G-STAR: Geometric STAteless Routing for 3-D Wireless Sensor Networks

Min-Te Sun[a,1], Kazuya Sakai[b], Benjamin R. Hamilton[c], Wei-Shinn Ku[b], Xiaoli Ma[c]

[a]*Department of CSIE, National Central University, Taoyuan 320, Taiwan*
[b]*Department of CSSE, Auburn University, Auburn, AL 36849, USA*
[c]*School of ECE, Georgia Institute of Technology, Atlanta, GA 30332, USA*

Abstract

3-D aerial and underwater sensor networks have found various applications in natural habitat monitoring, weather/earthquake forecast, terrorist intrusion detection, and homeland security. The resource-constrained and dynamic nature of such networks has made the stateless routing protocol with only local information a preferable choice. However, most of the existing routing protocols require sensor nodes to either proactively maintain the state information or flood the network from time to time. The existing stateless geometric routing protocols either fail to work in 3-D environments or have tendency to produce lengthy paths. In this paper, we propose a novel routing protocol, namely Geometric STAteless Routing (G-STAR) for 3-D networks. The main idea of G-STAR is to distributively build a location-based tree and find a path dynamically. G-STAR not only generalizes the notion of geographic routing from two modes to one mode, but also guarantees packet delivery even when the location information of some nodes is either inaccurate or simply unavailable regardless the uses of virtual coordinates. In ad-

[*]Corresponding author: Min-Te Sun
E-mail address: msun@csie.ncu.edu.tw

dition, we develop a light-weight path pruning algorithm, namely Branch Pruning (BP), that can be combined with G-STAR to enhance the routing performance with very little overhead. The extensive simulation results by ns-2 have shown that the proposed routing protocols perform significantly better than the existing 3-D geometric routing protocols in terms of delivery rate with competitive hop stretch. We conclude that the proposed protocols serve as a strong candidate for future high-dimensional sensor networks.

Keywords: Geometric routing, wireless sensor networks, 3-dimensional networks

1. Introduction

3-D wireless sensor networks (WSNs), such as underwater acoustic sensor networks [1] and aerial networks [2], are a special type of wireless networks, which has vast applications in natural habitat monitoring, weather/earthquake forecast, terrorist intrusion detection, and homeland security [3]. The resource-constrained and dynamic nature of such networks, such as nodes have limited memory and inaccurate local information, has made the design of 3-D routing protocols complicated. Compared with the traditional proactive [4] and reactive [5] routing protocols, geometric routing [6, 7, 8, 9] is more suitable as it only uses the local location information to deliver packets with low communication and storage overheads. In most geometric routing protocols, there is no need for nodes to maintain global state information nor flood the network in search of a path to the destination.

In general, for many existing geometric routing algorithms to function correctly, three assumptions are required. First, the location information at each node needs to be accurate. Second, the link model is assumed to be the unit disk model for 2-D networks or unit ball model for 3-D networks, i.e., nodes within a predefined transmission radius can always exchange packets with each other. Last, the network topology needs to be on a 2-D plane. In practice, it is difficult to obtain accurate location information at each node regardless what localization algorithm is used. The protocols that assume the unit disk model or unit ball model can result in very poor performance in reality [10]. In addition, nodes in many WSNs are deployed over 3-D space such as aerial networks and underwater wireless sensor networks. As a result, these assumptions render most of the existing geometric routing protocols unusable for 3-D WSNs.

To do away with the aforementioned assumptions, a number of effort have been made. The geometric routing protocols proposed in [11, 9, 12, 13] are able to relax some of these assumptions, but they either produce lengthy path, introduce heavy communication overhead, or require each node to proactively maintain routing information. On the other hand, routing with virtual coordinates [14, 15] has been proposed for the locationless networks. In [14, 15], nodes in the network set up their virtual coordinate by using centroid transformation. 3RuleGeo protocol [?] routes a packet by traversing unvisited nodes based on the constructed virtual coordinate by using Path Recording [?]. VRR [16] and VCP [17] use a distributed hash table (DHT) to create an overlay network topology in ad hoc networks. In such a virtual topology, greedy strategy always works, and so packets are forwarded to the node closest to the destination in the virtual topology. Although these protocols work in the network where location information is not available or inaccurate, additional overhead is required to set up virtual coordinates or an overlay network topology.

In this paper, we propose a novel geometric routing protocol, namely Geometric STAteless Routing (G-STAR). The main idea of G-STAR is to distributively build a location-based tree and find a path dynamically. G-STAR has the following preferred properties:

- G-STAR is stateless: Unlike GDSTR [9], the prioritized DFS [18, 19], and GHG [13], no state information is proactively maintained at each node in G-STAR.
- G-STAR runs in one mode: In most of the geometric routing protocols, there are more than one mode, such as the greedy mode and the detouring mode, when a packet is routed toward its destination. Using only next-hop selection criterion, there is no need for G-STAR to switch between modes or perform network topology planarization.
- G-STAR guarantees packet delivery even when the location information at some nodes is inaccurate or missing. Unlike [14, 15, 20, 21, 16, 17], G-STAR does not rely on the availability of virtual coordinate.
- G-STAR can be combined with the Branch Pruning algorithm (BP), a lightweight algorithm which effectively removes the loops in the path with small overhead.
- G-STAR is compatible with the concept of virtual coordinates. When the location information is unavailable, G-STAR can route the traffic using virtual coordinates to further improve the performance.

The extensive simulation results validate that the proposed G-STAR routing algorithm significantly outperforms the existing 3-D geometric routing protocols, such as GRG [12], GHG [13], GDSTR [9], and 3RuleGeo [**?**].

The rest of the paper is structured as follows. Section 2 surveys the related work of geometric routing. Geometric STateless Routing protocol as well as the Branch Pruning algorithm are illustrated in Section 3 and the experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

The traditional wireless ad hoc routing protocols are classified into proactive and reactive protocols. In proactive protocols, such as DSDV [4], each node periodically exchanges routing information to maintain a routing table. In reactive protocols, such as DSR [5], a node floods the entire network to discover a path to its destination whenever it has a packet to send. While the traditional protocols are able to handle 3-D routing, they result in poor performance in large scale WSNs due to their communication and storage overheads.

For large-scale WSNs, geometric routing protocols [7, 22, 23, 8] are known to be light-weight and efficient. In general, these geometric routing protocols can determine where to forward a packet without maintaining a routing table or flooding the network as long as each node in the network knows the location of itself and its neighbors, and the source knows and encodes the destination location in the packet. These geometric routing protocols commonly consist of the greedy forwarding and detouring modes. In greedy forwarding mode, a node forwards a packet successively to the neighbor closest to the destination. When no closer neighbor to the destination is found, i.e., the local minimum is reached, the packet enters the detouring mode. Different geometric routing protocol handles the detouring forwarding differently. Depending on the dimension of each node's location, these geometric routing protocols are categorize into 2-D, 3-D, and no location information as follows.

2.1. 2-D Geometric Routing

GPSR [7] and the face routing family, e.g., AFR [22], GOAFR [23], and GOAFR$^+$ [8] are the most popular 2-D geometric routing protocols. The detouring strategy in this category recovers a packet by forwarding it on the planar graph with certain rules to avoid repeated loops. For a given network topology, several distributed algorithms, such as GG [24] and RNG [25], are available to planarize a network topology. In these algorithms, each node autonomously eliminates some edges to its neighbors from the consideration of the detouring routing based on the localized information so that the resulting graph does not contain cross edges. For example, in the detouring mode of GPSR, a packet is forwarded successively on closer faces of the planar graph until it reaches the destination or a node closer to the destination than the previous local minimum, where the packet is switched back to the greedy forwarding mode.

