Power + Cost(X,A)) + (Power + Cost(A,D))$, where $Power + Cost(X,A) = [\bar{f}(S) * (ah^{\alpha} + c)] + [f(A) * (a\bar{h}^{\alpha} + c)]$ and $Power + Cost(A,D) = qc(a(\alpha - 1)/c)^{1/\alpha} + qa(a(\alpha - 1)/c)^{(1-\alpha)/\alpha}) * \bar{f}(A)$.

Kuruvila et al. [28] proposed another set of power and cost-aware routing algorithms that choose the next node so to guarantee progress to the destination, if such a node exists. *Power Progress algorithm:* The current node forwards the packet to one of its neighboring nodes that is closer to the destination than itself and minimizes $(h^{\alpha} + c)/(p - q)$. Similarly, in *Cost Progress algorithm*, the next node is the one that is closer to the destination than the forwarding node and minimizes $f(A)/(p - q)$.

Since all the above algorithms are deterministic algorithms which suffer from the local minimum problem, they do not guarantee the delivery of the message in a connected graph. Stojmenovic et al. [38] propose guaranteed delivery algorithms in 2D space, which combine **Power** (**P**), **Cost** (**C**), and **Power * Cost** (**PC**) algorithms with Face routing algorithm, similar to the way *Greedy* algorithm is combined with Face to define the *GFG* algorithm. Those algorithms start with **P**, **C**, or **PC** forwarding decisions. Once a packet reaches a local minimum, the face routing starts. If the message arrives to a node closer to the destination than the local minimum node, the algorithm switches back to **P**, **C**, or **PC** forwarding again. These algorithms have been called **PFP**, **CFC**, and **PCFPC**, respectively; clearly these combined algorithms are not applicable in the 3D environment.

In [15], the authors proposed a power-aware routing protocol (MACRO) for sensors network which does not use periodic control messages but instead it uses an election algorithm to choose the next relay node on the fly. The protocol requires that each node knows its own position and the position of the destination, when a data packet arrives; it sends a message with a specific transmission power asking for neighbors. Each neighbor $A$ within that transmission range computes its *weight progress* factor ($WP_A$) according to the following formula: $WP_A = \frac{d(C,D) - d(A,D)}{P}$, where $P$ is the transmission range, $C$ is the current node. The neighbors answer with their weight progress. The current node increases its transmission power and repeats this process. The algorithm terminates when a timeout elapses or the probability of finding a better relay node is lower than a predefined threshold.

## 5. New power-aware position-based routing algorithms

Most of the routing algorithms without power-awareness use a fixed transmission range for all the nodes, so the nodes may waste power by transmitting more than is needed for correct reception. The above power-aware algorithms use an adaptive transmission range to transmit the data messages during the routing process, but they still use a fixed (maximum) transmission range for the control messages (periodic hellos) to tell neighboring nodes about their location. Our new power-aware routing algorithms are based on adjustments of the node transmission power at two stages: (i) while discovering the neighboring nodes, and (ii) during the routing process.

### 5.1. *Power adjusted greedy algorithm with half transmission range (PAGH)*

In PAGH (this algorithm has been proposed in a preliminary conference version of this paper in [2]) all nodes use a transmission range (HR) equal to half of the maximum transmission range, to discover their neighbors. This process is done periodically. Greedy routing is used between the source and the destination. If the packet reaches a local minimum (packet stuck at a node that does not have a neighbor that makes progress to the destination) at HR level, then the current node increases its transmission range to its maximum (MR) and runs neighbor nodes discovery step again. Fig. 2 gives an example of this point: when the message arrives to the node *B* that does not have any neighbor that makes a progress to the destination, *B* will increase its transmission range to MR to find new neighbors. Each node can adjust its transmission range just once while routing a single packet. If the node does not discover a new neighbor that makes progress to the destination, then the algorithm fails, otherwise greedy routing continues.

### 5.2. *Power adjusted greedy algorithm with optimal transmission range (PAGO)*

This algorithm is very similar to PAGH; the main difference is the value of transmission range for the periodic control messages. In PAGO we use the optimal transmission range ($\mathrm{OR} = (c/(\alpha - 1))^{1/\alpha}$) instead of the HR. The details of the algorithm are given in Algorithm 1.

**Algorithm 1.** *PAGO*

```
Input source node S, the destination node D.
Output: True if the packet arrives to D, and False otherwise.
for each node do
    if(MR < $(c/(\alpha - 1))^{1/\alpha}$) then
        OR ← MR
    else
        OR ← $(c/(\alpha - 1))^{1/\alpha}$
    end if
end for
Each node uses OR for the periodic neighbors discovery
while(the packet does not arrive to D)
    if(the current node does not have a neighbor that makes a progress to D)
        if(the current node uses OR)then
            The node uses MR to send a control message asking for neighbors.
            The nodes that make a progress to D inside the new sphere (Radius) send back a control message telling about their positions.
            if(there is a new neighbor that makes a progress to D)
                The current node sends the packet to the node that makes the greatest progress to D
            else
                return (False)
            end if
        else
            return (False)
        end if
    else
        The current node sends the packet to neighbor node that makes greatest progress to D.
    end if
end while
return true
```

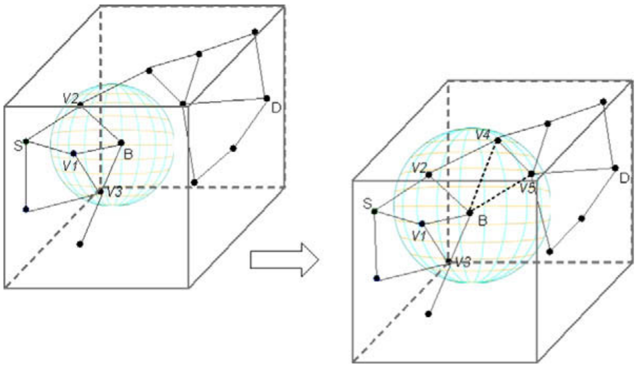

**Fig. 2.** *PAGH* algorithm, The packet arrives to the node *B* which is local minimum, so it increases the transmission range to the maximum.

### 5.3. *Power adjusted greedy algorithm with optimal transmission range and threshold (PAGR)*

This is another variation of the previous two algorithms, In PAGR; the nodes use the optimal transmission range (OR), if possible, for periodic discovery control messages. Whenever the degree of the node drops below a predefined threshold *DR*, it instead uses the maximum transmission range for the next periodic discovery control message, and then it goes back to use the optimal transmission range. If the degree is $< DR$, it continues using the maximum transmission range in the next two intervals and so on.

Table 1 gives an example of how a node changes its transmission range of the periodic discovery control messages depending on the node degree. In this example let $DR = 4$. Since the degree at time 0 is greater than 4, the node uses the optimal transmission range at the next time slot. At time 2 the node finds that the degree is dropped to less than 4 so it uses the maximum transmission range in the next time interval (3). At time interval (4) it tests the optimal transmission range again, and finds that the degree still less than *DR*, so it uses MR for the next two times intervals (5 and 6). At time interval (7), it test again the degree, because the degree is greater than *DR*, and uses OR for the next time interval (8) and resets the MR counter.

Similar to PAGH, and PAGO, *Greedy* routing is used for routing between the source and destination. If the packet reaches a local minimum at the low transmission level, then the current node increases its transmission range to its maximum and runs the neighbor nodes discovery step again. If a node does not discover a new neighbor that makes progress to the destination, then the algorithm fails, otherwise *Greedy* routing continues. The details of the algorithm are given in Algorithm 2.

**Table 1**
An example of how the node changes the transmission range of the periodic discovery control messages depending on the node degree, *DR*=4.

