Hierarchical Bipartition Routing for delivery guarantee in sparse wireless ad hoc sensor networks with obstacles

Daniel Gaußmann, Stefan Hoffmann, and Egon Wanke

Institute of Computer Science, Heinrich-Heine-Universität Düsseldorf, D-40225 Düsseldorf, Germany

Abstract. We introduce and evaluate a very simple landmark-based network partition technique called Hierarchical Bipartition Routing (HBR) to support routing with delivery guarantee in wireless ad hoc sensor networks. It is a simple routing protocol that can easily be combined with any other greedy routing algorithm to obtain delivery guarantee. The efficiency of HBR increases if the network is sparse and contains obstacles. The space necessary to store the additional routing information at a node $u$ is on average not larger than the size necessary to store the IDs of the neighbors of $u$. The amount of work to setup the complete data structure is on average proportional to flooding the entire network $\log_2(n)$ times, where $n$ is the total number of sensor nodes. We evaluate the performance of HBR in combination with two simple energy-aware geographic greedy routing algorithms based on physical coordinates and virtual coordinates, respectively. Our simulations show that the difference between using HBR and a weighted shortest path to escape a dead-end is only a few percent in typical cases.

1 Introduction and related work

A wireless ad hoc sensor network is a decentralized network not relying on a preexisting infrastructure in that the nodes operate on limited hardware (memory and energy) and can only interchange packets within a radio range. The number of deployed nodes could be very large. Wireless ad hoc sensor networks are receiving a lot of attention in recent years due to their potential applications in various areas such as monitoring, security and data gathering.

Routing is the process of sending a packet from one or more source nodes to one or more target nodes through intermediate nodes. Each node decides locally to which neighbor the packet is forwarded. The decision is determined by the routing algorithm based on the ID of the destination node, the local topology (and possibly geometry) of the network, extra information stored in each node about the routes (the routing tables) and information contained in the packet itself.

A routing protocol defines the rules for exchanging the information between nodes. In geographic routing protocols the decision to which neighbor the packet is sent is controlled by the position of the nodes and the distances between them. The position information to each node can be obtained either by devices such as GPS or Galileo (geographic coordinates) or by analyzing the network structure (virtual coordinates). Position awareness can often significantly improve the efficiency of routing. In [24] it is mentioned that protocols using position information for routing like MFR [30], COP [27], and GFG [14] are competitive alternatives to the classical routing protocols for wireless ad hoc networks as for example DSR [17], AODV [25], and OLSR [8]).

Most algorithms based on position awareness first try to deliver the packet using greedy techniques. For example, the simplest greedy routing technique [12] will forward the packet...
to the neighbor closest to the destination. Most Forward Routing (MFR) and Nearest with Forwarding Progress (NFP) consider the projected distance on the source-destination line. MFR tries to get closer to the destination by sending the packet to the neighbor with maximum projected distance, while NFP suggests to adjust the transmission power by sending the packet to the neighbor with the smallest projected distance.

Greedy routing algorithms can easily be extended by taking into account power consumption. Power aware greedy routing in most cases tries to minimize the ratio of the energy consumed by a transmission to the progress made. The progress is the distance reduction towards the destination. This cost over progress (COP) power-aware framework is first introduced in. If the cost is equal for all connections, we obtain the simple greedy algorithm as already discussed above. If the cost of a connection is proportional to the distance between the nodes, the resulting routing is similar to compass routing.

In general, greedy routing algorithms do not guarantee packet delivery. A packet can be trapped in a local minimum where the algorithm will fail to find a next neighbor. The probability of reaching a so-called dead-end increases if the network is less dense or if the network contains obstacles where no nodes can be placed and/or connections are truncated by obstacles.

There are several attempts to obtain delivery guarantee for greedy routing algorithms. The authors of propose face routing, which guarantees delivery in two-dimensional unit disk graphs (UDG). Face routing is applied to a planar sub-network obtained by considering the Gabriel Graph, the Relative Neighborhood Graph, or the Morelia Graph. In a greedy-face-greedy (GFG) approach is considered, where greedy routing is based on COP as in and face routing is similar to the one in. Energy-aware routing is also proposed in LEARN, SPFSP, End-to-End (EtE), and EEGR.

Landmark-based routing algorithms like VCap, JUMPS, GLIDER, VCost, and BVR use virtual coordinates computed from the distances to specific nodes called landmarks, anchors, or beacons. In the first phase, a global and distributed election mechanism elects a set of nodes acting as landmarks. Then the landmarks flood the entire network or only parts of the network such that every node can compute its virtual coordinate depending on the distances to the landmarks. The virtual coordinates can then be used to route a message greedily through the network. Packet delivery is also not guaranteed if different nodes have the same virtual coordinates.

An alternative way for delivery guarantee can be obtained by hierarchical addressing, see for example. Tsuchiya solves this problem by allowing nodes to self configure their addresses. The protocol uses a hierarchical set of landmark nodes that periodically send scoped route discovery messages. A node’s address is the concatenation of its closest landmark at each level in the hierarchy. The overhead of route setup can be reduced to $O(n \log n)$ and nodes only hold state for their immediate neighbors and their next hop to
each landmark. However, this requires a protocol that creates and maintains this hierarchy of landmarks and appropriately tunes the landmark scopes. Recent proposals adopting this approach have been fairly complex [20] in contrast to our design goal of configuration simplicity, see also [16] for an overview.

A second alternative to obtain delivery guarantee is clustering which is proposed by various researchers as for example in [7] and [19]. A closely related approach is the construction of connected dominating sets as routing backbones [34].

