This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Greedy Hop Distance Routing using Tree Recovery on Wireless Ad Hoc and Sensor Networks

Shao Tao
School of Computing
National University of Singapore
Email: shaot@comp.nus.edu.sg

A. L. Ananda
School of Computing
National University of Singapore
Email: ananda@comp.nus.edu.sg

Mun Choon Chan
School of Computing
National University of Singapore
Email: chanmc@comp.nus.edu.sg

***Abstract*—Connectivity-based routing protocols use the hop count vector to a group of anchors for packet forwarding. Due to the discrete nature of hop count based coordinates, without an effective recovery mechanism, these protocols will frequently encounter failures at network local minimum sites. In this paper, we propose a new connectivity-based routing protocol named Hop Distance Routing(HDR) with an efficient distance metric called *hop distance*. To ensure packet delivery, HDR complements greedy forwarding with a tree based recovery method, which allows the packet to traverse the branches and escape from local minimum locations. By labeling each tree node with an *angle range*, HDR can identify the subtree where the destination resides and forward the packet towards it until greedy forwarding can be resumed. Compared to other connectivity-based routing protocols such as NoGeo, BVR, LCR and VPCR, performance results show that the HDR protocol provides the highest packet delivery ratio and the lowest path stretch with the minimal communication overhead for each packet delivery on the node density range from 1.76 to 17.67.**

I. Introduction

Geometric routing protocols have shown great potential for wireless ad hoc and sensor networks, for having low control overhead and being highly adaptive to dynamic network topologies. Compared to the traditional on-demand wireless ad hoc routing, the geometric routing protocols eliminate the requirement of path discovery and maintaining routing states at intermediate nodes. The geometric routing protocols can be divided into two categories: position-based routing and connectivity-based routing. The position-based routing protocols use the physical device to identify each wireless node on a Cartesian coordinate space and apply greedy forwarding, face routing[1][2] or other mechanisms[3] for packet delivery. The connectivity-based routing protocols select a group of *anchor*(also named *landmark* or *beacon*) nodes such that by receiving the broadcast beacon from anchors, each node in the network will have a hop count vector denoting the distance to each anchor. For a network deployment with k anchors, the hop count vectors serve as the k-dimensional coordinates. By computing the k-dimensional Euclidean distance[4] or weighted Manhattan distance[5] through the hop count vectors, the connectivity-based routing protocols use greedy forwarding as the primary routing scheme and normally resort to scoped flooding or random walk to route packets out of the local minimum cases.

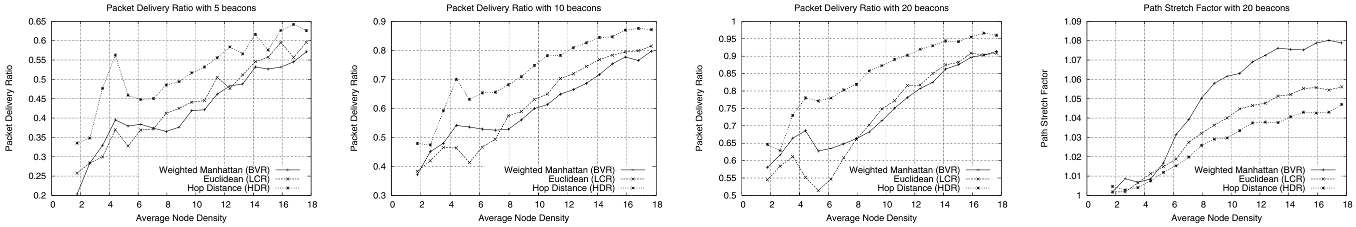
The Euclidean distance function fails to capture the fact that routing towards the anchors closer to the destination provides higher chance of success. The weighted Manhattan distance offers low precision due to its discrete nature and static priority allocation scheme. The flooding or random walk based recovery procedures also introduce enormous amount of overhead traffic. In this paper, we propose a new connectivity-based routing protocol named Hop Distance Routing(HDR) to alleviate the above problems. The HDR protocol uses a distance metric named *hop distance* computed from dynamic anchor priorities and hop count deviations to guide the packet forwarding. HDR employs one of the spanning trees rooted from the anchor nodes as the recovery backbone and allocates an angle range to each node based on their position on the tree, similar to the VPCR[6] protocol. By using the *hop distance* metric for greedy forwarding and the recovery tree for guaranteed packet delivery, the HDR protocol can achieve higher routing efficiency with lower overhead compared to other connectivity-based routing protocols such as NoGeo[7], BVR[5], LCR[4] and VPCR.