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| Transmission range | OR | OR | OR | MR | OR | MR | MR | OR | OR |
| Node degree | 5 | 6 | 3 | 7 | 3 | 8 | 7 | 5 | 4 |

**Algorithm 2.** *PAGR*

```
Input source node S, the destination node D and degree threshold DR.
Output: True if the packet arrives to D, and False otherwise.
for each node do
  if(MR < $(c/(\alpha-1))^{1/\alpha}$) then
    OR ← MR.
  else
    OR ← $(c/(\alpha-1))^{1/\alpha}$
  end if
end for
$Count_{MR} \leftarrow 0$
if(the node W uses OR, and has enough neighbors, greater than RD at time t) then
    W uses OR for discovering neighbors in the next time interval.
    $Count_{MR} \leftarrow 0$
else
    $Count_{MR} \leftarrow Count_{MR} + 1$
    W uses MR for discovering neighbors in the next $Count_{MR}$ time intervals.
end if
while(the packet does not arrive to D)
  if(the current node does not have a neighbor that makes a progress to D) then
    if(the current node does not use MR)
      The node uses the MR to send a control message asking for neighbors.
      The nodes that make a progress to D inside the new sphere (Radius) send back a control message telling about their positions.
      if(there is a new neighbor that makes a progress to D)
        The current node sends the packet to the node that makes the greatest progress to D
      else
        return (False)
      end if
    else
      return (False)
    end if
  else
    The current node sends the packet to neighbor node that makes greatest progress to D.
  end if
end while
return (True).
```

There are two main differences between *PAGR* and the non-power-aware *Greedy* algorithm. First, during the neighbor discovery phase, in *PAGR* all the nodes exchange periodic control messages with the optimal transmission range if possible. In *Greedy*, all the nodes exchange information by transmitting and receiving using the maximum transmission range. Also, during the routing phase (data packet routing process), in *PAGR* the current node $X$ forwards the message with a power cost equal to $ah^{\alpha} + c$, where $h$ is the distance between $X$ and the next node $A$, while *Greedy* forwards the message with power cost equal to $aR^{\alpha} + c$, where $R$ is the maximum transmission range.

### 5.4. Power-aware greedy-cost (PAGRC)

This algorithm is similar to *PAGR* algorithm, but the main difference is in the greedy part. It uses the Cost Progress idea from [28] as follows. When the packet arrives at some node $X$, instead of choosing the closest neighbor to the destination as a second node, *PAGRC* chooses the neighbor that makes progress to the destination and has the maximum remaining energy, where the power level of the neighbors can be obtained from the periodic control messages.

### 5.5. PAGR:CFace(1):PAGR

The previous algorithms *PAGH*, *PAGO*, *PAGR*, *PAGRC* and their associated fixed power *Greedy* algorithm have a great advantage in terms of power saving. In our simulations, though, they suffer from low delivery rates if the network is very sparse. Our solution is to use *CFace* routing if the *PAGR* algorithm fails to deliver the message. The combination is called *PAGR*:*CFace*(1):*PAGR*.

*PAGR*:*CFace*(1):*PAGR* starts with *PAGR* algorithm. Once a local minimum is reached at low transmission range, the current node holding the packet adjusts the transmission range. If it stays in the local minimum situation the algorithm extracts locally the 3D-GG graph from the UDG and projects it on one plane, which is randomly one of the *yx*, *yz* or *xz* planes, and then it switches to *CFace*(1). *CFace*(1) traverses that projective plane starting from the local minimum node as the new source node. When the packet reaches a node closer to the destination than the local minimum, the algorithm switches back to the *PAGR* routing algorithm on the UDG, and so on. In this combined algorithm, we used a threshold to stop any potential loop that may happen during the *CFace* phase, because *CFace* is not a loop-free algorithm. When the path length during routing exceeds the threshold, the algorithm is deemed to have failed.

*PAGR:CFace(1):PAGR* is summarized in Algorithm 3. Fig. 3 shows an example of this algorithm: when the increasing of the transmission range at the local minimum node $B$ did not give it any new neighbor that makes progress, the algorithm extracts the 3D-GG and projects it on the *xz* plane and switches to *CFace*(1). If face routing passes the local minimum, reaching a node closer to the destination than the local minimum, it switches back to *PAGR*.

**Algorithm 3.** *PAGR*:*CFace*(1):*PAGR*

```
 1: Input source node S, the destination node D and the path threshold ph
 2: Output: True if the packet arrives to D, and False otherwise.
 3: Let C be the current node holding the packet during the routing algorithm
 4: if (C is a neighbor to D)
 5:     return (True)
 6: end if
 7: if(the path length > ph)
 8:     return (False)
 9: enf if
10: call PAGR algorithm.
11: if(C adjusts its transmission range, and after that, it stays in the local minimum situation) then
12:     L ← C
13:     Extract 3D-GG from the UDG
14:     Project the 3D-GG on xy, yz or xz plane, which is chosen randomly,
15:     Call Face routing on the projected graph starting from L
16:     During Face routing in 14:
17:     if(dist(C, D) < dist(L, D))
18:         call(PAGR:CFace(1):PAGR) starting from the current node C
19:     end if
20: end if
```

## 6. Simulation and results

### 6.1. Simulation environment

#### 6.1.1. Wireless model

We use in our simulation the wireless model that has been proposed by Heinzelman et al. [21] which assumes the radio dissipates $E_{elec} = 50nJ/bit$ power to run the transmitter or receiver circuitry. Both transmitter and receiver nodes consume $E_{elec}$ to transmit one bit. The radio dissipates $E_{amp} = 100\ \mathrm{pJ/bit/m^2}$ to run the transmit amplifier, assuming $d^2$ energy loss due to channel transmission, where $d$ is the distance between nodes. This implies the sender consumes $(E_{amp} * d^2)$ power to transmit one bit. According to the above wireless model, transmitting a $k$-bit message at distance it d the transmitter expends $E_{Tx}(k,d) = E_{elec} * k + E_{amp} * k * d^2$. To receive it, the receiver node expends $E_{Rx}(k) = E_{elec} * k$. From the model above, it is clear that the final expression to transmit one bit message equal to $d^2 + 2000$, hence, $c = 2000$ and $\alpha = 2$ which means the optimal transmission range $(c/(\alpha - 1))^{1/\alpha}$ is equal to 44.7.

#### 6.1.2. Simulation setup

The simulator can be described as follows:

1. We define a box of side length equal to 250 units.
2. In each box we generate 200 nodes, positioned randomly, such that node $i$ has the coordinate $x_i,\ y_i,\ z_i$, where $0.0 \leqslant x_i, y_i, z_i \leqslant 250.0$.
3. All nodes start with the same energy level and same maximum possible transmission range.
4. Using the maximum transmission range, the UDG graph is calculated for the nodes. If the resulted graph is connected, we use this set of nodes for routing. Otherwise, a new set of nodes is randomly generated.
5. The routing algorithms were tested at different maximum transmission ranges (different node average degree). The average node degree is calculated as follows: For each node $u$, we calculate the number of other nodes in the sphere centered at $u$ with radius $r$, which gives the node degree for $u$. Then the average node degree is equal to $(\sum_{i=1}^{n}$ degree of node $i)/n$.
6. Initially, for the algorithms PAGO and PAGR, the transmission range for all nodes is set to 44.7 with the possibility that the node will increase its transmission range to its maximum.
7. For the algorithms with a constant transmission power, their transmission range is set to the maximum.
8. For each accepted set of nodes, we choose a set of 1000 source–destination pairs. The routing algorithms are then applied on the chosen source–destination pairs. An algorithm succeeds if a path to the destination is found. The results of these experiments have 95% confidence intervals.
9. A control message is sent from each node at the beginning of the routing algorithm between any (source–destination) pairs.
10. We find the node which has the lowest power after step 8. In addition, the number of successful packets from a source to its associated destination is determined.
11. To compute the average packet delivery rate, and the average remaining power for the maximum used node, this process is repeated with 10 random sets of nodes, and the percentage of successful deliveries is determined.
12. All the power consumption results are computed just from the delivered packets. If the packet fails to arrive to the destination, we roll back the power level of all nodes to their previous level before that packet starts its routing process.
13. For these simulations, we used data packets of size 16 bytes and control packets of 6 bytes.
14. We have used different maximum transmission ranges 55, 60, 65, 70, 75, 80, 90 and 100 to test our algorithms. In each figure, we show the corresponding average nodes degree.
15. We set the values of *DR* in PAGR algorithm to four in our experiments. A threshold equal to the number of nodes is used in the hybrid algorithms.
16. We use the wireless model described in the previous section for transmitting the packets from each source to its associated destination.