In this paper, we introduce and evaluate a very simple landmark-based network partition technique called Hierarchical Bipartition Routing (HBR) to support routing with delivery guarantee in wireless ad hoc sensor networks. It is a simple routing protocol that can easily be combined with any other greedy routing algorithm to obtain delivery guarantee. The efficiency of HBR increases if the network is sparse and contains obstacles. These are exactly the networks where greedy algorithms will fail with high probability. The hierarchical bipartition of the network is performed on a landmark-based data structure setup in a pre-processing phase.

The space necessary to store the additional routing information at a node $u$ is approximately $\log_2(n) \cdot \log_2(\deg(u))$ bits on average, where $n$ is the total number of sensor nodes in the network and $\deg(u)$ is the number of neighbors of $u$. This is in general not larger than the size necessary to store the IDs of all neighbors of $u$. The amount of work to setup the complete data structure is on average proportional to flooding the entire network $\log_2(n)$ times. The sizes of the virtual addresses are on average only a few bits larger than the sizes of the IDs.

We evaluate the performance of HBR in combination with two simple energy-aware geographic greedy routing algorithms based on physical coordinates and virtual coordinates, respectively. Our simulations show that the difference between using HBR and a weighted shortest path to escape a dead-end is only a few percent in typical cases.

This paper is organized as follows. In the next section, we define HBR. After that in Section 3, we describe two simple greedy routing protocols and how they can escape from a dead-end using HBR or a weighted shortest path. In section 4, we evaluate the performance of HBR. Conclusions are given in Section 5.

2 Hierarchical Bipartition Routing (HBR)

A network is modeled as an undirected graph $G = (V, E)$, where $V$ is the set of nodes (sensor nodes) and $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ is the set of undirected edges (connections) between sensor nodes. A connection $e = \{u, v\}$ may have a positive weight $\omega(e) \in \mathbb{R}_+$, also denoted by $\omega(u, v)$ or $\omega(v, u)$. In general, the weight represents the amount of energy necessary to reach the neighbor node. A path $p = u_1, \ldots, u_k$ is a non-empty sequence of nodes $u_i \in V, 1 \leq i \leq k$, such that $\{u_i, u_{i+1}\} \in E, 1 \leq i < k$, is a connection between $u_i$ and $u_{i+1}$. The weight of $p$ is

$$\omega(p) = \sum_{i=1}^{k-1} \omega(u_i, u_{i+1}).$$
Path $p$ is a shortest path between $u$ and $v$, if there is no path $p'$ between $u$ and $v$ with $\omega(p') < \omega(p)$.

The $\omega$-distance $d^\omega(u, v)$ between two nodes is the weight of a shortest path between $u$ and $v$.

The hop distance $d^h(u, v)$ between two nodes is the weight of a shortest path between $u$ and $v$ for the case that all connections $\{u, v\}$ have weight $\omega(u, v) = 1$.

To analyze the performance of HBR, we assume that every node $u \in V$ has a physical geographic position $(u_x, u_y) \in \mathbb{R}^2$ in the plane defined by a two-dimensional real vector. These positions are only used by the geographic greedy routing protocols, and not by the hierarchical bipartition technique. The euclidean distance between two nodes $u$ and $v$ is

$$d_e(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}.$$  

Note that the $\omega$-distance $d^\omega(u, v)$ and the hop distance $d^h(u, v)$ are defined by the network structure, whereas the euclidean distance is defined by the physical geographic positions of the nodes.

A network $G = (V, E)$ is connected if there is a path between every pair of nodes. Network $G' = (V', E')$ is a sub-network of $G$ if $V \subseteq V$ and $E' \subseteq E$, it is an induced sub-network of $G$ if $V' \subseteq V$ and $E' = \{\{u, v\} | \{u, v\} \in E, u, v \in V'\}$.

The nodes are assumed to be static. Each node is assumed to have a unique ID which is mainly used to break ties. The unique ID is also necessary to specify the target node of the packet, if we do not want to give every node a unique virtual address.

### 2.1 Initialization

The necessary data structure is built in several phases. In the first phase, an arbitrary node $w$ and two landmark nodes $x_0, x_1$ are selected. Landmark node $x_0$ is one of the nodes that has maximum $\omega$-distance to $w$, and landmark node $x_1$ is one of the nodes that has maximum $\omega$-distance to $x_0$. Every node $u$ with

$$d^\omega(u, x_0) \leq d^\omega(u, x_1) \quad \text{gets virtual address } 0,$$

and every node $u$ with

$$d^\omega(u, x_0) > d^\omega(u, x_1) \quad \text{gets virtual address } 1.$$

Let $G_0$ be the network induced by the nodes with virtual address 0 and let $G_1$ be the network induced by the nodes with virtual address 1.