The rest of the paper is organized as the following. The related work on various localization algorithms and connectivity-based routing protocols will be given in Section II. The details on the greedy forwarding and tree-based recovery procedures of HDR will be explained in Section III. In Section IV, we will present the performance comparison of NoGeo, BVR, LCR, VPCR and HDR. The conclusion and future work will be provided in Section V.

II. Related Work

Some connectivity-based routing protocols directly use the hop count vectors to compute the Euclidean distance, e.g. LCR[4] and HIR[8] or Manhattan distance for greedy forwarding such as BVR[5]. Other connectivity-based routing protocols compute the virtual coordinates of nodes in the network and perform routing over the virtual coordinate system. The NoGeo[7] protocol and VCap[9] protocol select a group of perimeter nodes and compute the relative node position by trilateration. The LER[10] protocol provides a robust landmark selection algorithm by selecting a random landmark at each round in order to counter the distortions generated in previous rounds. As the hop count based coordinate systems generally introduce large quantization errors, the Aligned-VCS[11] is

978-1-4244-2075-9/08/$25.00 ©2008 IEEE



(a) delivery ratio, 5 beacons (b) delivery ratio, 10 beacons (c) delivery ratio, 20 beacons (d) path stretch, 20 beacons

Fig. 1. Delivery Ratio and Path Stretch with Greedy Distance Metrics of BVR, LCR and HDR

designed to take the average hop count vector among neighbors as the aligned coordinates of a node, which can be converted into a continuous domain with finer granularity. The nQUAD protocol[12] uses the relative distance between nodes and localized anchors to estimate the node position and conserve the original network orientation. While most of the landmark selection algorithms either use a randomized scheme or prefer the network boundary nodes from heuristics, Nguyen *et al.*[10] proposed a *r-sampling* landmark selection method to provide a bounded path stretch performance and guarantee packet delivery on a continuous domain. Instead of resorting to flooding or random walk based recovery procedures at local minimum sites, some connectivity-based protocols construct an underlying infrastructure that can be utilized when greedy forwarding fails. The VPCR[6] protocol creates a spanning tree for assigning virtual polar coordinates to all nodes on the network. During the recovery, packets are passed to upper levels where a predecessor node of the destination can be found. The Axis-Based VCap[13] builds a coordinate system consisting of a longitude and latitude value for each node based on their hop count to four anchor nodes and routing is performed by following the virtual longitude and latitude lines.

The Hop Distance Routing protocol(HDR) introduced in this paper uses a new distance metric named *hop distance* to guide the greedy forwarding and creates a tree based recovery procedure improved from the VPCR protocol. By complementing the *hop distance* greedy forwarding a the tree based recovery method, the HDR protocol provides higher packet delivery ratio and better path stretch performance with lower network traffic and storage overhead.

III. Protocol Design

The HDR protocol computes the priority of anchors dynamically for each packet and routing recovery is done through a backbone tree if greedy fails to proceed. In this section, we will present the details of the *hop distance* metric, the tree based recovery procedure and the routing algorithm.

A. Distance Function for Greedy Forwarding

For connectivity-based routing protocols, the most popular distance metrics include the Euclidean distance and the weighted Manhattan distance. While the Euclidean distance treats all anchors equally, the weighted Manhattan distance gives higher priority to the anchors closer to the destination. On a discrete domain, the routing accuracy of the Manhattan distance gets lower as the packet moves closer to the destination. Neither the Euclidean distance or the Manhattan distance differentiates the anchors scattered around the network boundary from those clustered within a close proximity, which makes them more vulnerable to local minimum sites in the network.