### 6.2. Observed results

Fig. 4 shows the delivery rate of *PAGH*, *PAGO*, *PAGR*, *PAG-RC*, and the other power-aware routing algorithms, the delivery rate of *Greedy* algorithm is shown in the same figure. It is immediately evident from this figure that the delivery rate of *PAGR* is higher than the delivery rate of *PAGO* and *PAGH* algorithms, this can be explained by the periodic discovery messages, *PAGR* algorithm uses OR to

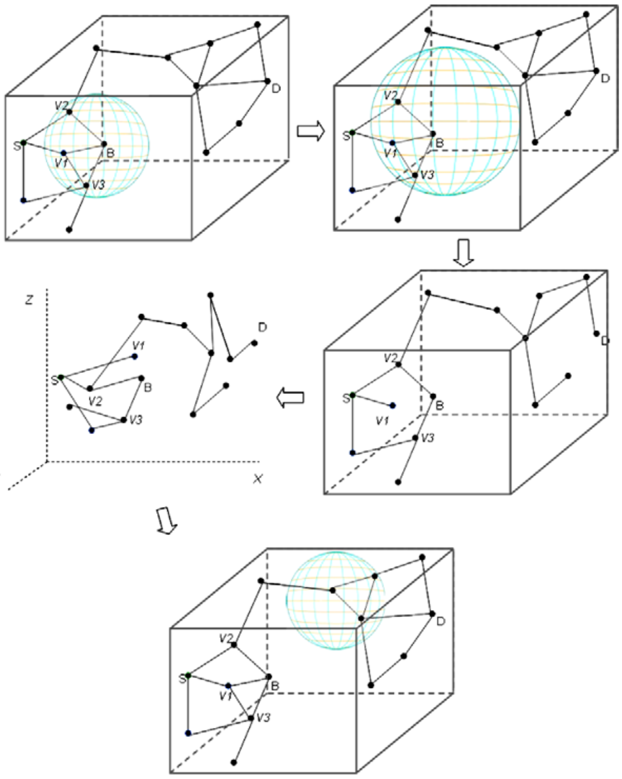

**Fig. 3.** *PAGR*:*CFace*(1):*PAGR* algorithm steps.

sense the degree of the node, so if the node degree drops below the threshold (DR), it uses MR right away to discover the neighbors, while *PAGO* and *PAGH* do not use MR unless the packet reaches a local minimum which might be too late to get out of this local minimum.

As shown in the same figure, *Greedy* algorithm for the low nodes degree gives the highest delivery rate, because *Greedy* algorithm uses MR all the time which decrease the chance of encounter a local minimum. At degree 3, there are 1.5 neighbors on average with forward progress. Thus in 50% of hops there are two options for forwarding neighbors. Thus PAGR frequently selects different forwarding neighbor from greedy algorithm even in very sparse networks, leading to 20% lower success in simulations. *MACRO* algorithm has a bit higher delivery rate than *PAGR* algorithm at low nodes degree, because *MACRO* keeps increases the power until it find a power efficient relay node.

All studied algorithms have a low delivery rate if the network is very sparse (low average nodes degree) and a 100% delivery rate at high average nodes degree. This can be explained by the number of neighbors. Fewer neighbors implies less chance to choose a good route that makes progress to the destination.

For all delivered packets, we show in Fig. 5 the average power consumed from the maximum used node, the new algorithms *PAGH* and *PAGRC* decrease the average power used by the maximum used node by around 40%, compared to those algorithms with a fixed transmission radius. The other power-aware routing algorithms do not gain more than a 4% increase in the network lifetime. *MACRO* algorithm consumes even more power than *Greedy* at low nodes degree.

To have a fair comparison we tried to normalize the two metrics, delivery rate and power consumption, by dividing the power consumed from the maximum used node over the delivery rate. Fig. 6, shows these normalized metrics. It is clear that our *PAGH*, *PAGRC*, and *PAGR* have better power saving in both high and low nodes degree. *PAGO* and *MACOR* starts saving power when nodes degree is higher than 6. As expected *greedy* has highest power consumption, simply because it uses the maximum transmission range all the time for both periodic discovery messages and real packets forwarding.

The performance results for our new algorithms *PAGR: CFace(1):PAGR* and *Greedy*:*CFace*(1):*Greedy* are shown in Figs. 7–9. These algorithms try to increase the delivery rate for sparse network (low nodes degree). From Fig. 7, *Greedy*:*CFace*(1):*Greedy* has increased the delivery rate to around 100% for any nodes degree. *PAGR:CFace(1):PAGR* has increased the delivery rate to above 93% for low nodes degree networks and 100% if the degree is above 6.

As shown in Figs. 8 and 9, the average power consumed by the most used node at the low degree (less than 6) for *PAGR*:*CFace*(1):*PAGR* and *Greedy*:*CFace*(1):*Greedy* is higher than other power-aware routing algorithms, this can be explained as follows:

First, *CFace* routing is not a loop-free algorithm, which makes it very power consuming if it enters a loop (it stops when the path length is longer than a given threshold). From Fig. 4, it is clear that the delivery rate of *PAGR* and *Greedy* at the degree less than 6 is very low. We know that both of *PAGR*:*CFace*(1):*PAGR* and *Greedy*:*CFace*(1):*Greedy* use *CFace* routing if *PAGR* and *Greedy* fail to deliver the packet. Thus *PAGR*:*CFace*(1):*PAGR* and *Greedy*:*CFace*(1):*Greedy*, have high average energy consumption at these transmission ranges.

Second, the other reason which makes *PAGR:CFace(1):PAGR* average power consumption for the maximum used node more than *Greedy*:*CFace*(1):*Greedy*, at low nodes degree, is the adaptive transmission range for the periodic control messages. When *PAGR* use the optimal transmis-

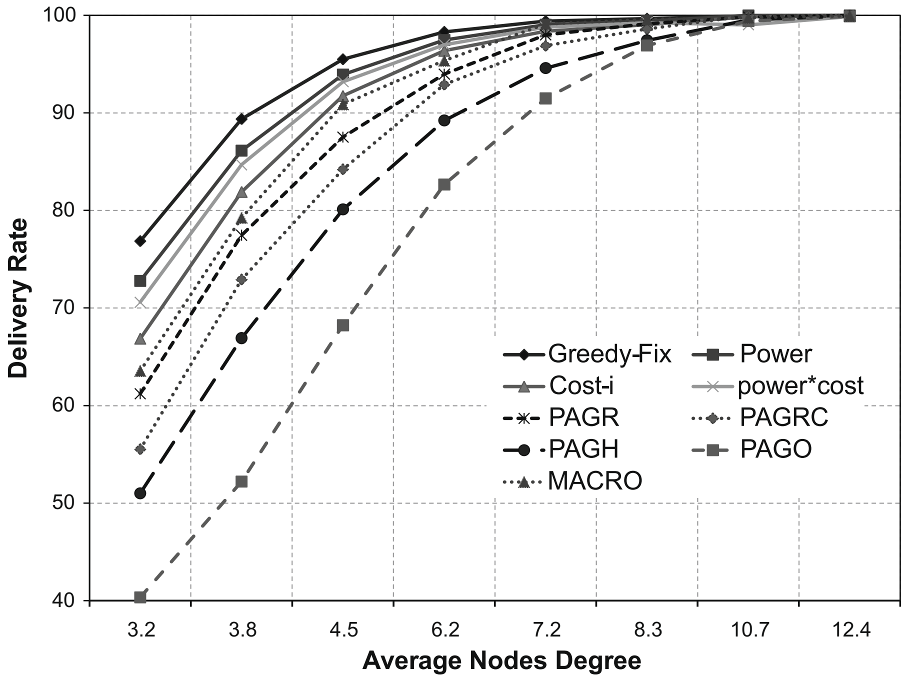

**Fig. 4.** The average delivery rate of *PAGR*, *Greedy* and other different power-aware routing algorithms.

sion range and determines that there are not enough neighbors, it sends using the maximum transmission range for a period of time before trying optimal again. Most of the time, sending using the optimal transmission range would just increase the overhead and the power consumption if the average node degree is small, and this is clear at the nodes degree 2.3 and 3.8 which makes the graphs more tree-like. If the average nodes degree is greater than 6, *PAGR*:*CFace*(1):*PAGR* has a longer network lifetime up to 25% more than *Greedy*:*CFace*(1):*Greedy*.