**Lemma 1.** If network $G$ is connected, then $G_0$ and $G_1$ are connected.

**Proof.** Let $v$ be a neighbor of a node $u$ on a shortest path between $u$ and $x_0$, then

$$\omega(u, v) + d^\omega(v, x_0) = d^\omega(u, x_0).$$
If the virtual address of $u$ is 0, then

$$d^\omega(u, x_0) \leq d^\omega(u, x_1).$$

Since

$$d^\omega(u, x_1) \leq \omega(u, v) + d^\omega(w, x_1),$$

we get

$$d^\omega(v, x_0) \leq d^\omega(v, x_1).$$

That is, all nodes $v$ on a shortest path between $u$ and $x_0$ have virtual address 0 and thus $G_0$ is connected. An analogous argumentation shows that $G_1$ is connected. $\square$

In the next phase, we select for every virtual address $\alpha \in \{0, 1\}$ two landmark nodes $x_{\alpha, 0}$, $x_{\alpha, 1}$ from the connected sub-network $G_\alpha$. Here $\alpha \cdot 0$ and $\alpha \cdot 1$ is the extension of $\alpha$ by symbol 0 or 1, respectively. Landmark node $x_{\alpha, 0}$ is one of the nodes of network $G_\alpha$ that has maximum $\omega$-distance to $x_\alpha$ in sub-network $G_\alpha$, and landmark node $x_{\alpha, 1}$ is one of the nodes of network $G_\alpha$ that has maximum $\omega$-distance to $x_{\alpha, 0}$ in sub-network $G_\alpha$. A node $u$ of $G_\alpha$ whose $\omega$-distance to $x_{\alpha, 0}$ is less than or equal to the $\omega$-distance between $u$ and $x_{\alpha, 1}$ gets virtual address $\alpha \cdot 0$. It gets virtual address $\alpha \cdot 1$, if the distance between $u$ and $x_{\alpha, 0}$ is greater than the distance between $u$ and $x_{\alpha, 1}$.

The bipartition of every $G_\alpha$ into two further sub-networks $G_{\alpha, 0}$ and $G_{\alpha, 1}$ can be continued with the new virtual addresses $\alpha$ until all the created sub-networks consist of only one single node. In this case, the nodes of the network are uniquely identified by the virtual addresses. An inductive application of Lemma 1 shows that all the sub-networks $G_\alpha$ are connected.

### 2.2 Distributed address computation

The hierarchical bipartition can easily be computed by a distributed algorithm. Once the network is deployed, an arbitrarily selected node $w$ starts flooding the network. The message carries a weight initialized to zero. The weight is increased by every forwarding node. If node $u$ sends a message to a neighbor $v$, then $u$ increases the weight by $\omega(u, v)$. If a node receives more than one message, it will store and forward only the one with the smaller weight. If a node does not receive a new message for a while, the current weight represents the $\omega$-distance to $w$. To be sure that the flooding is finished, the node has to wait for a time longer than the time required to propagate a message through the network. To reduce the overhead during the distance computation, the messages should be sent to the neighbors $v_1, \ldots, v_m$ in ascending order with respect to the costs at the edges, that is, $\omega(u, v_1) \leq \ldots \leq \omega(u, v_m)$.

The election of the landmark nodes for the bipartition of the network can be done by the following simple protocol. Assume we want to determine a unique node $u$ with maximum $\omega$-distance to some other node $w$. Then all nodes with a maximum $\omega$-distance to $w$ in its two-hop neighborhood (in case of parity the nodes with maximum IDs) start sending a message back to $w$. (The route back to $w$ can be stored during the update process of
the \( \omega \)-distance to \( w \).) Node \( w \) receives all these messages and can select the node \( u \) with maximum distance to \( w \). It has to wait for a while such that no further messages will arrive. Then it sends a message back to the winner, the node \( u \) with maximum distance to \( w \).

### 2.3 Routing protocol

The routing protocol is quite simple and straightforward. Assume a packet should be sent from a source node \( s \) to a target node \( t \). If the virtual address of \( s \) starts with symbol 0 and the virtual address of \( t \) starts with symbol 1, then \( s \) is in sub-network \( G_0 \) and \( t \) is in sub-network \( G_1 \). In this case, the packet is sent step by step to a neighbor whose distance to landmark node \( x_1 \) is minimum until it reaches a node in \( G_1 \). Then it is routed within the connected sub-network \( G_1 \) using the second symbol of the virtual addresses, and so on.

More generally, let \( \alpha \cdot d_u \cdot \alpha_u \) be the virtual address of the current node \( u \) and \( \alpha \cdot d_t \cdot \alpha_t \) be the virtual address of the destination \( t \) such that \( \alpha, \alpha_u, \alpha_t \in \{0, 1\}^* \), \( d_u, d_t \in \{0, 1\} \), and \( d_u \neq d_t \). That is, the symbols left to \( d_u \) and left to \( d_t \) are equal in both virtual addresses. If \( d_u = 0 \) and \( d_t = 1 \), the packet is sent greedily towards landmark node \( x_{\alpha \cdot 1} \), if \( d_u = 1 \) and \( d_t = 0 \), the packet is sent greedily towards landmark node \( x_{\alpha \cdot 0} \). The packet does not leave the connected sub-network \( G_\alpha \). An inductive argumentation proves that HBR guarantees delivery.

**Corollary 1.** A packet sent with HBR always reaches its destination.

### 2.4 Address size

The sizes \( |\alpha| \) of the virtual addresses \( \alpha \) depend on the number of partitions necessary to obtain single node graphs. In this case, the virtual addresses are unique and can be used for the identification of the nodes. If the weights of all edges are equal, then a worst case for the address length is a complete network, that is, a network where every node is connected to all other nodes. Then the number of bipartitions and thus the address length is \( n - 1 \).

To avoid these worst-case situations, we can stop the bipartition process when all nodes of a sub-network \( G_\alpha \) have hop distance \( \leq 1 \) to landmark node \( x_\alpha \). This will considerably reduce the address size. Since now the nodes do not have unique virtual addresses, it is necessary to include the target ID in the packets.