In order to assign higher priority to more promising anchors and adjust weights of anchors according to their mutual distance, we propose a new distance metric named *hop distance*, that will capture both features and provide more efficient routes for greedy forwarding. In a network with k anchors A_1, A_2, ..., A_k, each node will have a hop count vector to these anchor nodes once the broadcast beacons are received. For two nodes S and T, assuming that hop count vectors to these anchors are $[s_1, s_2, \ldots, s_k]$ and $[t_1, t_2, \ldots, t_k]$, the *hop distance* D_{hop} between nodes S and T can be computed from the hop-count difference vector Δ, the standard deviation σ of Δ and the maximum deviation η of Δ. The hop-count difference vector is computed as $\Delta = [\delta_1, \delta_2, \ldots, \delta_k]$, where $\delta_i = s_i - t_i$ for $i \in [1, k]$. The average hop-count difference $\bar{\delta}$ is calculated as $\bar{\delta} = (\sum_{i=1}^{k} \delta_i)/k$. The maximum deviation η is computed as $\eta = max\{\delta_i - \bar{\delta}\}, i \in [1, k]$. The standard deviation of hop-count difference Δ is $\sigma = \sqrt{(\sum_{i=1}^{k} (\delta_i - \bar{\delta})^2)/k}$.

To give higher priority to the anchors closer to the destination node T, we adjust the weight of the individual hop-count difference for each anchor, based on the hop count from the destination to the anchor nodes. The weights are denoted as $W = [w_1, w_2, \ldots, w_k]$ for $i \in [1, k]$, where $w_i = \frac{\sum_{i=1}^{k} t_i+1}{t_i+1}$ and t_i is the hop count from the destination to the i_{th} anchor. Once we have the hop-count difference vector Δ, the maximum deviation η and standard deviation σ of Δ and the weights W, we can compute the *hop distance* D_{hop} between node S and T by Equation. 1. The two adjustment factors ρ_1 and ρ_2 in Equation. 1 are used to control the significance of the hop-count difference deviations σ and η. Experiment results show that a combination of $\rho_1 = 5$ and $\rho_2 = 2$ gives good packet delivery ratio and path stretch values.

$$D_{hop}(S, T) \quad = \quad (\rho_1 + \eta)(\rho_2 + \sigma)\sqrt{\sum_{i=1}^{k} (w_i \delta_i^2)} \qquad (1)$$

To compare the efficiency of the *hop distance* metric with the Euclidean distance and the Manhattan distance, we created

20 topologies for 19 node densities ranging from 1.76 to 17.67, where the node number varies from 100 to 1000. We deploy 5, 10 and 20 anchors respectively and perform Greedy routing based on these three distance metrics. The average packet delivery ratios for the metrics are shown in Fig. 1. On average the packet delivery ratio of the *hop distance* metric is 12.05% higher than the Manhattan distance and 11.35% higher than the Euclidean distance. The path stretch factor for the *hop distance* metric also remains to be the lowest among the three metrics as shown in Fig. 1(d).

B. Recovery from Local Minimum

When greedy forwarding by the *hop distance* metric gets trapped at network local minimum, we resort to a tree-based recovery procedure adapted from the VPCR protocol. The VPCR protocol requires a localization method to generate virtual coordinates before building the spanning tree for angle range allocation. Our experiment reveals that even localization errors up to 40% of the communication range have only negligible effects on the path stretch performance of the tree-based recovery. Thus, the angle range assignment of HDR is randomized among the subtree nodes and localization is no longer required, which will significantly reduce the complexity and overhead traffic of the protocol.