The most important advantage of all our new algorithms is the substantial increase of the network lifetime while preserving the delivery rates. The same simulations were done using the *Compass* routing algorithm [27] in place of the *Greedy* algorithm, generating new routing algorithms, called *PACR*, and *PACR*:*CFace*(1):*PACR*. The re-

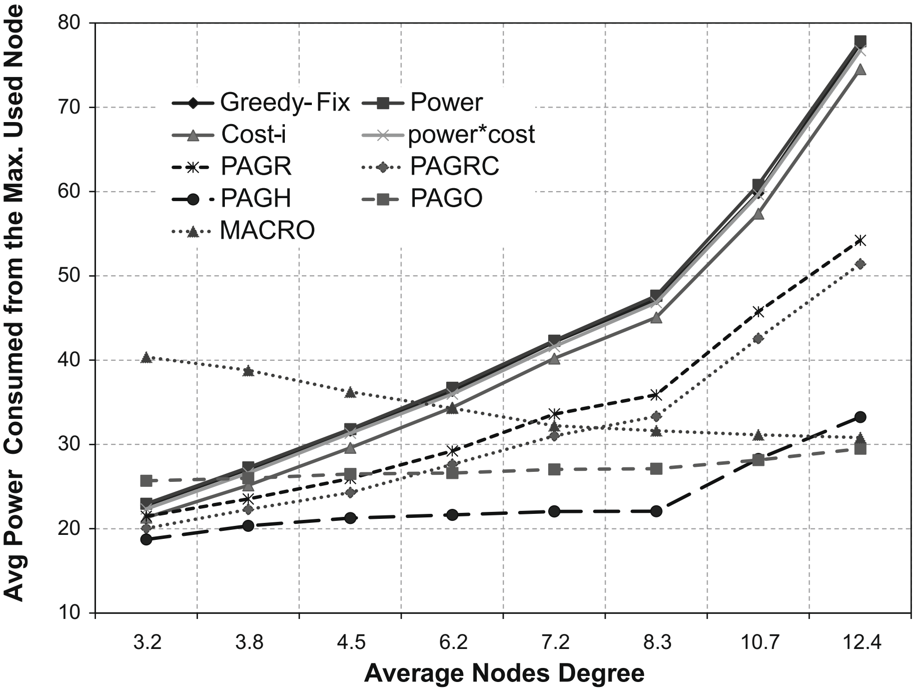

**Fig. 5.** The average power consumption for the maximum used node after 1000 source destination pairs routing process, for all delivered packets.

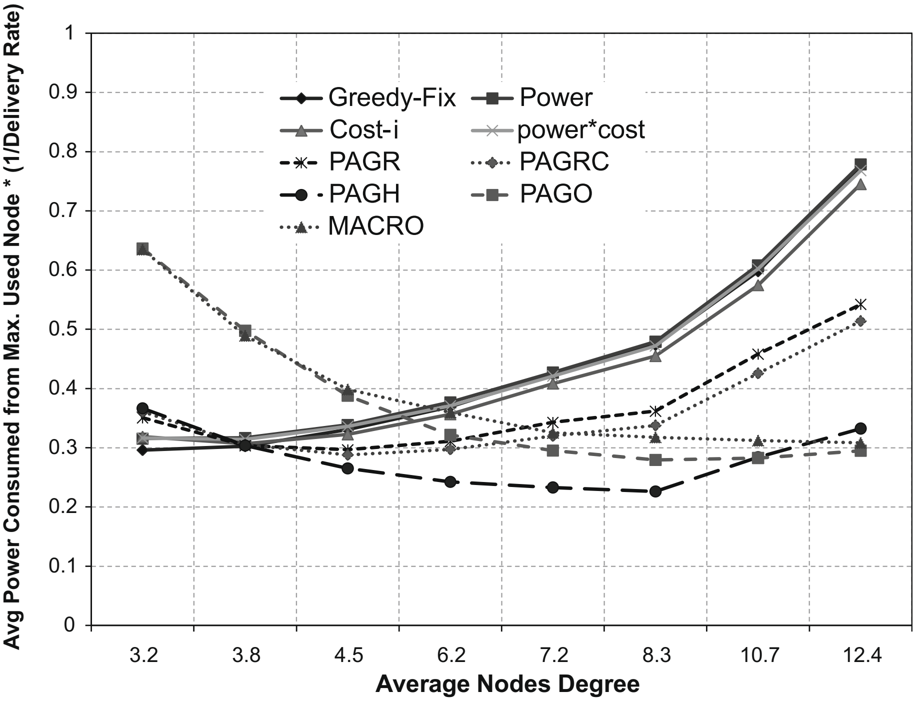

**Fig. 6.** A normalized metric of the delivery rate and the average power consumption for the maximum used node after 10 000 source destination pairs routing process.

sults are nearly the same as for the *Greedy-based* algorithms.

In order to evaluate the energy consumption while having nearly 100% delivery rate for the proposed algorithms, we use dense networks with average node degrees around 20. This average node degree has been proposed previously in [28,3]. We use a simulation environment similar to the environment above with the following differences:

1. A set of 205, 600, 1300, 2450, 4200, 6450 or 9500 random nodes is generated in different boxes of side length 200, 300, 400, 500, 600, 700 or 800, respectively.
2. The fixed maximum transmission range is set to 66 units.

In this environment, the success rate for all the algorithms discussed in Sections 3 and 4 is always around

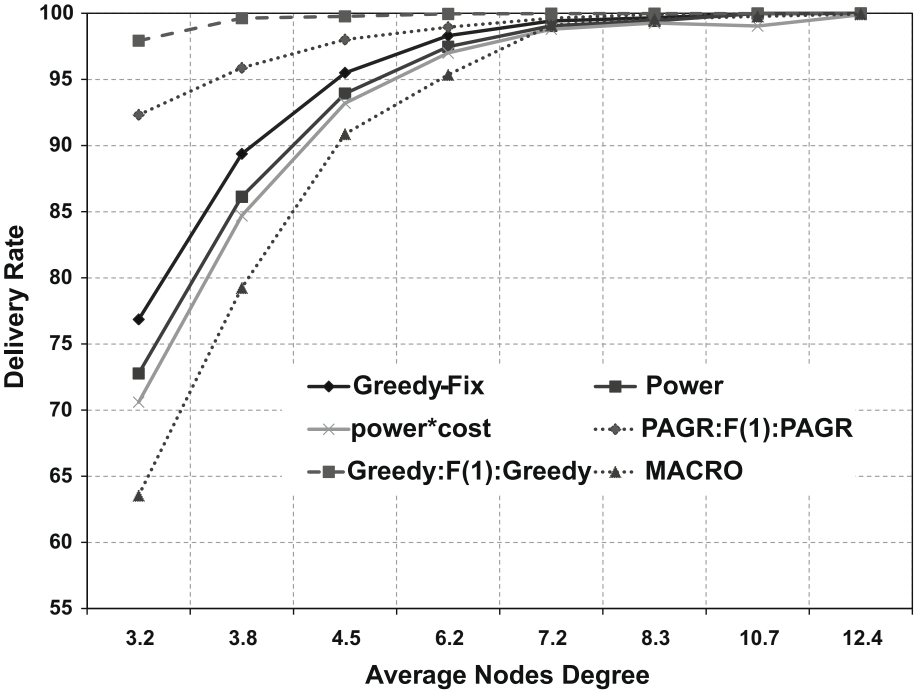

**Fig. 7.** The delivery rate for *PAGR*:*CFace*(1):*PAGR*, *Greedy*:*CFace*(1):*Greedy* and the other power-aware routing algorithms.