The routing protocol can be extended as follows: Assume the packet reaches a node \( u \) that has virtual target address \( \alpha \) but is not the target node. If a neighbor \( v \) of \( u \) is the target node, the packet can be sent to node \( v \). Otherwise, the packet can be sent to \( x_\alpha \) and from \( x_\alpha \) to the target node. This is always possible, because all nodes with virtual address \( \alpha \) are neighbors of \( x_\alpha \). The decision to stop the bipartition is very easy to implement, because the nodes of \( G_\alpha \) that are candidates for flooding only have to check their list of neighbors.

If the weights of the connections are not all equal, but depend on the distances of the connections, then it is in general not necessary to abort the bipartition process. In practical cases, the different lengths of the connections yields to a partition of the network into two almost equally sized sub-networks. In this case, the virtual addresses are unique and it will not be necessary to use the original IDs.
2.5 Storage size

A node has to store its ID, its virtual address, a routing table, and temporarily during the initialization phase some $\omega$-distances to landmark nodes and some source IDs. For the partition of sub-network $G_\alpha$ into $G_{\alpha \cdot 0}$ and $G_{\alpha \cdot 1}$, we only need the $\omega$-distances to $x_{\alpha \cdot 0}$ and $x_{\alpha \cdot 1}$. When the new virtual addresses $\alpha \cdot 0$ and $\alpha \cdot 1$ are assigned, it is no longer necessary to store these $\omega$-distances, and the IDs of $x_{\alpha \cdot 0}$ and $x_{\alpha \cdot 1}$. For routing a packet it is sufficient to know for every position $i$, $1 \leq i \leq |\alpha|$, the neighbor to which the packet has to be sent if the own address $\alpha$ and the target address of the packet are equal at the first $i-1$ positions and differ at position $i$. If a node $u$ with virtual address $\alpha$ has $\deg(u)$ neighbors, then the size of the additional routing information is only $|\alpha| \cdot \log_2 \deg(u)$ bits.

2.6 Worst case behavior

From a theoretical point of view, the weight of a path routed by HBR can be arbitrarily larger than a shortest path between the source and target node. Figure 1 shows a simple example. The virtual addresses of the black and white nodes start with symbol 0 and 1, respectively. A shortest path between the source node $s$ and the target node $t$ has weight $2 \cdot a$. HBR routing will send the packet from $s$ to $v$ and then via $x_1$ to $t$. The weight of this path is $m \cdot a$, where $m$ can be arbitrary large. However, this is not a typical case for randomly generated networks.

![Fig. 1. An unrealistic worst case for the stretch factor of HBR](image)

In the worst case, the size of the virtual addresses $\alpha$ can reach the number $n$ of nodes. This is for example the case if the network is complete and all edges have the same weight, or if the network is a path and the connections have exponentially increasing weights $1, 2, 4, 8, 16, 32, \ldots$. However, if the number of nodes reachable with increasing distance is for most nodes approximately the same, the size of the virtual addresses is at most $\log_2 n$. This is also confirmed by our experimental evaluations of randomly generated networks.

3 Two greedy routing protocols

We mainly want to use HBR to guarantee delivery for greedy routing protocols. For this reason, we combine HBR with geographic greedy routing based on physical and virtual coordinates.
3.1 Physical coordinates

As already mentioned in the introduction, the simplest energy-aware geographic greedy routing protocol sends the packet to a neighbor for which the ratio of cost over progress is minimum. This cost over progress (COP) power-aware framework is introduced in [21]. If we consider physical coordinates, the packet is sent from node \( u \) to a neighbor \( v \) of \( u \) for which

\[ d^e(v, t) < d^e(u, t) \]

and

\[ \omega(u, v) \frac{d^e(u, t) - d^e(v, t)}{d^e(u, t)} \]

is minimum. If \( \omega(u, v) = a, a > 0 \), for all connections \( \{u, v\} \), we obtain the simple geographic greedy routing that selects a neighbor closest to the destination [12]. For \( \omega(u, v) = d^e(u, v) \), the routing is similar to compass routing [18]. If the angle between \( (u, v) \) and \( (u, t) \) is \( \beta \) and the distance to the target node tends to infinity, the ratio of cost over progress tends to \( \frac{1}{\cos(\beta)} \). If \( \omega(u, v) \) is defined by the commonly used energy function

\[ a + b \cdot d^e(u, v)^e \]

\( a, b > 0, c \geq 2 \), then an optimal routing tries to use equidistant steps towards the target node \( t \). The best progress (also called the characteristic distance) is

\[ d^* = \sqrt[\alpha]{\frac{a}{b \cdot (c - 1)}} \]

see also [29]. The ratio of cost over progress has its minimum at the position with distance \( d^* \) from \( u \) in the direction to the destination.

3.2 Virtual coordinates

Our second greedy routing protocol is based on virtual coordinates which we will define by four landmark nodes denoted by \( A, B, C, \) and \( D \). These four landmark nodes are selected similarly as in VCap (virtual coordinate assignment protocol) from [5]. The first landmark node \( A \) is one of the nodes with maximum \( \omega \)-distance to an arbitrary node \( w \). The second
landmark node $B$ is one of the nodes with maximum $\omega$-distance to $A$. The third landmark node $C$ is one of the nodes for which

$$d^\omega(C, A) + d^\omega(C, B) - 2 \cdot |d^\omega(C, A) - d^\omega(C, B)|$$

is maximum. And finally, the fourth landmark node $D$ is one of the nodes for which

$$d^\omega(D, C) - |d^\omega(D, A) - d^\omega(D, B)|$$

is maximum. Since will consider energy efficient routing, we use the $\omega$-distances instead of hop distances [5] for the computation of the virtual addresses.