The geometric routing protocols [7, 22, 23, 8] guarantee packet delivery under the assumptions that all nodes in the network have accurate location information, the link model is the unit disk, and the network topology is on a 2-D plane. However, these assumptions are not available in real sensor network scenarios. When some of these assumptions are not satisfied, these protocols either fail to deliver the packet or lead to poor routing performance [10].

2.2. 3-D Geometric Routing

The stateless geometric routing in 3-D networks is more challenging and difficult than in 2-D networks. There are several works to do away with the aforementioned expensive assumptions. A practical planarization technique, CLDP [10], is proposed to make 2-D geometric routing work with imprecise location information and general link model. In CLDP, each node periodically probes its neighbor to eliminate cross edges. CLDP is able to accommodate arbitrary graph and can be applied to GPSR and the face routing family. The geometric routing using GPSR with CLDP will work on 3-D topologies, but the performance can drastically decreases on higher dimensional topologies. In addition, CLDP incurs too much communication overhead as shown in [9]. In [11], each node is assigned a virtual coordinate in hyperbolic plane and the greedy forwarding is performed with respect to these virtual coordinates. However, the evaluation of the virtual coordinate is not localized and has to be updated whenever the topology changes. In GDSTR [9], a spanning tree is constructed for the entire network and each node makes use of the convex hull of its children in the tree for routing. There are several drawbacks in GDSTR. First, GDSTR requires multi-dimensional convex hull calculations which would demand significant processing resources. Second, each node needs to maintain a convex hull for each children, and it introduces high storage overhead in proportion to the scale of the network. Third, if location information is inaccurate, a defective spanning tree will be constructed and routing may fail. Finally, the spanning tree needs to be reconstructed should the network topology change, and the overhead to reconstruct the spanning tree is extremely high. These protocols are considered too expensive for 3-D WSNs, as they commonly incur high control overheads.

In [12], a randomized geometric routing, called GRG, is proposed which consists of the greedy forwarding and random walk modes. When a packet reaches the local minimum, the packet is randomly forwarded within the bounded region until it recovers from the local minimum. If the packet is not recovered within a certain number of hops, the bounded region is extended and the random walk is repeated. While GRG is stateless and could be applied to an arbitrary graph, the random walk forwarding can lead to a long path. In [13], GHG is proposed which consists of the greedy forwarding and depth first search (DFS) modes. In detouring mode of GHG, a packet is forwarded by the prioritized DFS [18, 19] along a subgraph, called a hull graph. To obtain the hull graph, it employs PUDT [26] which captures the empty 3-D network subspaces to eliminate intersecting triangles and edges from the consideration of the DFS. To record visited nodes, each node keeps a routing table by marking visited neighbors for each destination. Even though GHG is very effective in an ideal environment, it does not work well in reality as PUDT assumes the link model of the unit ball graph. In addition, as each node has to maintain a routing table in the prioritized DFS, GHG incurs high storage overhead.

Although there are several routing 3-D geometric routing protocols, such as DBR [27], DSDR [28], and HH-VBF [29], they are primarily designed for underwater communications. Since our focus in this paper is geometric routing in general 3-D WSNs, we omit special physical constraints of acoustic communications from our consideration.

2.3. Routing without Location Information

The aforementioned geometric routing protocol require location information. However, in the real sensor networks, sensor nodes often do not have their location information or the location information is inaccurate. To handle such environments, routing with virtual coordinates are proposed in [14, 15**?**]. Since nodes are assumed not to have their location information, they first set up their virtual coordinate. Then, geometric routing is applied based on virtual coordinates instead of real location. Rao's protocol [14] consists of two phase. In the first phase, nodes are in perimeter are identified. In the second phase, non-perimeter nodes iteratively update their virtual coordinate by centroid transformation. In [15], centroid transformation is enhanced by eliminating the first phase of Rao's protocol. To fully take advantage of virtual coordinates, 3RuleGeo geometric routing protocol is proposed in [**?**]. Similar to the traditional geometric routing, 3RuleGeo protocol consists of the greedy and perimeter modes. In its perimeter mode, the sequence of already traversed nodes is used to select the next hop, which can be achieved by Path Recording [**?**].

Alternative approach is to construct an overlay network with a distributed hash table (DHT), which provide a look up service in distributed file systems and peer to peer networks. In [16], VRR is proposed to form a virtual ring to connect nodes by increasing node identifer across the networks. Then, a node forwards a packet to the node closest to the destination in ID space. The problem of VRR is that neighbors in the virtual ring may not be direct neighbors in the real network. The virtual cord protocol (VCP) is proposed in [17], in which the adjacent nodes in the virtual coordinate system are direct vicinity in the real network topology. Hence, VCP reduces the communication overhead and is more scalable. While routing on virtual coordinate can deliver packets in the networks where location information is not available, nodes are required to construct virtual coordinates or an overlay network before routing can take place.

3. Geometric Stateless Routing

In this section, we introduce our stateless geometric routing protocol, namely Geometric STAteless Routing (G-STAR), its properties, implementation considerations, as well as a post optimization technique, called Branch Pruning (BP).

3.1. Design Goals and Network Setup

Due to the nature of 3-D WSNs [1, 2], an ideal 3-D geometric routing protocol must achieve the following design goals.

- It should allow a node holding a packet to determine the next hop locally and distributively.
- It should be memory-efficient.
- It should not assume the model of unit ball graph (corresponding to the unit disk graph in 2-D).
- It should work even if location information is inaccurate or missing.

The notations used in the following discussion are listed in Table 1.

To ensure the proposed protocol functions properly, there are three required conditions of the network: (i) the link needs to be bi-directional so that a packet can traverse back as needed, (ii) source and destination are connected, and (iii) the network topology remains static during the routing process. Whether or not a link is bi-directional can be verified by its two endpoints by utilizing a link-layer protocol similar to the three-way handshake scheme in TCP [30]. As for the source-destination connectivity condition, since any pair of nodes can be selected as source and destination, this implicitly requires that the network to be

Table 1: Notations.

Symbol	Definition
$N(i)$	The open neighbor set of node i
L_i	The location of node i
p	The packet p
$p.src$	The source node of packet p
$p.dest$	The destination node of packet p
$p.L_d$	The destination location of packet p
$p.list$	The partial explored-node list of packet p

connected. The connectivity of a sensor network can be verified at the time of network deployment or be checked by one of localized topology control algorithms [31, 32, 33], regardless of the network dimension. The last condition is commonly assumed by most of the routing protocols. In fact, no routing protocol can guarantee packet delivery if the network topology changes fast during the routing process. In summary, these conditions can be fulfilled easily in real-world applications.

3.2. Protocol Description

In G-STAR, a node always routes a packet to the neighbor closest to the destination as long as no loop is created. To avoid loops, a packet in G-STAR is required to record the partial network topology it has already exploited. While there are known data structures to store the information of the graph for the partial network topology, such as the adjacency matrix and the adjacency lists [34], they require additional storage space and complicated operations for updates. In practice, we have found that storing merely a list of a subset of nodes which a packet has traversed is enough to effectively find a short route. This list is referred to as the *partial explored-node list*. By examining the partial explored-node list of a packet and the locations of neighbors, a node is able to determine where to forward the packet. When a packet is first generated, the partial explored-node list is initialized to be empty. When a node either generates or receives a packet, it first appends itself to the partial explored-node list and checks if it appears on the list more than once. If a node appears on the list twice, the nodes in between two entries are one of the branches the packet has just visited. There is no need to keep both entries in the list, so the earlier duplicated entry is kept and the new entry is removed. In other words, a node does not append itself to the list if it is already in the list. By doing so, a node records where the packet originally came from in the partial explored-node list as the last neighbor ahead of it in the list, which is called parent. At this moment, if the node still has neighbors not in the partial explored-node list, it forwards the packet to the one closest to the destination among them. Otherwise, the node will just forward the packet back to its parent. The pseudo code of G-STAR is given in Algorithm 1.