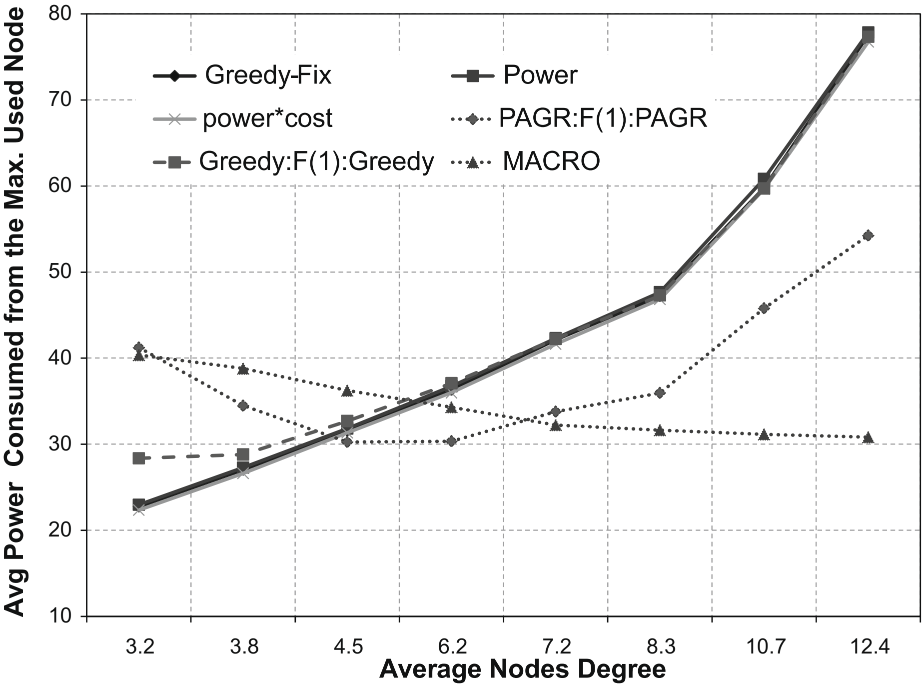

**Fig. 8.** The average power consumption for the maximum used node after 1000 source destination pairs routing process of *PAGR*:*CFace*(1):*PAGR*, *Greedy*:*CFace*(1):*Greedy* and other power-aware routing algorithms.

100%, therefore the success rate will not be a variable in the measurements. Also, there is no need to test the combined algorithms *PAGR*:*CFace*(1):*PAGR* and *Greedy*:*CFace*(1):*Greedy* in this environment, since the graphs are so dense, switching to face routing rarely occurs.

Fig. 10 shows the average power consumed for the maximum used node in *PAGH*, *PAGR*, *PAGO*, and other routing algorithms. It is clear from this figure that the average power consumed for maximum used node of the new algorithms *PAGO* and *PAGR* is decreased by around 45%, which in turn increases the network lifetime to around twice that compared to the fixed transmission radius algorithm *Greedy*.

As expected *MACRO* algorithm has a Slightly less power consumption than our proposed algorithms at very dense network, because this algorithm does not use any periodic hello messages, which consumes more power in dense network, receiver will receives more messages from all neighbors regularly.

We can explain the little increase of the average power consumed by the most used node for *PAGO* and *PAGR* routing algorithms when the box size increases, even though we have approximately the same average node degree at different box sizes, as follows: Since there are more nodes in the box and the box size is bigger, to maintain nearly the same average node degree, longer paths will be taken by the routing algorithms. This is especially true for the *PAGR* and *PAGO* algorithms because they use the optimal transmission range. This results in an increase the average power consumption for the maximum used node.

### 6.2.1. *Effect of control message frequency*

In order to test the effect of the control messages frequency on our proposed algorithms, we fixed the maximum transmission range to 75 (7.4 average nodes degree) and vary the control messages frequency. In this section, we send a control message from each node at following frequency 10 times during the routing process, 5 times, 2 times, 1 S-D (at the beginning of the routing process between each source–destination pairs), 2 S-D (at the beginning of every second source–destination pairs), 5 S-D, and 10 S-D.

Because the nodes are static and the network density is static too, the delivery rate is fixed for each algorithm. Actually even if the network is mobile, we do not expect it will affect the delivery rate much because mobility can be reported in advance (speed and direction), so node can know exactly which nodes are still neighbors. These neighbors are then used at their revised (exact and correct) positions.

Fig. 11 shows the average power consumption from the maximum used node, it is expected that *PAGR*, consumes more power than *PAGO* and *PAGH*, because some nodes in *PAGR* uses the maximum transmission range for periodic hello messages depending on the node degree. Because the maximum transmission range we have chosen for this simulation setup is 75, *PAGO* (uses transmission range equal to 44.7 for the periodic hello messages) consumes more power than *PAGH* (uses transmission range equal to 37 for the periodic hello messages).

### 6.2.2. *Effect of sleep-active nodes*

Each node in *PAGR* algorithm changes the transmission range of the periodic discovery control messages depending on the node degree; this can be tested by having nodes sleeps at random paces. In this section, we tested our proposed algorithms on the following sleeping ratios: 5%, 10%, 15%, 20%, 25%, 30%, and 40%.

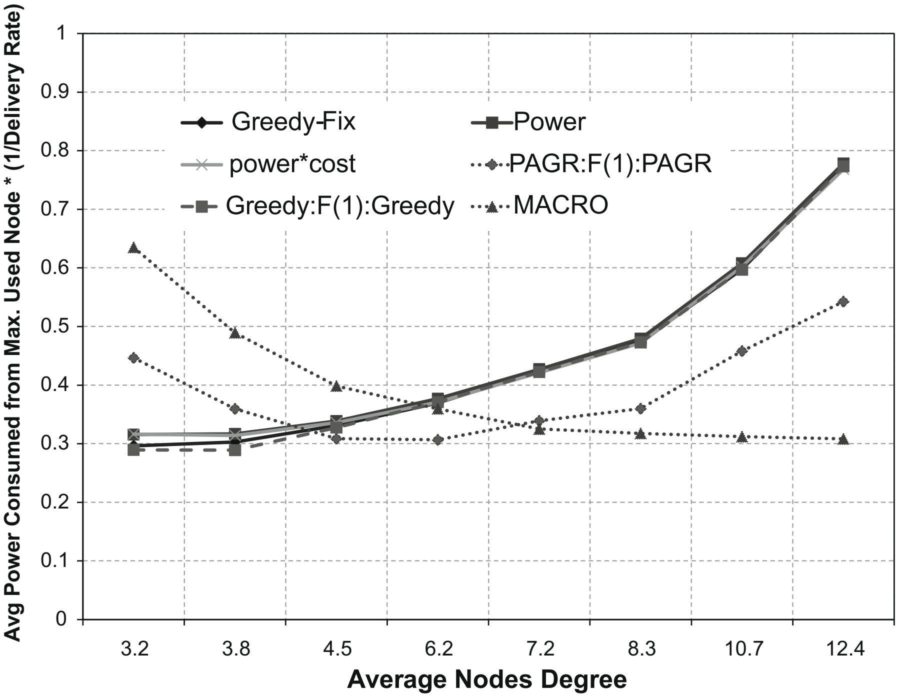

**Fig. 9.** A normalized matric of the delivery rate and the average power consumption for the maximum used node after 1000 source destination pairs routing process.