In case of parity, the node with larger ID is chosen. The four landmark nodes $A, B, C, D$ define for every node $u$ a 4-tuple

$$(d^\omega(u, A), d^\omega(u, B), d^\omega(u, C), d^\omega(u, D)).$$

Our landmark-based routing protocol sends the packet to a neighbor $v$ for which the ratio cost over progress is minimum. The progress $d'(u, t) - d'(v, t)$ is defined by distance function

$$d'(u, v) = \sqrt{(d^\omega(u, A) - d^\omega(v, A))^2 + (d^\omega(u, B) - d^\omega(v, B))^2 + (d^\omega(u, C) - d^\omega(v, C))^2 + (d^\omega(u, D) - d^\omega(v, D))^2}.$$  

3.3 Dead-end handling

Both greedy routing protocols can reach a so-called dead-end, i.e., a node $u$ that has no neighbor $v$ closer to the destination than $u$. If a dead-end is reached, the packet is either sent along a shortest path or by HBR towards the destination node. In both cases the weight function $\omega$ is applied. The packet is sent hop by hop until a node is reached whose distance to the destination is less than the distance from the last dead-end node to the destination. Then the original greedy routing is continued.

It is obvious that a shortest-path routing is not possible in practice. We use shortest-path routing only to get a comparison with HBR under the assumption that following a shortest path is a good idea to get out of a dead-end.

The two geographic routing variants based on physical coordinates are denoted by GEO$^{SP}$ and GEO$^{HBR}$, the two variants based on virtual coordinates are denoted by LMR$^{SP}$ or LMR$^{HBR}$, depending on whether the dead-end problem is cleared with the help of a shortest path or by HBR, respectively.

4 Analysis

The analysis of HBR is done by randomly generated networks and randomly selected source and target nodes. The test environment and the obtained evaluation results is explained in the next subsections.
4.1 Experimental environment

Our networks have a size of $1000m \times 1000m$. The radio range is fixed at $50m$, the node density $\delta$ varies between $0.5 \cdot 10^{-3}$ and $9.2 \cdot 10^{-3}$ nodes per $m^2$, which corresponds to an average node degree between 4 and 72. If one of the randomly created networks is disconnected, we use the largest connected component, if its size is at least $\frac{2}{3}$ of the size of the complete network.

Networks with holes or obstacles are created with the help of black/white-masks. If the randomly selected position of a node hits a white-entry of the mask, the node is omitted. We do not try to find another position for this node. The masks we use for our evaluations are shown in Figure 3, 4, and 5.

![Fig. 3. Left: Lakes-mask, lat. 51.19°, lon. 6.37°; Right: Routing in a network generated with lakes-mask and density $\delta = 1.0 \cdot 10^{-3}$](image)

The lakes-mask of Figure 3 (latitude 51.19°, longitude 6.37°) represents wet areas where sensors are lost during the dispersion process. The streets-mask of Figure 4 (latitude 40.70°, longitude −73.93°) represents an area where the sensor nodes are assumed to be dispersed by vehicles driving along streets. The buildings-mask of Figure 5 (latitude 52.50°, longitude 13.35°) represents an example of a metropolitan area. Here we additionally remove all connections between sensors that can not see each other, because there is a building in between.

We use the energy function

$$\omega(u, v) = 400 + d^e(u, v)^2$$

(1)

for every connection $\{u, v\} \in E$, such that the characteristic distance is $d^* = 20 m$.

Each of the Figures 3, 4, and 5 shows to the right two routing paths. The start node is encircled green, the destination node is encircled blue. The nodes traversed by HBR
are colored black. The nodes traversed by GEO\textsuperscript{SP} are colored yellow. The dead-end nodes of GEO\textsuperscript{SP} are colored red. The small light green (light red) nodes have a virtual address starting with 0 (with 1, respectively).

For every node density $\delta$ between $0.5 \cdot 10^{-3}$ and $9.2 \cdot 10^{-3}$ in steps defined by factor 1.2, we randomly create 1000 networks. For every network, we randomly selected 1000 source nodes and 1000 target nodes. Let $C$(HBR) and $C$(SP) be the sum of the costs to route
from the 1000 source nodes to the corresponding 1000 target nodes in all 1000 networks using HBR and SP, respectively. The cost of a route is the sum of the weights of the used connections. The overhead of HBR is defined by

$$\frac{C(\text{HBR}) - C(\text{SP})}{C(\text{SP})}.$$ 

It is defined in the same way for the other routing protocols $\text{LMR}^{\text{SP}}, \text{LMR}^{\text{HBR}}, \text{GEO}^{\text{SP}},$ and $\text{GEO}^{\text{HBR}}$.

### 4.2 Evaluations

The tables 1, 2, 3, and 4 show the average overhead in percent as a function of the node density $\delta$. The overhead entries are colored continuously between green (0%) and red (50%). We think that it is more valuable to present the main results in a table than in a graphical illustration, because it is easy to create a graphical view from the values of the table, but not vice versa.