3.3. Example of G-STAR

In Figure 1, a 3-D example is provided to illustrate how G-STAR routes a packet. In the figure, the source and destination are S and D, respectively. The neighborhood relationship of the network is denoted by the dotted lines. According to the G-STAR algorithm, S appends itself to the list ($< S >$) and forwards it to N_1 since N_1 is closer to D than N_2. N_1 again appends itself to the list ($< S, N_1 >$) and then forwards it to N_3 as N_3 is closer to D than N_2. N_3 appends

Algorithm 1 G-STAR(N_s, N_d, L_d)

```
/* Node $i$ is the source node */
if ($i = N_s$) then
    $p.src \leftarrow N_s$
    $p.dest \leftarrow N_d$
    $p.L_d \leftarrow L_d$
    $p.list \leftarrow \phi$
end if
/* Node $i$ is the destination node */
if ($i = p.dest$) then
    a packet reaches the destination.
end if
/* Node $i$ processes the packet $p$ */
if ($\exists j \in N(i)$, $j$ is not in $p.list$) then
    adds $i$ into $p.list$.
    forwards $p$ to $j$ closest to the destination.
else
    if ($i = p.src$) then
        declares no route to the destination.
    end if
    forwards the parent node of $i$.
end if
```

itself to the list ($< S, N_1, N_3 >$) and forwards the packet to N_4, the neighbor closest to D. N_4 is a dead-end, so it appends itself to the list ($< S, N_1, N_3, N_4 >$) and sends it back to N_3, which it originally received the packet from. Note that N_4 can identify N_3 is its parent, as the last neighbor ahead of N_4 in the list is N_3. N_3 appends itself to the list and finds that there is a duplicated entry of N_3 on the list, so it removes the new entry from the list ($< S, N_1, N_3, N_4, >$) and forwards the packet to N_5, the last neighbor not on the list. N_5 is also a dead-end, so it appends itself to the list ($< S, N_1, N_3, N_4, N_5 >$) and forwards the packet back to N_3. At this moment, all of N_3's neighbors have been visited and on the list, so it forwards the packet to N_1, the last neighbor ahead of N_3 in the list. Similarly, after N_1 receives the packet, it forwards the packet to N_2, which is the only neighbor of N_1 not on the list, without appending itself to the list. Consequently, N_2, N_6, N_7, N_9, N_{12}, and N_{13} will append themselves to the list as the packet travels through them before the packet eventually reaches D.

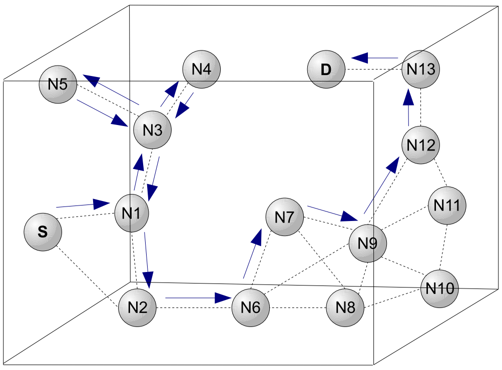


Figure 1: G-STAR routes a packet from S to D.

3.4. *Properties of G-STAR*

G-STAR has several unique and useful properties. Before we start introducing the properties, we define some useful terms.

Definition 1 (Connected component) *A component of the graph that is completely connected, i.e. for every pair of nodes in the component there is at least one sequence of edges that form a path from one node to the other.*

Definition 2 (Degree) *The degree of a vertex is defined as the number of incident edges on a given vertex. In a network, this is equivalent to the number of neighbors which a node has.*

Property 1 (DFS) *The path established by G-STAR forms a depth-first search (DFS) tree on an undirected graph where links are visited in increasing order of the corresponding node's distance to the destination.*

Proof 1 *The DFS algorithm recursively traverses a graph. At each node, the node is marked as visited and a DFS is completed for each of the unvisited neighbors in turn – when a subgraph rooted in a given neighbor has been completely traversed, the search begins at the next unvisited neighbor [35]. Under the G-STAR protocol path finding obeys similar rules. As in the DFS, at each node the packet is forwarded to the unvisited neighbor with the shortest distance to the destination. When there are no unvisited neighbors the packet will backtrack. The parent node that G-STAR backtracks to is the same node that the depth-first search returns to. Thus the path G-STAR follows in the network directly maps to the path formed by DFS of the network.*

Property 2 (Overhead) *The transmission overhead (the size of partial explored-node list) is upper bounded by $O(n)$ where n is the number of nodes in the network.*

Proof 2 *A node ID, say i, is added into the partial explored-node list $p.list$ of a packet p, when it has been visited. When the packet traverses node i the second time, it finds a duplicated entry of i, so the new entry i is removed. Hence, there is no duplicate entry in the partial explored-node list. The transmission overhead is upper bounded by $O(n)$.*

Property 3 (Edge crossing) *When routing using the G-STAR protocol, a packet will only cross an edge at most twice.*

Proof 3 *In the G-STAR algorithm, a packet is forwarded from a parent to an unvisited node. After this node is visited it is marked visited. Thus, the parent will not forward the packet again to this node. The packet will explore the subtrees rooted at the children of this node. If the destination is not part of the subtree, the current node will eventually forward the packet to the parent. Since this node is marked visited, no other node will forward the packet to the current node. This implies that this node will never receive the packet again, and thus the link between the parent and the current node is only crossed twice. This is true for all nodes in the tree. Since the path searched forms a tree, the only edges in the network used are those from parents to their children. Thus this is true for all links used.*

Note that this property shows that there is no repeated loop in the path. It also demonstrates that when the G-STAR protocol is adopted, the number of hops (including the return hops) necessary to explore any given subtree that does not contain the destination is equal to twice the number of nodes in the subtree.

Property 4 (Reception guarantee) *If there is at least one route from the source to the destination the G-STAR algorithm will find one.*

Proof 4 *From Property 1, G-STAR performs DFS of the network and only stops when it reaches the destination. Thus if G-STAR completes the DFS, the destination does not belong in the connected component of the network explored by the packet. Since the connected component of the network by definition contains all of the vertices reachable from any node in that component [35], there is no route to the destination.*

Property 5 (Termination) *When there is no route to the destination, G-STAR will visit each node in the connected component containing the source at least once. All nodes except the source will be visited no more than their* $degree$*. The source may be visited* $degree + 1$ *times.*

Proof 5 *If the destination is not part of the connected component containing the source, G-STAR will perform a depth-first search of this component. From [35], a depth-first search visits every node in the connected component. This search only visits a node from its parent and each of its children. A node can have at most* $degree - 1$ *children and* 1 *parent or* $degree$ *children and no parent. All nodes except the source have a parent. A node will only receive a packet once from its parent and each child. The source will generate the packet and visit itself. So the source can be visited at most* $degree + 1$ *times and all other nodes can be visited* $degree$ *times.*

Property 6 (Undirected graph) *The G-STAR protocol works on any undirected graph.*

Proof 6 *From Property 1, G-STAR travels along the same path as a DFS on an undirected graph. Since DFS will visit each node on the connected component containing the source, G-STAR will deliver the packet to the destination if it is contained in the connected component, or return to the source.*

This means that G-STAR functions on networks with significant location estimation errors or even when the network topology is independent of geography (such as Bernoulli graphs [36]).