As connectivity-based routing protocols require k anchor nodes to flood the network, we select the spanning tree rooted from the anchor with the smallest node ID to be the recovery tree. Each child node reports its subtree size to the parent in the bottom-up direction, until the parent nodes at each layer receive the complete subtree size information. The angle range is then assigned to each child in the top-down direction, starting from the root. Suppose a node P has k child nodes N_1, N_2, …, N_k, whose subtree sizes are s_1, s_2, …, s_k accordingly. If the current angle range of node P is $[\theta_1, \theta_2](0 \leq \theta_1 < \theta_2 \leq 2\pi)$, the angle range $[N_i.\theta_1, N_i.\theta_2]$ of its child node $N_i(i \in [1, k])$ can be computed as in Equation. 2, where $\varepsilon = 0.999$ is used to avoid the overlapping of angle ranges among sibling nodes.

$$\begin{aligned} N_i.\theta_1 &= \frac{\sum_{j=1}^{i-1} s_j}{\sum_{j=1}^{k} s_j}(P.\theta_2 - P.\theta_1) + P.\theta_1 \\ N_i.\theta_2 &= N_i.\theta_1 + \frac{s_i}{\sum_{j=1}^{k} s_j}(P.\theta_2 - P.\theta_1) \times \varepsilon \qquad (2) \end{aligned}$$

Since the parent node strictly assigns non-overlapping angles to different child nodes, the angle ranges of two nodes P and Q, denoted as $P.\Theta$ and $Q.\Theta$, can be either disjoint or contained one in another. If P is a predecessor of Q in the tree, we have $Q.\Theta \subseteq P.\Theta$ and vice versa. If P and Q are not the predecessor or successor for each other, we have $P.\Theta \cap Q.\Theta = \emptyset$. The detailed routing state transition diagram is given in Fig. 2. During recovery, if a node S holding the packet cannot find any neighbor that is a predecessor or successor of the destination node T, node S will route the packet in *Fallback* mode, in which the packet goes to the parent. If some neighbor N of S has $N.\Theta \supseteq T.\Theta$($N$ is a predecessor of T) or $N.\Theta \subseteq T.\Theta$($N$ is a successor of T), S will pass the packet to N and route the packet in *Branch* mode. In general, the *Greedy* mode is applied whenever we can minimize the current *hop distance* computed by Equation. 1. When a local minimum is reached, we switch to *Fallback* mode and the packet is passed to the upper layer until it encounters a node from where greedy forwarding can be resumed or the *Branch* mode can be enabled. In the *Branch* mode, a packet is forwarded along the branch to the destination relying on the hop count to the root as shown in Fig. 3.

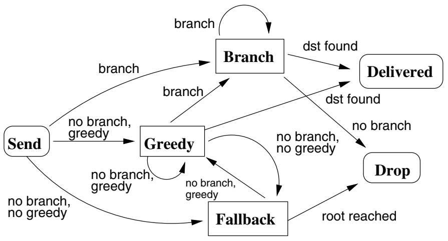


Fig. 2. Routing State Transition Diagram

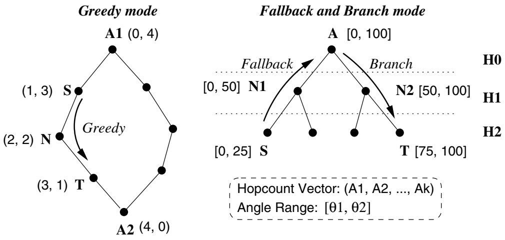


Fig. 3. Routing Modes: Greedy, Fallback and Branch

C. Routing Algorithm of HDR

The header of a HDR data packet to destination T contains the following fields: hc - hop count from T to root, Θ - angle range of T, $[A_1, A_2, ..., A_m]$ - hop count vector of T, $dist$ - minimum *hop distance* to T, H_d - the minimum hop count to T and m - the routing mode, which can be *greedy*, *fallback* or *branch*. Assuming a packet p arrives at node S with k neighbors N_1, N_2, ..., N_k, the routing decision is made by using Algorithm. 1. If there is a neighbor N_j that is a predecessor or successor of destination T and the hop count between N_j and T is smaller than $p.H_d$, node S set the routing mode $p.m$ to *Branch* and pass the packet to N_j. If no such neighbor is available, node S will compute the *hop distance* between $N_i(i \in [1, k])$ and T. If a neighbor N_j has the minimum hop distance to T less than $p.dist$, S will set the routing mode $p.m$ to *Greedy*, update the minimum hop distance $p.dist = D_{hop}(N_j, T)$ and pass the packet p to N_j. If both *Branch* and *Greedy* routing modes cannot be applied, S will set the routing mode to be *Fallback* and pass the packet to its parent node until the *Branch* mode or *Greedy* can be resumed. If in *Fallback* mode, the packet is trapped in the root or in *Branch* mode, there is no neighbor closer to the destination in terms of hop count, the packet will be dropped.