Table 1 considers networks where the sensor nodes are uniformly distributed. If the node density is greater than or equal to $1.0 \cdot 10^{-3}$, the two greedy routing algorithms GEO$^{\text{SP}}$ and GEO$^{\text{HBR}}$ on physical coordinates have less overhead than the greedy routing algorithms LMR$^{\text{SP}}$ and LMR$^{\text{HBR}}$ on virtual coordinates. For less dense networks ($\delta < 1.0 \cdot 10^{-3}$), HBR has even less overhead than GEO$^{\text{SP}}, \text{GEO}^{\text{HBR}}, \text{LMR}^{\text{SP}},$ and LMR$^{\text{HBR}}$. The difference between a shortest path dead-end handling and a HBR dead-end handling, i.e., the difference between GEO$^{\text{SP}}$ and GEO$^{\text{HBR}}$ and between LMR$^{\text{SP}}$ and LMR$^{\text{HBR}}$, is only a few percent. This holds for greedy routing with physical coordinates as well as for greedy routing with virtual coordinates.

The average address lengths and the average number $\gamma$ of routes that have at least one dead-end are shown in Table 1, 2, 3, and 4 to the right. These tables show that the sizes of the virtual addresses are only a few bits larger than the sizes of the IDs, which are at least $\lceil \log_2 n \rceil$.

HBR seems to be well suited for resolving the dead-end problem. The advantage of HBR is the very good performance in particular for sparse networks and for networks with obstacles. This show Table 2, 3, and 4.

### 4.3 Coarsening the edge weights

During the initialization phase, the nodes have to store and manage distance values to landmark nodes. The distances are built by sums of edge weights. Our weights of the connections defined in equation 1 can simply be coarsed to only $k$ different values by

$$\omega_k(u,v) = \left\lceil \frac{k \cdot 400 + d^2(u,v)}{400 + r^2} \right\rceil.$$ 

(2)

If we use weight function $\omega_k$ instead of $\omega$, the overhead of HBR will increase, depending on the number $k$ of different weights for the connections. In Table 5, it is shown how the
becomes. However, if the network is dense, the greedy algorithms reach a dead-end less by a few percent. The table also shows that the denser the network is, the larger the overhead increases when the number of different weights decreases. If we have the same weight for all connections, the algorithm considers hop distances. Table 5 to the left shows that 8 different weights for the connections will decrease the overhead of HBR only by a few percent. The table also shows that the denser the network is, the larger the overhead becomes. However, if the network is dense, the greedy algorithms reach a dead-end less
Table 3. Left: Average overhead in percent as a function of density $\delta$ with streets-mask; Right: Same as Table 1 for streets-mask

| $\delta$ | HBR | LMR$^{st}$ | LMR$^{hfr}$ | GEO$^{st}$ | GEO$^{hfr}$ |
|---------|-----|------------|------------|-----------|-----------|
| 1.2     | 4.08| 21.59      | 23.23      | 27.53     | 29.35     |
| 1.5     | 7.57| 23.27      | 26.50      | 28.56     | 32.14     |
| 1.8     | 12.87| 22.53    | 27.46      | 27.18     | 30.38     |
| 2.1     | 16.70| 18.36    | 22.89      | 22.72     | 28.67     |
| 2.6     | 17.59| 14.61    | 17.67      | 17.82     | 22.09     |
| 3.1     | 17.81| 12.62    | 14.58      | 14.23     | 16.87     |
| 3.7     | 17.95| 11.67    | 12.96      | 11.87     | 13.38     |
| 4.5     | 18.03| 11.09    | 11.98      | 10.42     | 11.34     |
| 5.3     | 18.14| 10.76    | 11.42      | 9.62      | 10.19     |
| 6.4     | 18.19| 10.40    | 10.87      | 9.06      | 9.46      |
| 7.7     | 18.26| 10.18    | 10.53      | 8.63      | 8.94      |
| 9.2     | 18.18| 9.92     | 10.20      | 8.26      | 8.52      |

Table 4. Left: Average overhead in percent as a function of density $\delta$ with buildings-mask and only visible connections; Right: Same as Table 1 for buildings-mask and only visible connections

| $\delta$ | HBR | LMR$^{st}$ | LMR$^{hfr}$ | GEO$^{st}$ | GEO$^{hfr}$ |
|---------|-----|------------|------------|-----------|-----------|
| 1.8     | 2.95| 24.06      | 25.55      | 27.79     | 29.29     |
| 2.1     | 4.70| 25.40      | 27.76      | 29.37     | 32.01     |
| 2.6     | 7.56| 26.45      | 30.27      | 31.14     | 35.71     |
| 3.1     | 11.90| 27.01    | 32.89      | 32.67     | 40.11     |
| 3.7     | 16.40| 27.03    | 34.64      | 33.05     | 43.29     |
| 4.5     | 19.10| 26.31    | 34.17      | 32.49     | 43.86     |
| 5.3     | 20.71| 25.60    | 32.98      | 31.51     | 43.04     |
| 6.4     | 21.20| 24.89    | 31.38      | 30.40     | 41.42     |
| 7.7     | 21.31| 24.71    | 30.87      | 29.97     | 40.36     |
| 9.2     | 21.37| 24.59    | 30.57      | 29.61     | 39.55     |

often. If HBR is used only to guarantee delivery for greedy routing protocols, the overhead will increase only minimal.

4.4 Energy to setup the data structure

During the setup of the initial data structure, the network is flooded several times. To determine the distances to the arbitrarily chosen start node $w$, the complete network is flooded. This is also the case for determining the distances to the landmark nodes $x_0$ and $x_1$. After that the network is flooded only partially, because the distances to landmark nodes $x_{00}$ and $x_{01}$ ($x_{10}$ and $x_{11}$) are only relevant for the nodes having a virtual address starting with 0 (with 1, respectively). In general, the total amount of energy to determine all distances to the $2 \cdot 2^k$ landmark nodes with the same address length $k$ is less than the amount of energy to flood the network two times. Our experimental analyses show that the amount of energy necessary for the next bipartition decreases for increasing address lengths.