3.5. Extension to Multi-Dimensional Networks

Property 6 also implies that G-STAR algorithm can be naturally extended to operate on multi-dimensional networks. The key for this extension is two-fold: (i) the introduction of an N-D node ID; and (ii) a definition of the metric used to determine the "better" neighbor. With these two definitions, an N-D network is mapped to a space with a measure. Define an ID for N-D node n_i as:

$$\bar{d}_i = [d_i^1 \; d_i^2 \; \cdots \; d_i^N]^T, \tag{1}$$

where d_i^n represents the node characteristic on the n-th domain, and T denotes the transpose of the vector. d_i^n can be coordinate on the n-th axis or generally a value to quantify the node's character on this domain. The forwarding and tree-search metric used to determine the "closest" neighbor to the destination is a scalar function $M(\bar{d}_i, \bar{d}_j)$. One example of such a function is the Euclidean distance

$$M(\bar{d}_i, \bar{d}_j) = ||\bar{d}_i - \bar{d}_j||_2^2, \tag{2}$$

where geographic location is an example. Another example of the metric is the Hamming distance

$$M(\bar{d}_i, \bar{d}_j) = |\{\eta : \bar{d}_i \neq \bar{d}_j\}|, \tag{3}$$

where $|\cdot|$ denotes the cardinality of the set. This metric can be applied to some networks with binary ID like Internet with IP address. Different metrics can be defined for different networks.

With this definition of the N-D node ID and distance metric, G-STAR can be applied to N-D networks. This suggests that G-STAR can be applied to networks where the “position” is either replaced or augmented by an application-dependent vector ID. Such an augmented position could be used to allow networks with incompatible physical layers to be merged into a single routing domain. This can also be used as a model for social networks where each attribute can be considered as a dimension.

3.6. Branch Pruning

In [37], a post optimization technique, namely Path Pruning (PP), is introduced that can be applied to improve the performance of geometric routing protocols. We have found that the similar technique works even better with G-STAR. The basic idea of PP is that a node listens to the wireless radio channel after it transmits a packet. If after a short period of time the node finds that the same packet is transmitted by one of its neighbors different from the one it previously forwarded the packet to, it identifies this neighbor as the next hop for the destination of the packet and forwards the subsequent packets for the same destination directly to this neighbor. PP helps identify the shortcuts that are not utilized by the non-flooding detouring strategies in the existing geometric routing.

Although PP keeps next hop entries at a subset of nodes on the path, this information is passively acquired by listening to the wireless channel. There is no communication cost to maintain the next hop entry. In addition, the state information is kept only for active connections. If a node with a next hop entry for a destination does not receive subsequent packets for the destination, the next hop entry will time out and be deleted. With the help of PP, the routing performance of geometric routing protocols improves dramatically at critical network densities.

To further minimize the overhead of path pruning, we propose a light-weight path pruning, namely Branch Pruning (BP) that goes with G-STAR. In BP, if a node forwards a packet to a neighbor different from its greedy choice, it keeps a next hop entry for the destination of that packet. When a packet comes back from one neighbor and gets forwarded to another one, the next hop entry is updated accordingly. If a node receives subsequent packets for a destination and it has a next hop entry for the destination, it forwards the packets directly to the one indicated in the next hop entry. Otherwise, it forwards subsequent packets to its greedy choice. The pseudo code of the BP algorithm is provided in Algorithm 2. As an example, in Figure 1, when N_1 first forwards a packet destined for D to N_3, N_1 will not record the next hop entry for D as N_3 is the greedy choice for destination D. However, when N_1 later forwards the packet to N_2, N_1 will keep a next hop entry $< D, N_2 >$ and N_1 will then forward the subsequent packets for D directly to N_2.

Since there is no planarization process involved in G-STAR, all edges, including different types of shortcuts, will be considered for packet routing. In addition, BP helps G-STAR to identify and remove loops. As a result, although G-STAR with BP does not require a node to passively listen to neighbor's transmissions in search of the same packet it transmitted earlier nor a sensor node to keep the information of every packet it transmits, it shows excellent routing performance. The only type of shortcuts G-STAR with BP fails to identify is those that lead the packet away from the destination, such as the link between N_6 and N_9 in Figure 1. In the original path pruning algorithm, N_6 will overhear the transmission of the same packet from N_9, and therefore N_6 is able to identify a shortcut to N_9. However, we have found that such shortcuts appear rather infrequently in our performance studies.

Algorithm 2 BP(N_s, N_d, L_d)

```
/* Node $i$ receives the first packet $p$ of a flow. */
if ($i \in p.list$) then
    if ($\exists j \in N(i)$, $j$ is not in $p.list$) then
        Set next node as $j$ closest to the destination.
    end if
end if
/* Node $i$ sends subsequent packets of a flow. */
if ($i$ has a routing entry for $p.dest$) then
    Forward $p$ to the next node.
else
    Forward $p$ by the greedy choice.
end if
```

3.7. Implementation Considerations

There are a few considerations and optimizations on G-STAR that may be beneficial in actual implementation.

3.7.1. Inaccurate Location Information

Basically, G-STAR is a prioritized DFS that gives the visit (i.e., packet forwarding) priority to the neighbor closer to the destination. Hence, it goes without saying that a packet will definitely reach the destination as long as the network topology remains stable during the time of routing process and source and destination are connected, regardless how accurate the location information at each node is.

3.7.2. Partial Location Information

In case if some nodes in the network do not have the location information at all, the order of exploration among neighbors can be assigned so that the priority is given to those that lead the packet closer to the destination, then to those without the location information, and finally to those that lead the packet away from the destination.

3.7.3. No Location Information

When no node has location information, G-STAR is still able to route the packet, but the performance may be low. In such a case, one of the virtual coordinate protocols, such as centroid coordinates [15], can be used to obtain the virtual coordinate for each node. Subsequently, G-STAR can be applied to route the traffic by means of virtual coordinates with a much better performance.

3.7.4. Mobility

G-STAR is also robust to mobility. GHG uses PUDT to obtain a hull graph by exchanging beacon within two-hop neighbors. Thus, the subgraph is vulnerable to small topology changes. On the other hand, G-STAR only relies on one dynamic list, $p.list$. Since the partial explored-node list is only valid for the lifetime of the packet, G-STAR is relatively robust to topology changes. This leads to the following property:

Property 7 (Mobility) *As long as the connectivity of any edge in the tree does not change during the lifetime of a packet, G-STAR will still perform identically to the static case.*

Proof 7 *Since a packet in G-STAR visits nodes in the network in the order of the DFS, if the connectivity of edges in the DFS tree does not change the route should be identical.*

3.8. Wireless Fading Environments

Wireless links experience path loss, multipath effects (reflection, diffraction, shadowing), and Doppler shifts due to mobility. Each of the multipath components is characterized by an attenuation, a phase shift, and a time-delay. The time-varying impulse response of the physical link can be expressed as:

$$c^{ch}(t;\tau) = \sum\ {}_{\nu}\alpha_{\nu}(t)e^{j\theta_{\nu}(t)}\delta(\tau - \tau_{\nu}(t)) \tag{4}$$

where $\delta(\cdot)$ denotes Dirac's delta function; and index ν denotes the ν-th path arriving with a delay $\tau_{\nu}(t)$, amplitude $\alpha_{\nu}(t)$, and phase $\theta_{\nu}(t)$.