Algorithm 1 Routing Algorithm of HDR

```
inBranch(N,T) - true if T.Θ ⊆ N.Θ or T.Θ ⊇ N.Θ, else false
D_hop(N,T) - returns the hop distance between N and T
if p.m = Branch then                                   ▷ mode is Branch
    if ∃N_i, inBranch(N_i,T)=true and |N.hc − p.hc| < p.H_d then
        p.H_d = |N_i.hc − p.hc|, nexthop = N_i.
    else
        drop packet p.
    end if
else                                         ▷ mode is Greedy or Fallback
    if ∃N_i, inBranch(N_i,T) = true and |N_i.hc−p.hc| < p.H_d then
        p.m = Branch, p.H_d = |N_i.hc − p.hc|, nexthop = N_i.
    else if ∃N_i, D_hop(N_i,T) < p.dist then
        p.m = Greedy, p.dist = D_hop(N_i,T), nexthop = N_i.
    else
        if S ≠ root then
            p.m = Fallback, nexthop = parent.
        else                               ▷ root reached during fallback
            drop packet p.
        end if
    end if
end if
```

IV. Performance Evaluation

To compare the routing performance between HDR and some existing connectivity-based routing protocols, we implemented Greedy, NoGeo, BVR, LCR, VPCR and HDR in the network simulator *ns-2*. For NoGeo, we choose 2-hop local maximum nodes to be the candidate perimeter nodes and each perimeter node will suppress other candidates within 5 hops as suggested in [7]. For VPCR, we use NoGeo as the localization method to compute the polar coordinates. For BVR, LCR and HDR, we let 5% of the nodes be candidate anchors randomly, where each will suppress other anchors within 5 hops and the maximum number of routing anchors is 10. The performance metrics include the packet delivery ratio, the path stretch factor, and the communication overhead per packet delivery. We also compare the path stretch performance of HDR under different number of anchors in order to evaluate the tradeoff between routing efficiency and control overhead.

TABLE I
Input of Simulation Parameters

Name	*Value*
Field Size (A)	$400 \times 400m^2$
Node Number (n)	$100 \sim 1000$, step = 50
MAC Protocol	IEEE 802.11, 11Mps
Radio Range (R)	$30m$, CS range = $84m$
Topology	50 topos/density, uniform random
Connection	100 random pairs/topology
Routing Protocols	Greedy, NoGeo, BVR, LCR, VPCR, HDR
Beacon Number	$max = 10$ for BVR, LCR, HDR

The configuration of the simulation parameters are shown in Table. I. The average node density $d = \frac{n \times \pi R^2}{A}$ covers a range from 1.76 to 17.67. For each node density, 50 uniformly random topologies are generated and from each topology, 100 random pairs of nodes are chosen to be the source and destination. We use each of the 6 routing protocols to route data packets for each connection and the performance results are presented in the following sections.