Fig. 12 shows the delivery rate of *PAGR*, *PAGO*, and *PAGH* algorithms, we can see that these algorithms slightly effected by the sleeping nodes, this because each node can tell all neighbors nodes about its coming situation sleep or active, only the nodes that wakes up who did not send yet hello messages.

Fig. 13 shows the power consumption from *PAGR*, *PAGO*, and *PAGH*. We can see that the power consumption is de-

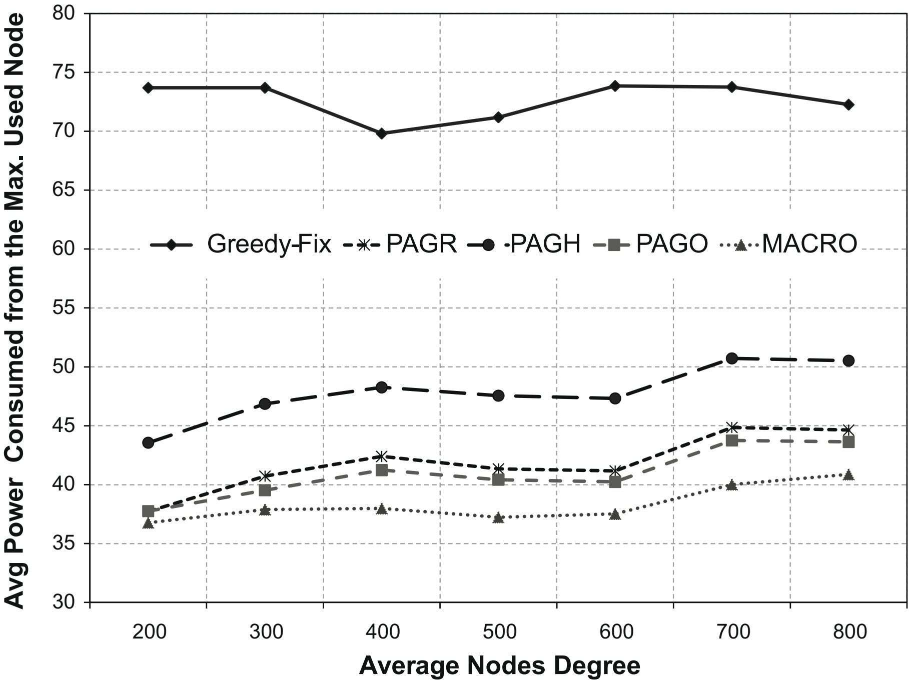

**Fig. 10.** The average power consumption for the maximum used node after 2000 source destination pairs routing process of *PAGR* and other power-aware algorithms.

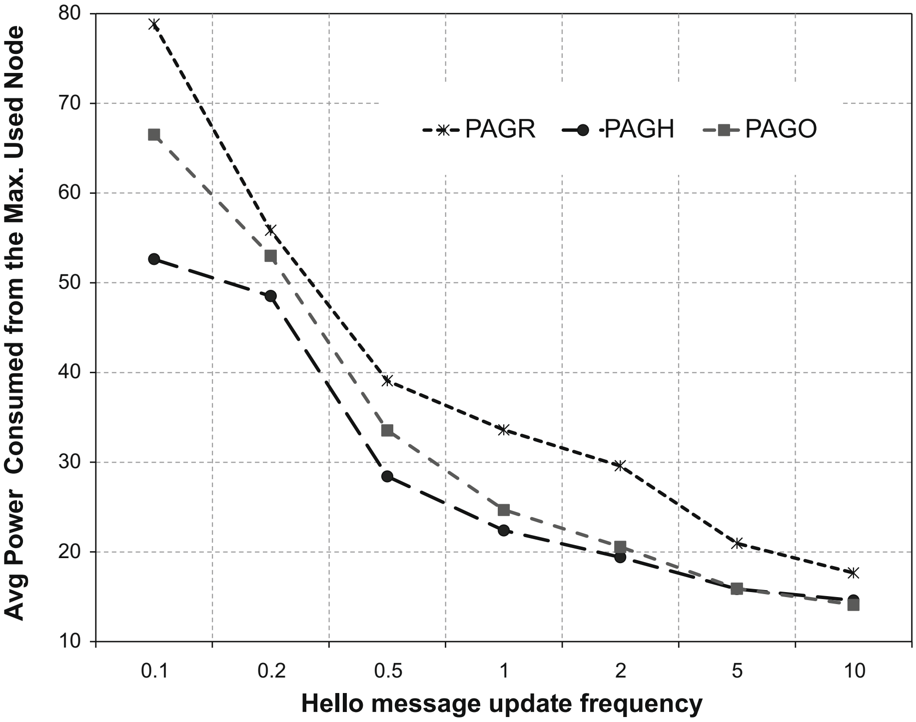

**Fig. 11.** The average power consumption for the maximum used node after 2000 source destination pairs routing process of *PAGR* and other power-aware algorithms at different periodic hello messages frequency.

creased by increasing the number of sleeping nodes, this can be explained by the number of nodes: less number of nodes means less path length, thus less power consumption. We can also see that *PAGR* starts consuming more power than *PAGH* when the sleeping percentage is more than 15%, also this can be explained by the average nodes degree, while increasing the number of sleeping nodes, we actually increasing the number of nodes that have degree less than RD, these nodes uses MR for hello messages, thus increase the power consumption. Fig. 14 shows the normalization of the two metric delivery rate and power consumption.

## 7. Conclusion

This paper describes several localized power-aware routing algorithms for 3D ad hoc network under two concurrent constraints: maximize the delivery rate while maximizing the lifetime of the network by minimizing the energy con-

