IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

An Efficient Geometric Routing using Convex Hull Tree Information in Geographic Protocol

T. Nivetha[1], S.R. Lavanya[2]

Research Scholar, Dept of Computer Science, Sri Ramakrishna of Arts and Science for Women, Coimbatore, India[1]

Assistant Professor, Dept of Computer Science, Sri Ramakrishna of Arts and Science for Women, Coimbatore, India[2]

Abstract: Geographic routing is of interest for sensor networks because a point-to-point primitive is an important building block for data-centric applications. Present a new geographic routing algorithm, Improved Greedy Distributed Spanning Tree Routing (GDSTR), which finds shorter routes and generates less maintenance traffic than previous algorithms. While geographic routing potentially scales well, it faces the problem of what to do at local dead ends where greedy forwarding fails. Existing geographic routing algorithms handle dead ends by planar zing the node connectivity graph and then using the right-hand rule to route around the resulting faces. The proposed system the definition of a new kind of spanning tree, which called hull tree, for use in networks where each node has an assigned coordinate. A hull tree is a spanning tree where each node has an associated convex hull that contains within it the locations of all its descendant nodes in the tree. Hull trees provide a way of aggregating location information and they are built by aggregating convex hull information up the tree. This information is used in routing to avoid paths that will not be productive; instead we are able to traverse a significantly reduced sub tree, consisting of only the nodes with convex hulls that contain the destination point uses new kind of spanning tree, which it calls hull tree, for use in networks where each node has an assigned coordinate. The experimental result show the best performance to reduce routing overhead and less delay compare with existing routing protocol.

Keywords: Geographic Routing, Greedy Distributed Spanning Tree Routing, Hull Tree, Routing protocol.

I. INTRODUCTION

In Geographic routing, each node can determine its own location and that the source is aware of the location of the destination. With this information, a message can be routed to the destination without knowledge of the network topology or a prior route discovery. Different geographic routing protocols have different requirements on routing metric designs to ensure proper operation. Wrong type of routing metric leads to unexpected results such as routing loops and unreachable nodes. Route selection in some routing algorithm is done by selection on multiple metric by combining them in single metric. Several routing metrics are used to achieve efficient routing in various geographic routing protocols. Some problems are identified among the routing protocols namely, Greedy routing is simple and it does not provide delivery guarantee, On the other hand MFF routing provides delivery guarantee but is complicated and may create very inefficient path and finally Cost for planarization and unavailability of location information are major issues in deployment of geographic routing.

GDSTR (Greedy Distributed Spanning Tree Routing) switches to routing on a spanning tree instead of a planar graph when packets end up at dead ends during greedy forwarding. To choose a direction on the tree that is most likely to make progress towards the destination, each GDSTR node maintains a summary of the area covered by the sub tree below each of its tree neighbours using convex hulls. This distributed data structure is called a hull tree. GDSTR not only requires an order of magnitude less bandwidth to maintain these hull trees, it often achieves better routing performance than other planarization-based geographic routing algorithms.

A related and somewhat subtler source of difficulty was that these algorithms also assumed that nodes knew their radio ranges and locations accurately. A recent empirical study has found that the communication ranges of wireless networks are highly dependent on the environment and may be highly irregular. Errors in the localization of the nodes can also cause planarization to fail. A major breakthrough was made by Kim et al. in developing the Cross-Link Detection Protocol (CLDP), which produces a sub graph on which face-routing-based algorithms are guaranteed to work correctly. Their key insight is that starting from a connected graph nodes can independently probe each of their links using a right-hand rule to determine if the link crosses another link in the network. CLDP uses a two-phase locking protocol to ensure that no more than one link is removed at any time from any given face; in this way it guarantees that the removal of a crossed link will not disconnect the network. While CLDP is able to planarians an arbitrary graph, every single link in the network has to be probed multiple times, and has a high cost.

First proposed Face Routing, they only described a routing mechanism and did not propose a method for constructing

DOI 10.17148/IJARCCE.2017.6230

planar sub graphs from an existing network connectivity graph. Bose et al. proposed using the Gabriel Graph (GG) as the planar sub graph, while Karp and Kung suggested using the Relative Neighbourhood Graph (RNG) as well. There are well-established algorithms that allow computing both the Gabriel Graph and Relative Neighbourhood Graph in a distributed way when nodes have only local information. The main drawback of these algorithms is that they depend on the Unit Disk Graph assumption, which unfortunately has been shown not to hold practical radio networks. Other distributed algorithms that produced planar spanners on UDG graphs have also been proposed. These include the Localized Delaunay Graph and the Restricted Delaunay Graph.

The problem with such a trap array approach, however, is that it is unlikely to route efficiently: the approach can guarantee that a packet will be delivered in no more than 2n- 3 hops, but need to do much better than that. No planarization techniques to apply for avoid the high traffic link.

Geographic routing is of interest for sensor networks because a point-to-point primitive is an important building block for data-centric applications. Present a new geographic routing algorithm, Improved Greedy Distributed Spanning Tree Routing (GDSTR), which finds shorter routes and generates less maintenance traffic than previous algorithms. While geographic routing potentially scales well, it faces the problem of what to do at local dead ends where greedy forwarding fails. Existing geographic routing algorithms handle dead ends by planar zing the node connectivity graph and then using the right-hand rule to route around the resulting faces. The proposed system contribution of our work is the definition of a new kind of spanning tree, which we call hull tree, for use in networks where each node has an assigned coordinate. A hull tree is a spanning tree where each node has an associated convex hull that contains within it the locations of all its descendant nodes in the tree.