The model in Eq. (4) is general enough to cover different types of wireless links. Here for simplicity, we present the well known Rayleigh fading channel to test our protocol. Suppose the wireless link $h_m(k)$ between nodes n_m and n_{m+1} in the kth time-slot is modeled with three effects [38]: the shadowing effect ζ_m, the attenuation due to the distance between these two nodes d_m and the small-scale random fading effect $\eta_m(k)$ as

$$h_m(k) = \zeta_m^{\frac{1}{2}}\ \eta_m(k)\ d_m^{-\frac{\alpha}{2}} \tag{5}$$

where d_m is the distance between n_m and n_{m+1} and α is the power loss exponent with a value between 2 (free space) and 4. The shadowing component ζ_m is assumed having a log-normal distribution [38] whose pdf can be described as

$$f_{\zeta}(x) = \frac{1}{x\sigma_{\zeta}\sqrt{2\pi}}\ e^{-(\ln x - \mu_{\zeta})^2/2\sigma_{\zeta}^2} \tag{6}$$

with μ_{ζ} and σ_{ζ}^2 being the mean and the variance of $\ln x$. The large-scale shadowing effect ζ_m and the attenuation term $d_m^{-\frac{\alpha}{2}}$ do not change during the time period of interest and therefore they do not depend on k. For small-scale fading, we assume that all paths are non-line-of-sight (NLOS) and $\eta_m(k)$'s are complex Gaussian distributed with zero mean and unit variance. Within one time-slot, the link does not change. For different time-slots, i.e., different k, $\eta_m(k)$'s are independent.

Note that when fading channel model is employed, the neighbors of a node can no longer be defined as nodes within a certain transmission range. In the actual implementation, if the signal power received by a node is higher than a pre-defined threshold, we consider the node a neighbor of the signal transmitter.

Greedy routing tends to choose a longer link for each hop. In a realistic channel as described in Equation 5, this has been shown to produce large packet error rates [39]. In [39, 40], modifications of the greedy forwarding metric were proposed to reduce the error rates associated with these wireless links. Unfortunately no solution has yet been found to improve the error rate associated with the detouring mode of protocols such as GPSR and GOAFR. Since G-STAR does not use a detouring mode, it is better able to take advantage of the reduced error rates.

The links are generated based on the model in Eq. (5). Because of the link uncertainty, apparently choosing the distance as routing metric is not an efficient way. If d_i is the distance between the node i and destination, here we choose the metric as $\Delta/h_m(k)$, where $\Delta = d_{next\ node} - d_{current\ node}$. That is, the neighbor with the smallest $\Delta/h_m(k)$ among these unvisited is chosen as the next hop.

4. Performance Evaluation

To evaluate the performance of the proposed G-STAR and G-STAR with BP protocols, we compared our protocols with other four well-known 3-D routing protocols, GRG [12], GHG [13], GPSR with CLDP [10], GDSTR [9], and 3Rule-

Geo /citeconf/ipmrc/3ruleGeo on top of centroid virtual coordinates [15], by ns-2 [41]. Note that CLDP can not planarize the graph in 3-D, and thus we project the network topology into 2-D to calculate a subgraph. In this section, the simulation configurations, metrics, and results of different routing protocols are presented.

4.1. Simulation Configurations and Metrics

In our simulations, the network topology is randomly generated by placing nodes in a $500 \times 500 \times 500 m^3$ cube. The effective transmissions in free space propagation model and fading channel model are set to be $100m$. IEEE 802.11 [42] is used as the MAC layer protocol. The total number of nodes ranges from 100 to 350, which corresponds to network density ranging from 3.27 to 11.72 neighbors per node. For a given network density, 100 realizations of network topology are generated. For each realization, a pair of source and destination, which are connected through the network, are randomly selected. Five flows are simultaneously created and each source node generates a constant bit-rate (CBR) traffic flow to its destination. Each CBR flow sends 5 consecutive packets of 100 $bytes$ at the transmission rate of 2 $Mbps$. The inter-arrival time of packets is set to be 0.25 seconds.

Four kinds of simulation scenarios are conducted, (i) all nodes have accurate location information, ii) location information of some nodes contains errors, (iii) some nodes do not possess any location information at all, and (iv) the wireless fading model is applied. In scenario (ii), the degree of error ranges from 0% to 100% with respect to the transmission range. In scenario (iii), the percentage of nodes which do not hold location information at all ranges from 0% to 50%. In scenario (iv), the Rayleigh fading channel model in Section 3.8 is used. The receiving power threshold, which is used to determine if a node is a neighbor in the fading channel model, is set to be $4.22e-9$. We consider the value of signal-to-noise ratio (SNR), which represents the ratio of the transmitted power and the noise at the receiver, to be 5dB and 20dB. Note that the results of 3RuleGeo on top of virtual coordinates are included only in scenario (ii) and (iii) as it makes no sense to use virtual coordinates when the location information is accurate.

To evaluate the proposed routing protocols, different metrics including delivery rate, end-to-end delay, hop stretch, communication overhead, and storage overhead are utilized. Their definitions are provided as follows.

1. Delivery rate – The delivery rate is defined as the ratio of the total number of received packets and the total number of generated packets. In wireless fading model, some bits in a packet may not be correctly received. Hence, in scenario (iv), the delivery rate is defined as the ratio of the number of bits delivered at the destination without error and the number of generated bits at the source. Note that here we are simply trying to quantify the performance of different routing protocols. Since in the actual implementation of wireless networks bits are transmitted in packets with certain error correction code (e.g., FEC [30]), some bit errors may be recovered. Hence, the actual delivery rate in the presence of channel fading should be slightly higher.
2. End-to-end delay – The end-to-end delay is defined as the duration from the time a packet is generated to the time the packet is received by its destination. Note that only packets reach their destinations are considered.
3. Hop stretch – The hop stretch, as defined in [23], is calculated by the number of hops obtained by geometric routing protocols divided by the hop number of the shortest path obtained by Dijkstra's algorithm [34].
4. Communication overhead – For G-STAR, the extra information (i.e., source

node ID, destination node ID, location of the destination, and partial explored-node list) stored in the packet header is considered as its communication overhead. The size of the partial explored-node list increases by 4 bytes (to store the ID of the node currently holding the packet) for each hop. For GHG, in addition to source node ID, destination ID, and location of the destination, it needs to include packet mode (greedy or detouring) and location that a packet enters detouring mode. GRG further includes the information of the bounded region used in the random walk mode. For CLDP, in addition to the extra information in the packet header, probes to planarize the graph is considered as communication overhead. Likewise, GDSTR introduces communication overhead in its convex hull tree construction.

5. Storage overhead – G-STAR, GRG and CLDP do not employ any routing table and hence no storage overhead is introduced. For G-STAR with BP, a node needs to keep the destination node ID and the next node ID in case a shortcut exists. In the detouring mode of GHG, nodes keep routing table including destination node ID, parent node ID, and visited node IDs. These are considered as storage overhead. In GDSTR, the nodes with children in the tree need to maintain convex hull information for each child and the visited children in the routing process.

4.2. Simulation Results of Different Protocols

In this subsection, we provide the simulation results in case all nodes have accurate location information.

Figure 2 shows the hop stretch and greedy success rate with respect to the network density. When the greedy success rate is low, a packet is more likely

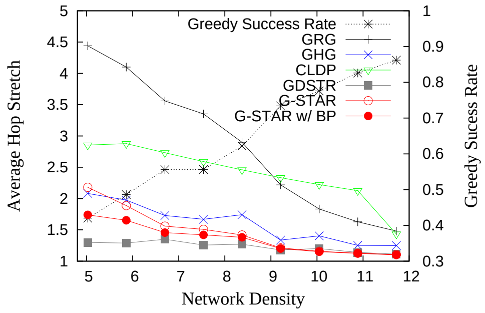


Figure 2: The hop stretch with accurate location information.

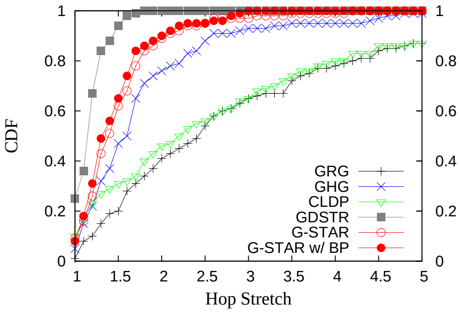


Figure 3: The cumulative distribution.

to enter the detouring mode. As shown in Figure 2, GDSTR generally results in shorter hop stretch than the other protocols. Note that GDSTR requires the calculation of a hull tree for the entire network before a packet can be transmitted. Although G-STAR and G-STAR with BP have higher hop stretch than that of GDSTR, the differences are not significant. GRG and CLDP incur much longer hop stretch. The reason why the hop stretch of CLDP is not stable is that CLDP does not guarantee packet delivery in 3-D and thus a packet may fall in an infinite loop.