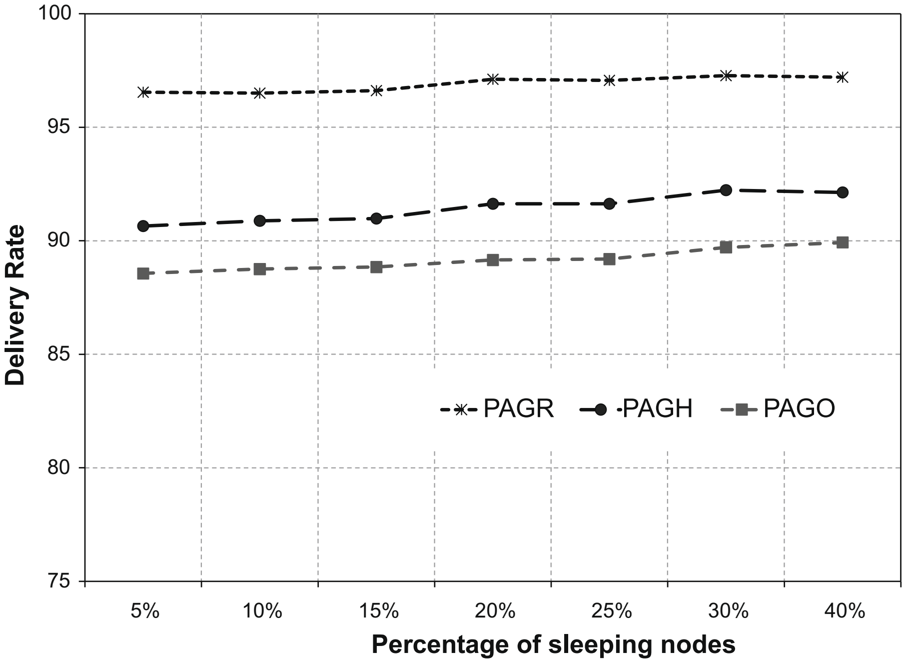

**Fig. 12.** The delivery rate of *PAGR* and other power-aware algorithms at different sleeping nodes ratio.

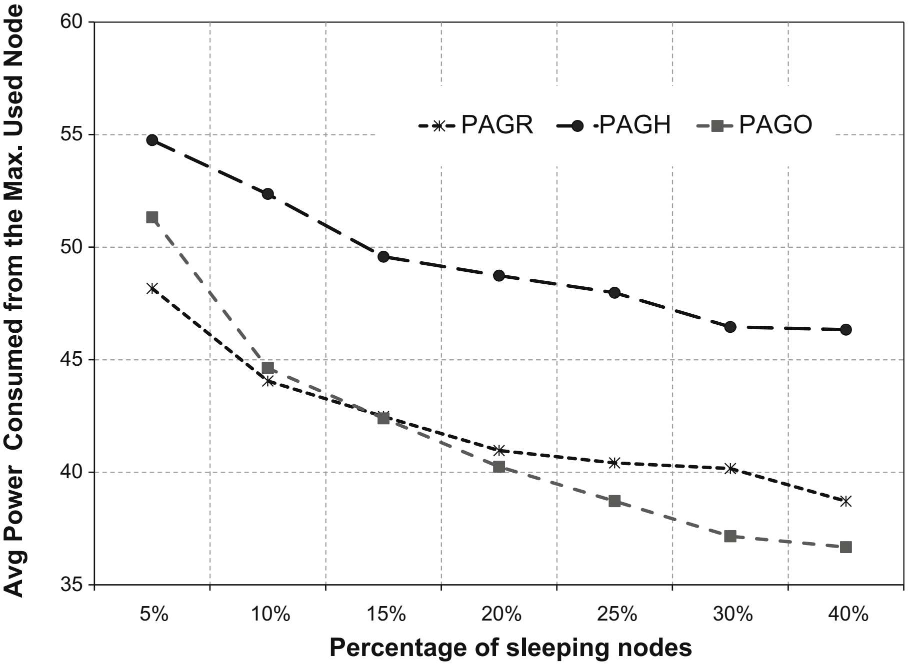

**Fig. 13.** The average power consumption for the maximum used node after 2000 source destination pairs routing process of *PAGR* and other power-aware algorithms at different sleeping nodes ratio.

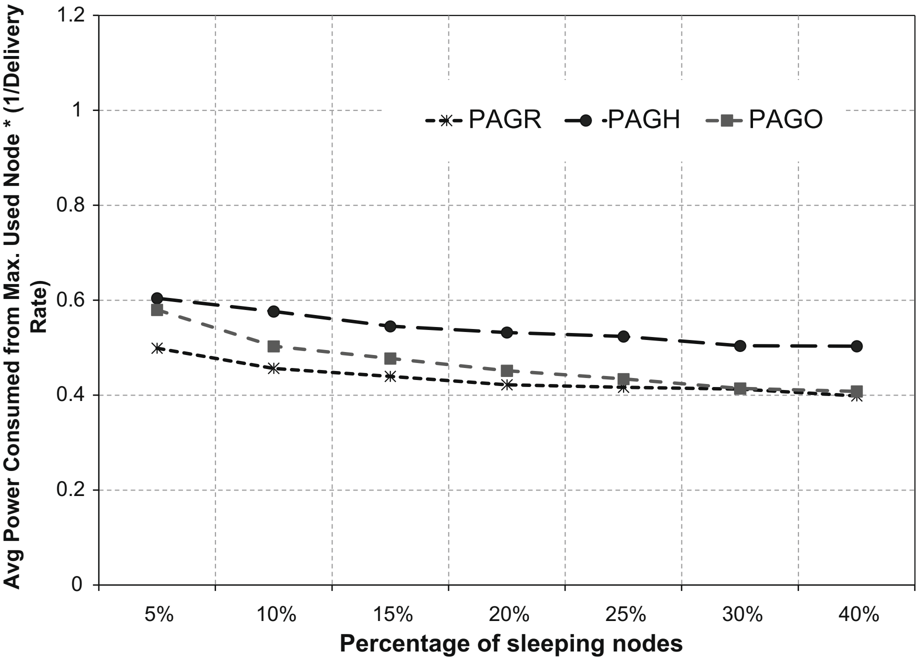

**Fig. 14.** A normalized matric of the delivery rate and the average power consumption for the maximum used node after 2000 source destination pairs routing process at different sleeping nodes ratio.

sumption by the nodes. Our new algorithms are based on the idea of replacing the constant transmission power of the node with an adjusted transmission power. The simulation results demonstrate that the new routing algorithms *PAGR*, *PAGO* with dense network have a delivery rate near 100% and increases the network lifetime to around twice that for the *Greedy* algorithm. Our algorithm, *PAGR:CFace(1):-PAGR*, give a significant increase in the delivery rate for sparse networks while increasing the network average lifetime.

## Acknowledgements

This research is partially supported by NSERC CRDPJ 319848 - 04 (Scalable, secure and localizes data communication and topology control protocols for wireless ad hoc networks) and NSERC discovery grants.

## References

[1] A.E. Abdallah, T. Fevens, J. Opatrny, Randomized 3-d position-based routing algorithm for ad-hoc networks, in: Proc. of the Third Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS), San Jose, July, 2006, pp. 1–8.

[2] A.E. Abdallah, T. Fevens, J. Opatrny, Power-aware 3d position-based routing algorithms for ad hoc networks, in: Proc. of the IEEE International Conference on Communications ICC-2007, Glasgow, June 2007, pp. 1–6.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40 (8) (2002) 102–114.

[4] J. Aslam, Q. Li, D. Rus, Three power-aware routing algorithms for sensor networks, Wireless Communications and Mobile Computing 2 (3) (2003) 187–208.

[5] L. Barrière, P. Fraigniaud, L. Narayanan, J. Opatrny, Robust position-based routing in wireless ad hoc networks with irregular transmission ranges, Wireless Communications and Mobile Computing Journal 3 (2) (2003) 141–153.

[6] S. Basagni, I. Chlamtac, V. Syrotiuk, Dynamic source routing for ad hoc networks using the global positioning system, in: Proc. of the IEEE Wireless Communications and Networking Conference (WCNC'99), New Orleans LA, September 1999, pp. 21–24.

[7] P. Bose, L. Devroye, W. Evans, D. Kirkpatrick, On the spanning ratio of gabriel graphs and beta-skeletons, in: Proc. of the Latin American Theoretical Infocomatics (LATIN), Mexico, April 2002, pp. 479–493.

[8] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Wireless Networks journal 7 (6) (2001) 609–616.

[9] J. Broch, D. Maltz, D. Johnson, Y. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, Dallas TX, October 1998, pp. 85–97.

[10] S. Capkun, M. Hamdi, J.-P. Hubaux, Gps-free positioning in mobile ad-hoc networks, Cluster Computing Journal 5 (2) (2002) 118–124.

[11] C. Chang, L. Tassiulas, Maximum lifetime routing in wireless sensor networks, IEEE/ACM Transactions on Networking 12 (4) (2004) 609–619.

[12] I. Chlamtac, M. Conti, J. Liu, Mobile ad hoc networking: imperative and challenges, Ad Hoc Network Journal 1 (1) (2003) 13–64.

[13] T. Clausen, P. Jacquet, Optimized link state routing protocol (OLSR), in: RFC 3626, IETF Network Working Group, October 2003.

[14] S. Durocher, D. Kirkpatrick, L. Narayanan, On routing with guaranteed delivery in three-dimensional ad hoc wireless networks, To Appear in the Proc. of the Ninth International Conference on Distributed Computing and Networking (ICDCN 2008), India, January 2008.

[15] D. Ferrara, L. Galluccio, A. Leonardi, G. Morabito, S. Palazzo, Macro: an integrated MAC/routing protocol for geographic forwarding in wireless sensor networks, in: Proc. of the IEEE INFOCOM, Miami, March 2005, pp. 1770–1781.

[16] G. Fin, Routing and addressing problems in large metropolitan-scale internetworks, Technical Report ISU/RR-87-180, USC ISI, Marina del Ray, California, 1987.