Table 5 to the right shows the number of transmissions to build the initial data structure. This is not the average over 1000 networks but only one typical example. The network
is created with a node density of $2.5 \cdot 10^{-3}$ nodes per $m^2$ without any masks. It has 2500 nodes and 23462 edges. The first line shows the total number of transmissions for flooding the network from the arbitrarily chosen node $w$. The second line shows the total number of transmissions for flooding the network from the two landmark nodes $x_0$ and $x_1$. The third line shows the total number of transmissions for flooding two sub-networks from four landmark nodes $x_{00}$, $x_{01}$, $x_{10}$, and $x_{11}$, and so on.

| $\delta$ | $\omega_1$ | $\omega_16$ | $\omega_4$ | $\omega_2$ | $\omega_1$ |
|---------|-------------|-------------|------------|------------|------------|
| 0.5     | 2.94        | 3.02        | 3.25       | 4.05       | 5.08       | 7.44       |
| 0.6     | 5.83        | 5.89        | 6.22       | 7.32       | 8.67       | 11.40      |
| 0.7     | 11.52       | 11.77       | 12.22      | 13.68      | 15.26      | 19.82      |
| 0.9     | 16.87       | 17.06       | 17.62      | 19.02      | 21.79      | 27.62      |
| 1.0     | 17.92       | 18.43       | 19.12      | 21.37      | 23.95      | 31.33      |
| 1.2     | 17.96       | 18.47       | 19.43      | 22.37      | 25.16      | 34.49      |
| 1.5     | 18.02       | 18.62       | 19.83      | 23.21      | 26.06      | 37.34      |
| 1.8     | 18.05       | 18.88       | 20.27      | 24.12      | 26.85      | 40.39      |
| 2.1     | 18.09       | 19.17       | 20.76      | 25.01      | 27.37      | 43.21      |
| 2.6     | 18.21       | 19.49       | 21.26      | 25.93      | 27.71      | 45.78      |
| 3.1     | 18.33       | 19.86       | 21.78      | 26.66      | 27.87      | 48.22      |
| 3.7     | 18.59       | 20.26       | 22.22      | 27.29      | 27.90      | 50.41      |
| 4.5     | 18.82       | 20.67       | 22.54      | 27.76      | 27.92      | 52.30      |
| 5.3     | 19.08       | 21.06       | 23.04      | 28.09      | 28.02      | 53.97      |
| 6.4     | 19.32       | 21.43       | 23.30      | 28.31      | 28.09      | 55.40      |
| 7.7     | 19.58       | 21.75       | 23.53      | 28.45      | 28.26      | 56.80      |
| 9.2     | 19.82       | 22.02       | 23.67      | 28.46      | 28.54      | 57.67      |

| $|\alpha|$ | transmissions | sub-networks |
|--------|--------------|--------------|
| 1      | 137658       | 1            |
| 2      | 164885       | 2            |
| 3      | 149594       | 4            |
| 4      | 123651       | 8            |
| 5      | 115242       | 16           |
| 6      | 109842       | 32           |
| 7      | 108777       | 64           |
| 8      | 101131       | 128          |
| 9      | 99712        | 255          |
| 10     | 95281        | 466          |
| 11     | 84330        | 699          |
| 12     | 55758        | 561          |
| 13     | 21186        | 234          |
| 14     | 4668         | 54           |
| 15     | 366          | 5            |

Table 5. Left: Average overhead for coarsened edge weights, Right: Number of transmissions during the preprocessing phase for a network with 2500 nodes and 23462 edges

5 Conclusions

Greedy algorithms for routing in wireless ad hoc sensor networks are easy to implement. They are very effective but can unfortunately reach an impasse. In this paper, we have introduced and analyzed a very simple hierarchical bipartition technique for wireless ad hoc sensor networks. Every node gets a unique virtual address that can be used to route through the network with delivery guarantee. To keep the advantages of greedy routing, we suggest to use HBR for finding the way out of a dead-end. This is especially very interesting if the network is sparse or contains obstacles. In these cases, the probability to reach a dead-end is very high. Our experimental evaluations even show the following: If more than 50% of the routes reach at least one dead-end, then the performance of stand-alone HBR is in general better than the performance of geographic greedy routing with a shortest path dead-end handling.

The main assets and drawbacks of HBR are the following:
Assets:

1. packet delivery guaranty
2. no packet overhead, because the unique virtual addresses are of size $\log n$ on average
3. small routing tables
4. no geographic coordinates necessary

Drawbacks:

1. works only for static network structures and is vulnerable to even small changes of topology
2. relative time-consuming and energy-consuming set-up phase
3. unbalanced partitioning process results in large address lengths
4. unbounded worst-case stretch factor