Figure 3 illustrates the cumulative distribution of the hop stretch with respect to the number of hops. As can been seen in the figure, G-STAR, G-STAR with BP, and GDSTR can route a packet within 3 hops in most cases. This implies that G-STAR and G-STAR with BP have better routing performance than GHG, GRG, and CLDP in most cases.

Figure 4 presents the delivery rate with respect to the network density. G-STAR and G-STAR with BP always achieve 100% delivery rate, while GRG, GHG, CLDP, and GDSTR have lower delivery rate especially at the critical network density. CLDP does not guarantee packet delivery in 3-D networks. On the

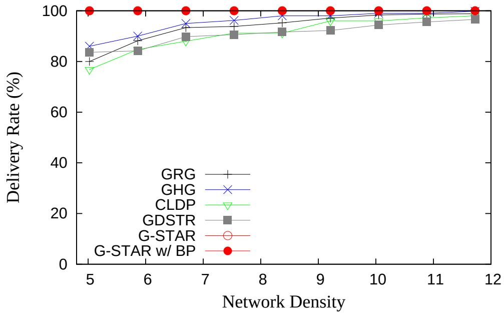


Figure 4: The delivery rate (%) with accurate location information.

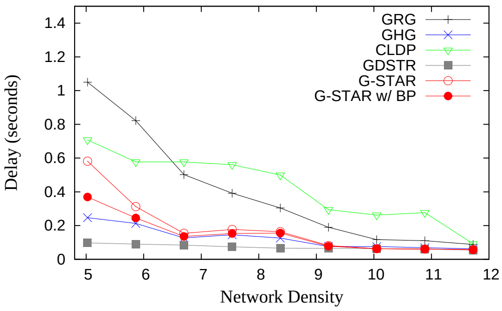


Figure 5: The end-to-end delay (sec.).

other hand, GDSTR floods the entire network to create a hull tree, which may lead to packet collisions. This suggests that our G-STAR and G-STAR with BP protocols are highly scalable.

Figure 5 demonstrates the end-to-end delay with respect to the network density. GDSTR results in shorter delay compared with the other protocols when the network density is low. While the hop stretch of G-STAR with BP is as low as that of GHG, the delay of G-STAR with BP is longer than that of GHG. This is because BP obtains a shorter route only after the first packet reaches its destination. Although GDSTR achieves the shortest delay, its low delivery rate, as illustrated in Figure 4, is not acceptable.

Figure 6 depicts the average communication overhead per delivered packet with respect to the network density. Note that CLDP introduces much higher communication overhead than the other five protocols by a large margin, and thus for ease of presentation of the other five protocols we omit the result of CLDP in this figure. In Figure 6, GDSTR has the highest communication overhead, as it requires the construction of the routing table. Although GDSTR results in the lowest hop stretch, it introduces high communication overhead, which is not acceptable

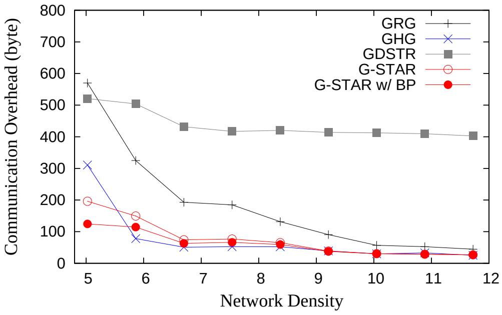


Figure 6: The communication overhead (byte).

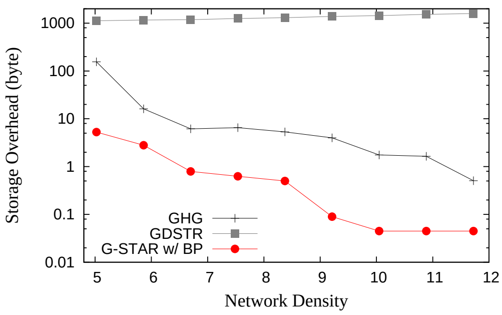


Figure 7: The storage overhead (byte).

in resource-limited sensor networks. Similar to GDSTR, the communication overhead of GRG is very high when the network density is low. Even though GRG is stateless, it produces longer routes and thus the communication overhead is very high. Although the packet size of G-STAR increases in proportion to the size of its partial explored-node list, the packet header is very simple compared with that of GHG. Therefore, the communication overhead of G-STAR and G-STAR with BP is lower than that of GHG.

Figure 7 provides the average storage overhead for delivered packets with respect to the network density. Recall that GRG, G-STAR, and CLDP are stateless. As can be seen from this figure, GDSTR incurs much higher storage overhead than GHG and G-STAR with BP. In GDSTR, each node needs to maintain a convex hull for each child in a hull tree. The size of storage increases in proportion to the number of nodes in the network. This implies that GDSTR is not scalable. On the other hand, GHG introduces much higher storage overhead than G-STAR with BP especially at the critical network density. This is because in the detouring mode of GHG nodes have to maintain their routing table.

4.3. Simulation Results with Inaccurate Location Information

In this subsection, the simulation results when nodes have inaccurate location information are presented. The location error is modeled as a uniform random variable in the range of $[-Err \times r, Err \times r]$, where Err is the percentage of error and r is the transmission range. In this scenario, the number of nodes is set to be 200, corresponding to 6.69 neighbors per node.

Figure 8 shows the hop stretch with respect to the percentage of error. It exhibits that the hop stretch increases in proportion to the percentage of location error. CLDP always results in the highest hop stretch. In general, GDSTR and 3RuleGeo have lower hop stretch than G-STAR and G-STAR with BP, but they either need to construct convex tree or virtual coordinates before routing can take place.

Figure 9 presents the delivery rate with respect to the percentage of error. As can be seen from the figure, the performance of GRG, GHG, CLDP, and GDSTR drastically decreases in proportion to the percentage of error. This is because GHG requires PUDT process to obtain hull graph which requires the assumption of the unit ball graph and GRG tends to produce longer routes. Although CLDP is able to accommodate location errors in 2-D, it results in poor performance as it is not designed for 3-D networks. In GDSTR, the location error leads to the construction of a defective hull tree, and routing along the hull tree is more likely to fail. G-STAR, G-STAR with BP, and 3RuleGeo always achieve 100% delivery rate. Although GHG has lower hop stretch than G-STAR, the low delivery rate of GHG is not acceptable. Figures 8 and 9 imply that our G-STAR, G-STAR with BP, and 3RuleGeo outperform the other protocols and are more suitable for 3-D WSNs when the location information is inaccurate.

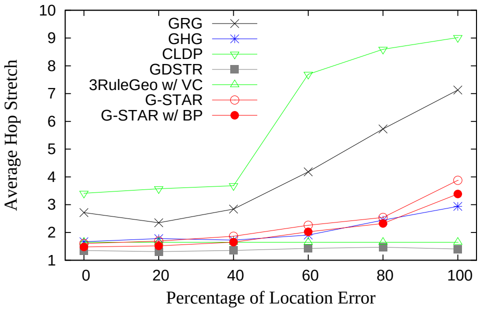


Figure 8: The hop stretch with inaccurate location information.