A. Packet Delivery Ratio

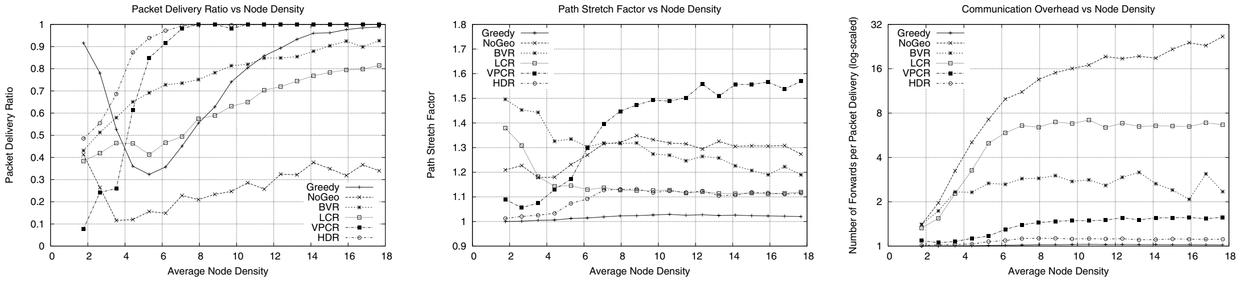
The packet delivery ratios of the protocols are shown in Fig. 4(a), where the flooding based recovery in BVR and LCR is disabled for a fair comparison. At the critical node density of 5.3 when the network is sparse, the delivery ratios for Greedy, NoGeo, LCR and BVR are 32.3%, 15.6%, 41.3% and 69.1%. In a sparse network, some nodes may be disconnected from the recovery tree, in which case only greedy forwarding is available for packet delivery. Due to the lack an effective greedy forwarding procedure, VPCR has a delivery ratio of 84.9%. By combining *hop distance* based greedy forwarding with the tree based recovery, HDR achieves the highest delivery ratio of 93.8%. At the maximum node density of 17.67, the packet delivery ratios of NoGeo, BVR and LCR increase to 33.9%, 92.7% and 81.5%. For VPCR and HDR, since the network is fully connected at this density, the packet delivery ratio converges to 100%.

B. Path Stretch Factor

The path stretch factor refers to the hop count ratio between the paths found by the routing protocols and the shortest path. A routing protocol with lower path stretch generates shorter paths on average and provides higher energy efficiency with less network forwarding traffic. The path stretch results for the protocols are given in Fig. 4(b), where the path stretch exhibits large variation in the density range of $2 \sim 8$ and stabilizes after the density goes beyond 8 for all protocols. At the critical density of 5.3, the path stretch factors of NoGeo, BVR, LCR, VPCR and HDR are 1.23, 1.33, 1.15, 1.17 and 1.07. At the highest node density, the converged path stretch values for NoGeo, BVR, LCR are 1.27, 1.19, 1.12. The VPCR protocol simply utilizes the backbone tree for routing, thus obtains a much higher path stretch of 1.57 compared to other protocols. With the effective *hop distance* metric, HDR manages to maintain the lowest path stretch of 1.11, which is 12.6% lower than NoGeo, 6.7% lower than BVR and 29.3% lower than VPCR.

C. Communication Overhead

The communication overhead indicates the amount of traffic required for each packet delivery. If the number of forwards required for a packet delivery by a routing protocol is H_p and the number of forwards needed by the shortest path routing is H_s, the overhead is computed as $\lambda = \frac{H_p}{H_s}$. NoGeo, BVR and LCR rely on restricted flooding to guarantee packet delivery, which means the amount of traffic will grow exponentially during the recovery stage. By routing the packets along the tree branches, VPCR and HDR generate significantly less overhead traffic for recovery. The communication overhead displayed in Fig. 4(c) shows that the average overhead values for NoGeo, BVR, LCR, VPCR and HDR are 26.48, 2.35, 6.67, 1.57 and 1.12. The tree based recovery procedure used by HDR manages to maintain a considerably lower overhead for each packet delivery compared to flooding based techniques and the greedy forwarding procedure in HDR helps to reduce the number of recoveries further from VPCR.



(a) Packet Delivery Ratio

(b) Path Stretch Factor

(c) Communication Overhead

Fig. 4. Routing Performance Comparison of Greedy, NoGeo, BVR, LCR, VPCR and HDR

D. Effects from Various Number of Anchors

The number of anchor nodes plays a crucial role for the routing performance of all connectivity-based routing protocols. By putting more anchor nodes into the network, the greedy forwarding procedure has a higher probability of success and routing protocols can achieve lower path stretch. However, the amount of initialization overhead will also grow linearly with the increasing number of anchors. We evaluate the path stretch performance of HDR under 5, 10, 15 and 20 anchors. The path stretch values are plotted in Fig. 5, indicating that as the number of anchor nodes increases from 5 to 20, the path stretch values descend at a decreasing rate. The path stretch values with 5, 10, 15 and 20 anchors are 1.26, 1.12, 1.08 and 1.06. This implies that by using approximately 10 anchor nodes in a network with a maximum scale of 1000 nodes, HDR can achieve excellent routing efficiency with low network overhead for a wide range of node densities.