References

1. INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 13-17 March 2005, Miami, FL, USA. IEEE, 2005.
2. F Benbadis, J.-J. Puig, M. Dias de Amorim, C. Chaudet, T. Friedman, and D. Simplot-Ryl. Jumps: Enhancing hop-count positioning in sensor networks using multiple coordinates. CoRR, abs/cs/0604105, 2006.
3. P. Boone, E. Chávez, L. Gieizky, E. Kranakis, J. Opatrny, G. Salazar, and J. Urrutia. Morelia test: Improving the efficiency of the gabriel test and face routing in ad-hoc networks. In R. Kravlic and O. Šykora, editors, SIROCCO, volume 3104 of Lecture Notes in Computer Science, pages 23–34. Springer, 2004.
4. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks, 7(6):609–616, 2001.
5. A. Caruso, S. Chessa, S. De, and A. Urpi. Gps free coordinate assignment and routing in wireless sensor networks. In INFOCOM [1], pages 150–160.
6. E. Chávez, N. Mitton, and H. Tejeda. Routing in wireless networks with position trees. In E. Kranakis and J. Opatrny, editors, ADHOC-NOW, volume 4686 of Lecture Notes in Computer Science, pages 32–45. Springer, 2007.
7. C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks. In IEEE Singapore International Conference on Networks, pages 197–211, 1997.
8. T. Clausen, P. Jacquet, A. Laouiti, P. Mühlthaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol (OLSR). RFC 3626, 2003.
9. E. Elhafsi, N. Mitton, and D. Simplot-Ryl. Cost over progress based energy efficient routing over virtual coordinates in wireless sensor networks. In WOWMOM, pages 1–6. IEEE, 2007.
10. E.H. Elhafsi, N. Mitton, and D. Simplot-Ryl. End-to-end energy efficient geographic path discovery with guaranteed delivery in ad hoc and sensor networks. In PIMRC, pages 1–5. IEEE, 2008.
11. Q. Fang, J. Gao, L.J. Guibas, V. de Silva, and L. Zhang. GLIDER: gradient landmark-based distributed routing for sensor networks. In INFOCOM [1], pages 339–350.
12. G.G. Finn. Routing and addressing problems in large metropolitan scale internetworks. Technical Report ISU/RR-87-180, Internetworks,ISI, 1987.
13. R. Fonseca, S. Ratnasamy, J. Zhao, C.T. Ee, D.E. Culler, S. Shenker, and I. Stoica. Beacon vector routing: Scalable point-to-point routing in wireless sensor networks. In NSDI. USENIX, 2005.
14. H. Frey and I. Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In M. Gerla, C. Petrioli, and R. Ramjee, editors, MOBICOM, pages 390–401. ACM, 2006.
15. T.C. Hou and V. Li. Transmission range control in multihop packet radio networks. In IEEE Transactions on Communications, volume 34, pages 38–44. IEEE, IEEE, January 1986.
16. K. Iwanicki and M. van Steen. On hierarchical routing in wireless sensor networks. In IPSN, pages 133–144. ACM, 2009.
17. D.B. Johnson, D.A. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In C.E. Perkins, editor, Ad Hoc Networking, pages 139–172. Addison-Wesley, 2001.
18. E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In CCCG, pages 51–54, 1999.
19. P. Krishna, M. Chatterjee, N.H. Vaidya, and D.K. Pradhan. A cluster-based approach for routing in ad-hoc networks. In *Symposium on Mobile and Location-Independent Computing*, pages 1–10. USENIX, 1995.

20. S. Kumar, C. Alaettinoglu, and D. Estrin. Scalable object-tracking through unattended techniques (scout). In *ICNP*, pages 253–262, 2000.

21. J. Kuruvila, A. Nayak, and I. Stojmenovic. Progress and location based localized power aware routing for ad hoc and sensor wireless networks. *IJDSN*, 2(2):147–159, 2006.

22. J. Li, L. Gewali, H. Selvaraj, and M. Venkatesan. Hybrid greedy/face routing for ad-hoc sensor network. In *DSD*, pages 574–578. IEEE, 2004.

23. K. Liu and N.B. Abu-Ghazaleh. Stateless and guaranteed geometric routing on virtual coordinate systems. In *MASS*, pages 340–346. IEEE, 2008.

24. N. Mitton, T. Razafindralambo, D. Simplot-Ryl, and I. Stojmenovic. Hector is an energy efficient tree-based optimized routing protocol for wireless networks. In *MSN*, pages 31–38. IEEE, 2008.

25. C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. *RFC 3561*, 2003.

26. J.A. Sánchez and P.M. Ruiz. Exploiting local knowledge to enhance energy-efficient geographic routing. In J. Cao, I. Stojmenovic, X. Jia, and S.K. Das, editors, *MSN*, volume 4325 of *Lecture Notes in Computer Science*, pages 567–578. Springer, 2006.

27. I. Stojmenovic. Localized network layer protocols in wireless sensor networks based on optimizing cost over progress ratio. *IEEE Network*, 20(1):21–27, 2006.

28. I. Stojmenovic and 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):175–188, 2004.

29. I. Stojmenovic and X. Lin. Power-aware localized routing in wireless networks. *IEEE Transactions Parallel Distrib. Syst.*, 12(11):1122–1133, 2001.

30. H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

31. G.T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.

32. P.F. Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. In *SIGCOMM*, pages 35–42, 1988.

33. Y. Wang, W.-Z. Song, W. Wang, X.-Y. Li, and T.A. Dahlberg. LEARN: Localized energy aware restricted neighborhood routing for ad hoc networks. In *SECON*, pages 508–517. IEEE, 2006.

34. J. Wu. Dominating-set-based routing in ad hoc wireless networks. In I. Stojmenovic, editor, *Handbook of Wireless Networks and Mobile Computing, chapter 20*, pages 425–450. John Wiley & Sons, 2002.

35. H. Zhang and H. Shen. Eegr: Energy-efficient geographic routing in wireless sensor networks. In *ICPP*, pages 67–75. IEEE Computer Society, 2007.