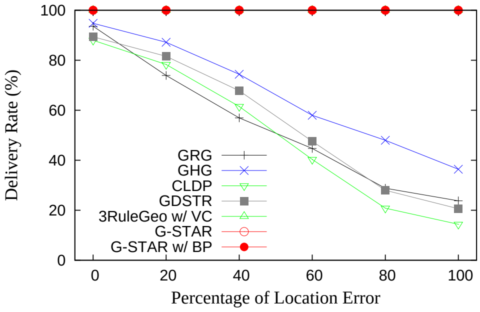


Figure 9: The delivery rate (%) with inaccurate location information.

4.4. Simulation Results with Partial Location Information

In this subsection, we demonstrate the simulation results when some nodes in the network do not possess any location information. The percentage of nodes without location information ranges from 0% to 50%. In this scenario, the number of nodes is set to be 200, corresponding to 6.69 neighbors per node. The routing of GRG is nearly random when some nodes do not have location information. In addition, PUDT in GHG, the right hand rule used in CLDP to remove cross edges, and the construction of a hull tree in GDSTR all assume that nodes have location information. The only protocols that work in locationless network are our two G-STAR variations and 3RuleGeo with virtual coordinates. Therefore, in this scenario we compare only the performance of G-STAR, G-STAR with BP, and 3RuleGeo. For fair comparison, all three protocols make use of virtual coordinates, which are computed by centroid transformation [15].

Figures 10, 11, and 12 demonstrate the hop stretch with respect to the percentage of nodes without location information for 0, 1, and 10 iterations of centroid transformation. As shown in the figure, the hop stretch increases when more number of nodes do not hold location information. In addition, higher number of

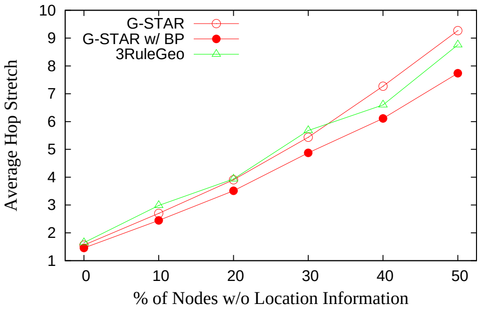


Figure 10: The hop stretch with partial location information. (0 iteration of centroid transformation)

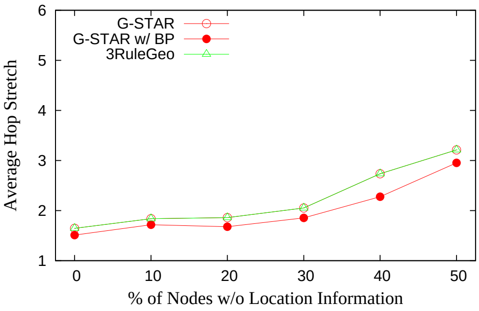


Figure 11: The hop stretch with partial location information. (1 iteration of centroid transformation)

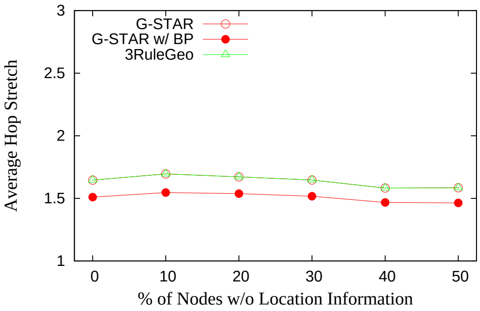


Figure 12: The hop stretch with partial location information. (10 iterations of centroid transformation)

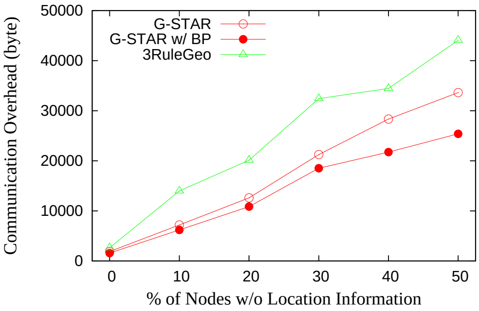


Figure 13: The communication overhead with partial location information. (0 iteration of centroid transformation)

iterations of centroid transformation leads to smaller hop stretch. Among the tree protocols, we can see that BP is very effective in reducing the hop stretch in case location information of some nodes is not available.

Figures 13, 14, and 15 show that the communication overhead with respect to the percentage of nodes without location information for 0, 1, and 10 iterations of centroid transformation. As can be seen, the communication overhead decreases as the number of iteration increases. Compared with 3RuleGeo, G-STAR incurs

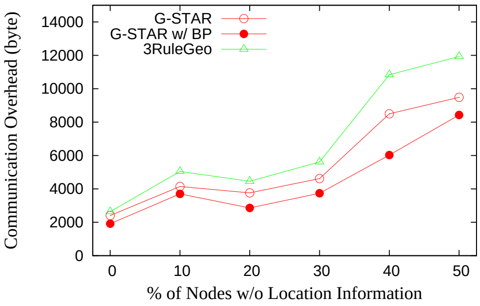


Figure 14: The communication overhead with partial location information. (1 iteration of centroid transformation)

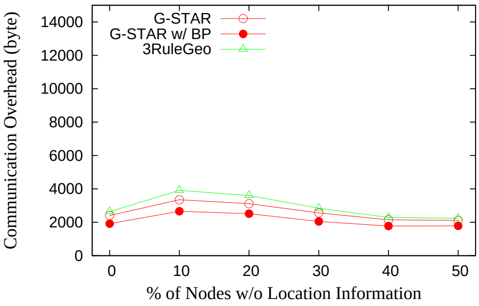
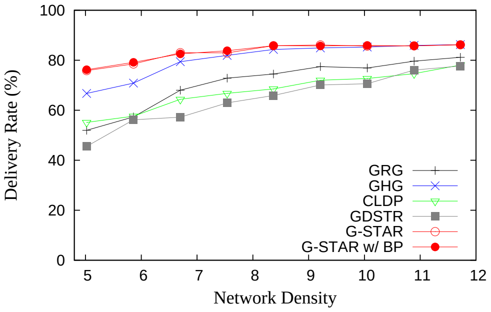


Figure 15: The communication overhead with partial location information. (10 iterations of centroid transformation)

less communication overhead. This is because that G-STAR does not record the same node ID in the partial explored-node list. G-STAR wit BP requires the smallest amount of communication overhead among three protocols.

4.4.1. Simulation Results in Wireless Fading Model

In this subsection, we show the simulation results when the Rayleigh fading channel model is applied. Figure 16 and 17 illustrate the delivery rate with respect to the network density with the SNR being set to 5dB and 20dB, respectively. Generally, the lower value of SNR, the more a loss wireless link experiences. As shown in the figures, all protocols result in higher delivery rate when SNR is set to 20dB. G-STAR and G-STAR with BP achieve higher delivery rate than the other protocols regardless of the dB value. This is because our metric considers the effect of wireless fading and avoids links with higher error rate. In addition, unlike other protocols, G-STAR and G-STAR with BP do not have the detouring mode. This means that our protocols can always take the effect of wireless fading into consideration.



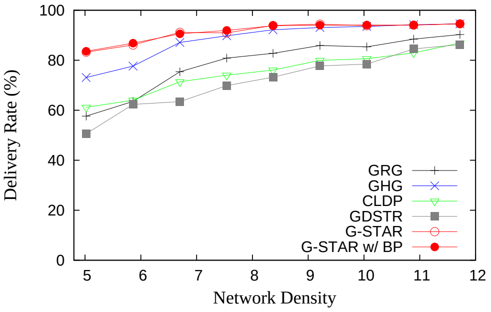


Figure 16: The delivery rate (%) (SNR is 5dB). Figure 17: The delivery rate (%) (SNR is 20dB).