[17] H. Frey, I. Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks, in: Proc. of the 12th Annual International Conference on Mobile computing and Networking, Los Angeles, September 2006, pp. 390–401.

[18] K. Gabriel, R. Sokal, A new statistical approach to geographic variation analysis, Systematic Zoology 18 (3) (1969) 259–278.

[19] S. Giordano, I. Stojmenovic, L. Blazevic, Position based routing algorithms for ad hoc networks for ad hoc networks: a taxonomy, in:

Cheng, X. Huang, D.Z. Du (Eds.), Ad hoc wireless Networking, Kluwer, pp. 103–136.

[20] I. Haque, C. Assi, J. Atwood, Randomized energy aware routing algorithms in mobile ad hoc networks, in: Proc. of the Eighth ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2005), Montreal, October 2005, pp. 71–78.

[21] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocols for wireless microsensor networks, in: Proc. of the International Conference on System Sciences, Hawaii, January 2000, pp. 3005–3014.

[22] J. Hightower, G. Borriello, Location systems for ubiquitous computing, Computer 34 (8) (2001) 57–66.

[23] D. Johnson, D. Malts, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic Publishers, 1996, pp. 153–181 (chapter 5).

[24] G. Kao, T. Fevens, J. Opatrny, Position-based routing on 3D geometric graphs in mobile ad hoc networks, in: Proc. of the 17th Canadian Conference on Computational Geometry (CCCG'05), Windsor, August 2005, pp. 88–91.

[25] G. Kao, T. Fevens, J. Opatrny, 3-d localized position-based routing with nearly certain delivery in mobile ad hoc networks, in: Proc. of the Wireless Pervasive Computing, ISWPC'07, San Juan, February 2007, pp. 344–349.

[26] E. Kaplan, Understanding GPS, Artech house (1996).

[27] E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in: Proc. of the 11th Canadian Conference on Computational Geometry (CCCG'99), Vancouver, August 1999, pp. 51–54.

[28] J. Kuruvila, A. Nayak, I. Stojmenovic, Progress and location based localized power aware routing for ad hoc and sensor wireless networks, in: Proc. of the Third International Conference on AD-HOC Networks and Wireless ADHOC-NOW, Vancouver, July 2004, pp. 294–299.

[29] J. Li, J. Jannotti, D. De Couto, D. Karger, R. Morris, A scalable location service for geographic ad-hoc routing, in: Proc. of the Sixth ACM International Conference on Mobile Computing and Networking (MobiCom), Boston, August 2000, pp. 120–130.

[30] M. Mauve, J. Widmer, H. Hartenstein, A survey of position-based routing in mobile ad-hoc networks, IEEE Network Magazine 15 (6) (2001) 30–39.

[31] C. Perkins, P. Bhagwat, Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers, in: Proc. of the SIGCOMM, Conference on Communications Architectures, Protocols and Applications, London, September 1994, pp. 234–244.

[32] C. Perkins, E. Royer, Ad hoc on-demand distance vector routing, in: Proc. of the Second IEEE Workshop on Mobile Computing System and Application (WMCSA), San Juan, February 1999, pp. 90–100.

[33] R. Ramanathan, R. Rosales-Hain, Topology control of multihop wireless networks using transmit power adjustment, in: Proc. of the IEEE INFOCOM, March 2000, pp. 404–413.

[34] V. Rodoplu, T. Meng, Minimum energy mobile wireless networks, IEEE Journal Selected Areas in Communications 17 (8) (1999) 1333–1344.

[35] E. Royer, C. Toh, A review of current routing protocols for ad hoc mobile wireless networks, IEEE Personal Communications 6 (4) (1999) 46–55.

[36] R. Shah, J.M. Rabaey, Energy-aware routing for low-energy ad-hoc sensor networks, in: Proc. of the Wireless Communications and Networking Conference, Orlando, March 2002, pp. 812–817.

[37] I. Stojmenovic, Location updates for efficient routing in ad hoc networks, in: Handbook on Wireless Networks and Mobile Computing, 2002, pp. 451–471.

[38] I. Stojmenovic, S. Datta, Power and cost aware localized routing with guaranteed delivery in unit graph based ad hoc networks, Wireless Communications and Mobile Computing 4 (2) (2004) 175–188.

[39] I. Stojmenovic, X. Lin, Power aware localized routing in ad hoc networks, IEEE Transactions on Parallel and Distributed Systems 12 (11) (2001) 1122–1133.

[40] H. Takagi, L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Transactions on Communications 32 (3) (1984) 246–257.

[41] C. Yu, B. Lee, H. Youn, Energy efficient routing protocols for mobile ad hoc networks, Wireless Communications and Mobile Computing Journal 3 (11) (2003) 959–973.

[42] B. Zhang, H. Mouftah, Energy-aware on-demand routing protocols for wireless ad hoc networks, Wireless Networks 12 (4) (2006) 481–494.

**A.E. Abdallah** obtained his B.Sc. in Computer Science (Yarmouk University 2000), M.Sc. in Computer Science University of Jordan 2003), Ph.D. in Computer Science (Concordia University 2008) Montreal-Canada. He is currently a Network Researcher at consulting private company in Montreal. His research interests include the Routing protocols for Ad Hoc Networks, Parallel and Distributed Systems, and image processing.

**Thomas Fevens** is an Associate Professor in the Department of Computer Science and Software Engineering of Concordia University, having joined in 2001. He obtained the degrees of B.Sc. (1990) Honours in Astrophysics, M.Sc. (1993) in Physics (General Relativity), M.Sc. (1994) in Computer Science (Numerical Analysis), and Ph.D. (1999) in Computer Science (Computational Geometry) from Queen's University in Kingston, Canada. Before joining Concordia University, he held positions as Adjunct Assistant Professor at Queen's University (1999–2000) and Postdoc at McGill University (2000–2001). His research includes the areas of Computational Geometry, Ad Hoc Networks, Computer Graphics and Gaming, and Medical Imaging.

**Jaroslav Opatrny** received his first degree in 1968 from Charles University in Prague, Czech Republic, and the Ph.D. in computer science from the University of Waterloo, Canada in 1975. From 1975 to 1977 he was an assistant professor in the Computer Science Department at the University of Alberta, Canada. In 1977, he joined the Computer Science Department at Concordia University, Montreal, Canada, where he is presently a professor. His main interests are interconnection networks, communications in networks, and design of algorithms.

**Ivan Stojmenovic** received Ph.D. degree in mathematics. He held positions in Serbia, Japan, USA, Canada, France, Mexico, Spain and UK (as Chair in Applied Computing at the University of Birmingham, UK), and is Full Professor the University of Ottawa, Canada. He published over 250 different papers, and edited four books on wireless, ad hoc and sensor networks and applied algorithms with Wiley/IEEE. He is currently editor of over dozen journals, and founder and editor-in-chief of three journals (Journal of Multiple-Valued Logic and Soft Computing, International Journal of Parallel, Emergent and Distributed Systems, and Ad Hoc & Sensor Networks, An International Journal). Stojmenovic is in the top 0.56% most cited authors in Computer Science (Citeseer, 2006). One of his articles was recognized as the Fast Breaking Paper, for October 2003 (as the only one for all of computer science), by Thomson ISI Essential Science Indicators. He is recipient of the Royal Society Research Merit Award, UK. He is recently elected to IEEE Fellow status (Communications Society, class 2008). He chaired and/or organized >30 workshops and conferences, and served in over 100 program committees since 2004. Among others, he was/is program co/vice-chair at IEEE PIMRC 2008, IEEE AINA-07, IEEE MASS-04 and -07, EUC-05 and -08, WONS-05, MSN-05 and -06, ISPA-05 and -07, founded workshop series at IEEE MASS, IEEE ICDCS and IEEE DCOSS, and Workshop Chair at ACM Mobicom/Mobihoc 2007 and ACM Mobihoc 2008. He presented over dozen tutorials.