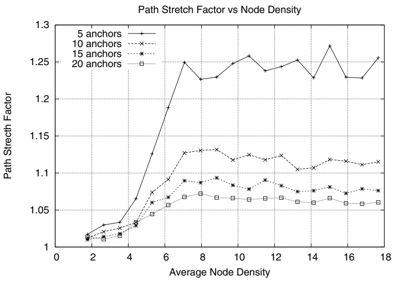


Fig. 5. HDR: Path Stretch vs. Number of Anchors

V. Conclusion

In this paper, we introduced a new connectivity-based routing protocol - Hop Distance Routing. By combining greedy forwarding on the *hop distance* metric with a tree-based recovery procedure, the HDR protocol can provide guaranteed packet delivery and lower path stretch with lower communication overhead, compared to other connectivity-based routing protocols evaluated. The recovery procedure of HDR does not rely on any localization methods, which makes HDR an efficient and desirable protocol for network deployments where localization methods are not effective and guaranteed packet delivery is required.

In the future work, we are going to evaluate the efficiency of different distance functions through mathematical analysis and explore alternative solutions for improving the accuracy of the hop-count based localization method. The current tree-based recovery algorithm may create hot spots at tree nodes located in the upper layer. A fully distributed recovery algorithm based on multiple trees or a complementary coordinate system should be created for better scalability and robustness.

Acknowledgement

The work described here is supported by the National University of Singapore research grant R252000265112 and R252000203112.

References

[1] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of the 6th ACM MobiCom*, pages 243–254, 2000.

[2] H. Frey and I. Stojmenović. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In *Proc. of the 12th ACM MobiCom*, pages 390–401, 2006.

[3] B. Leong, B. Liskov, and R. Morris. Geographic routing without planarization. In *Proc. of the 3rd NSDI*, pages 339–352, 2006.

[4] Q. Cao and T. Abdelzaher. Scalable logical coordinates framework for routing in wireless sensor networks. *ACM Transactions on Sensor Networks*, 2(4):557–593, 2006.

[5] R. Fonseca, S. Ratnasamy, J. Zhao, and et al. Beacon vector routing: Scalable point-to-point routing in wireless sensornets. In *Proc. of the 2nd NSDI*, volume 2, pages 329–342, 2005.

[6] J. Newsome and D. Song. GEM: Graph embedding for routing and data-centric storage in sensor networks without geographic information. In *Proc. of the 1st SenSys*, pages 76–88, 2003.

[7] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proc. of the 9th ACM MobiCom*, pages 96–108, 2003.

[8] Yao Zhao, Bo Li, Qian Zhang, Yan Chen, and Wenwu Zhu. Efficient hop id based routing for sparse ad hoc networks. In *Proc. of the 13th IEEE ICNP*, pages 179–190, 2005.

[9] A. Caruso, S. Chessa, S. De, and A. Urpi. GPS free coordinate assignment and routing in wireless sensor networks. In *Proc. of the 24th IEEE Infocom*, volume 1, pages 150–160, 2005.

[10] A. Nguyen, N. Milosavljević, and et al. Landmark selection and greedy landmark-descent routing for sensor networks. In *Proc. of the IEEE Infocom*, pages 661–669, 2007.

[11] K. Liu and N. A. Ghazaleh. Aligned virtual coordinates for greedy routing in wsns. In *Proc. of the 3rd IEEE MASS*, pages 377–386, 2006.

[12] M. Wong and D. Aksoy. Relative accuracy based location estimation in wireless ad hoc sensor networks. In *Proc. of the IEEE ICC*, 2007.

[13] M. J. Tsai, H. Y. Yang, and W. Q. Huang. Axis-based virtual coordinate assignment protocol and delivery-guaranteed routing protocol in wireless sensor networks. In *Proc. of the 26th IEEE Infocom*, pages 2234–2242, 2007.