5. Conclusion

Routing in 3-D wireless sensor networks (WSNs) is challenging due to their resource constraints and dynamic nature. While the stateless property of existing geometric routing protocols is attractive, they are not appropriate for 3-D WSNs because of their tendency to incur higher communication and storage overheads as well as their quick performance degradation when the location information of some nodes is inaccurate or missing. In this paper, we propose a stateless geometric routing protocol for 3-D wireless sensor networks, namely Geometric STAteless Routing (G-STAR). Our routing protocol builds a location-based tree on-the-fly and finds a short path when traversing the tree. G-STAR is robust in the sense that it functions well even when the location information is inaccurate or not available for some nodes in the network. To further reduce the path length, we propose a light-weight algorithm, namely Branch Pruning, to effectively optimize the path obtained from the proposed G-STAR routing protocol. The extensive simulation results have validated that the proposed routing algorithms achieve much higher delivery rate and competitive hop stretch compared with existing 3-D routing protocols, including GRG, GHG, CLDP, GDSTR, and 3RuleGeo.

References

[1] E. M. Sozer, M. Stojanovic, J. G. Proakis, Underwater Acoustic Networks, IEEE Journal of Oceanic Engineering 25 (1) (2000) 72–83.

[2] V. R. Syrotiuk, C. J. Colbourn, Routing in mobile aerial networks, in: Proc. WiOpt, 2003, pp. 293–301.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.

[4] C. E. Perkins, P. Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, in: Proceedings of the ACM International Conference on Communications Architectures, Protocols and Appliations (SIGCOMM), 1994, pp. 234–244.

[5] D. B. Johnson, Routing in Ad Hoc Networks of Mobile Hosts, in: Proceedings of IEEE Internatinal Workshop on Mobile Computing Systems and Applications.

[6] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with Guaranteed Delivery in Ad Hoc Wireless Networks, in: Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), 1999, pp. 48–55.

[7] B. Karp, H. T. Kung, GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, in: Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM), 2000, pp. 243–254.

[8] F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, Geometric Ad-hoc Routing: of Theory and Practice, in: Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing (PODC), 2003.

[9] B. Leong, B. Liskov, R. Morris, Geographic Routing without Planarization, in: In Proceedings of the 3rd conference on Networked Systems Design and Implementation (NSDI), 2006, pp. 25–25.

[10] Y.-J. Kim, R. Govindan, B. Karp, S. Shenker, Geographic Routing Made Practical, in: Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2005, pp. 217–230.

[11] R. Kleinberg, Geographic Routing Using Hyperbolic Space, in: Proceedings of the IEEE International Conference on Computer Communications (Infocom), 2007, pp. 1902–1909.

[12] R. Flury, R. Wattenhofer, Randomized 3D Geographic Routing, in: Proceedings of the IEEE International Conference on Computer Communications (Infocom), 2008, pp. 834–842.

[13] C. Liu, J. Wu, Efficient Geometric Routing in Three Dimensional Ad Hoc Networks, in: Proceedings of the IEEE International Conference on Computer Communications (Infocom), Mini Conference, 2009.

[14] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, I. Stoica, Geographic Routing without Location Information, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom), 2003, pp. 96–108.

[15] T. Watteyne, I. Augé-Blum, M. Dohler, S. Ubéda, D. Barthel, Centroid Virtual Coordinates - A Novel Near-Shortest Path Routing Paradigm, Computer Network 53 (10) (2009) 1697–1711.

[16] M. Caesar, M. Castro, E. B. Nightingale, Virtual Ring Routing: Network Routing Inspired by DHTs, in: Proceedings of the ACM International Conference on Communications Architectures, Protocols and Applications (SIGCOMM), 2006, pp. 351–362.

[17] A. Awad, C. Sommer, R. German, F. Dressler, Virtual Cord Protocol (VCP): A Flexible DHT-like Routing Service for Sensor Networks, in: Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2008), 2008, pp. 133–142.

[18] I. Stojmenovic, M. Russell, B. Vukojevic, Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks, in: Proceedings of the International Conference on Parallel Processing (ICPP), 2000, p. 173.

[19] I. Stojmenovic, M. Russell, B. Vukojevic, Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks 2 (21) (2002) 149–165.

[20] T. Watteyne, D. Simplot-Ryl, I. Auge-Blum, M. Dohler, On Using Virtual Coordinates For Routing in The Context of Wireless Sensor Networks, in: Proceeding of the International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2007.

[21] T. Watteyne, I. Auge-Blum, M. Dohler, D. Barthel, Geographic Forwarding in Wireless Sensor Networks with Loose Position-Awareness, in: Proceeding of the International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2007.

[22] F. Kuhn, R. Wattenhofer, A. Zollinger, Asymptotically Optimal Geometric Mobile Ad-Hoc Routing, in: Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), 2002.

[23] F. Kuhn, R. Wattenhofer, A. Zollinger, Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing, in: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2003, pp. 267–278.

[24] K. Garbriel, R. Sokal, A New Statistical Approach to Geographic Variation Analysis 18 (1969) 259–278.

[25] G. Toussaint, The Relative Neighborhood Graph of A Finite Planarset 12 (4) (1980) 261–268.

[26] X.-Y. Li, G. Calinescu, P.-J. Wan, Y. Wang, Localized Delaunay Triangulation with Application in Ad Hoc Wireless Networks, IEEE Transactions on Parallel Distributed Systtems 14 (10) (2003) 1035–1047.

[27] H. Yan, Z. J. Shi, J.-H. Cui, DBR: Depth-Based Routing for Underwater Sensor Networks., in: Proceedings of the IFIP Networking, 2008, pp. 1–13.

[28] D. Pompili, T. Melodia, I. F. Akyildiz, Routing Algorithms for Delay-Insensitive and Delay-Sensitive Applications in Underwater Sensor Net-

works, in: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), 2006.

[29] N. Nicolaou, A. See, P. Xie, J.-H. Cui, D. Maggiorini, Improving the Robustness of Location-Based Routing for Underwater Sensor Networks, 2007, pp. 18–21.

[30] J. F. Kurose, K. W. Ross, Computer Networking: A Top-Down Approach, 5th Edition, Addison-Wesley Publishing Company, 2009.

[31] R. Ramanathan, R. Hain, Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment, in: Proceedings of the IEEE 23rd International Conference on Computer Communication (INFOCOM), 2000, pp. 404–413.

[32] M. Bahramgiri, M. Hajiaghayi, V. S. Mirrokni, Fault-Tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-Hop Networks, Wireless Networks 12 (2).

[33] A. Ghosh, Y. Wang, B. Krishnamachari, M. Hsieh, Efficient Distributed Topology Control in 3-Dimensional Wireless Networks, in: Proceedings of the IEEE 4th International Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Vol. 12, 2007, pp. 91–100.

[34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, The MIT Press; 2nd Edition, 2001.

[35] S. Even, Graph Algorithms, Computer Science Press, 1979.

[36] B. Bollobas, Random Graphs, Academic Press, 1985.

[37] X. Ma, M.-T. Sun, X. Liu, G. Zhao, An Efficient Path Pruning Algorithm for Geographical Routing in Wireless Networks, IEEE Trans. on Vehicular Technology 57 (4) (2008) 2474–2488.

[38] T. S. Rappaport, Wireless Communicaitons: Principles and Practice, 2nd Edition, Princeton Hall, 2001.

[39] K. Seada, M. Zuniga, A. Helmy, B. Krishnamachari, Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys), 2004, pp. 108–121.

[40] B. R. Hamilton, X. Ma, Noncooperative Routing with Cooperative Diversity, in: Proceedings of the IEEE International Conference on Communications (ICC), 2007, pp. 4237–4242.

[41] The Network Simulator (ns-2) http://www.isi.edu/nsnam/ns/.

[42] IEEE 802.11b Standard
http://standards.ieee.org/getieee802/download/802.11b-1999.pdf.