Hull trees provide a way of aggregating location information and they are built by aggregating convex hull information up the tree. This information is used in routing to avoid paths that will not be productive; instead we are able to traverse a significantly reduced sub tree, consisting of only the nodes with convex hulls that contain the destination point uses new kind of spanning tree, which we call hull tree, for use in networks where each node has an assigned coordinate. A hull tree is a spanning tree where each node has an associated convex hull that contains within it the locations of all its descendant nodes in the tree. Hull trees provide a way of aggregating location information and they are built by aggregating convex hull information up the tree. This information is used in routing to avoid paths that will not be productive; instead it is able to traverse a significantly reduced sub tree, consisting of only the nodes with convex hulls that contain the destination point.

II. RELATED WORK

Virtual Ring Routing (VRR), a new network routing protocol that occupies a unique point in the design space. VRR is inspired by overlay routing algorithms in Distributed Hash Tables (DHTs) but it does not rely on an underlying network routing protocol. It is implemented directly on top of the link layer. VRR provides both traditional point-to-point network routing and DHT routing to the node responsible for a hash table key. VRR can be used with any link layer technology but this paper describes a design and several implementations of VRR that are tuned for wireless networks. We evaluate the performance of VRR using simulations and measurements from a sensor network and an 802.11a test bed. The experimental results show that VRR provides robust performance across a wide range of environments and work- loads. It performs comparably to, or better than, the best wireless routing protocol in each experiment. VRR performs well because of its unique features: it does not require network flooding or translation between fixed identifiers and location-dependent addresses.

The scalable technique for point-to-point routing in wireless sensor nets. This method, called Beacon Vector Routing (BVR), assigns coordinates to nodes based on the vector of hop count distances to a small set of beacons, and then defines a distance metric on these coordinates. BVR routes packets greedily, forwarding to the next hop that is the closest (according to this beacon vector distance metric) to the destination. We evaluate this approach through a combination of high-level simulation to investigate scaling and design trade-offs, and a prototype implementation over real test beds as a necessary reality check.

Greedy Perimeter Stateless Routing (GPSR), a novel routing protocol for wireless datagram networks that uses the positions of routers and a packet’s destination to make packet forwarding decisions. GPSR makes greedy forwarding decisions using only information about a router’s immediate neighbours in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region. By keeping state only about the local topology, GPSR scales better in per-router state than shortest-path and ad-hoc routing protocols as the number of network destinations increases.
Under mobility’s frequent topology changes, GPSR can use local topology information to find correct new routes quickly. We describe the GPSR protocol, and use extensive simulation of mobile wireless networks to compare its performance with that of Dynamic Source Routing. Our simulations demonstrate GPSR’s scalability on densely deployed wireless networks. GPSR, packets are marked by their originator with their destinations’ locations. As a result, a forwarding node can make a locally optimal, greedy choice in choosing a packet’s next hop.

DOI 10.17148/IJARCCE.2017.6230

For many years, scalable routing for wireless communication systems was a compelling but elusive goal. Recently, several routing algorithms that exploit geographic information (e.g., GPSR) have been proposed to achieve this goal. These algorithms refer to nodes by their location, not address, and use those coordinates to route greedily, when possible, towards the destination. However, there are many situations where location information is not available at the nodes, and so geographic methods cannot be used.

In parallel to the efforts on compact routing, another line of research, namely geometric routing (or geo-routing), has considered specialized routing methods for wireless networks by taking advantage of the nodes' geometric positions. In this approach, it is assumed that every node knows its own position, and the source of a message knows the position of the destination (through for example a distributed hash table). The algorithm forwards packets in a greedy manner by selecting next hops that are progressively closer to the destination. When the packet encounters a local minimum (LM) and cannot move forward, a recovery scheme is executed. The defining characteristic of geo-routing is that its performance depends on the network's geometric properties. In geometrically simple environments, this approach produces shortest paths with state size independent of network size, offering limitless scalability. In other environments, however, it may have significantly increased stretch or state overhead.

III.PROPOSED APPROACH

GPSR performs greedy forwarding whenever possible: Every node tries to find a neighbour node that makes the largest positive progress in terms of geometric distance toward the target. If a node cannot find a neighbour node that is closer than itself to the target, it enters a local minimum, at which point the algorithm runs a face routing procedure. GPSR also runs an independent algorithm to planarize the original network, so that no two-links cross each other, and the plane is divided into a set of faces. The face routing procedure guides the packet to move around the current face, following a right-hand rule (in analogy to following the right-hand wall in a maze), until it finds a node that can bring the packet forward compared to the latest local minimum point. It is proved that under the UDG radio model, GPSR guarantees delivery.

The key idea in GDSTR+ is to augment GDSTR with two forests of local trees and an additional greedy-hull forwarding mode. In GDSTR+, a node will first attempt to forward a packet greedily as before. If greedy forwarding fails, it will switch to the new greedy-hull forwarding mode by using the information contained in the convex hulls of a local hull tree. By local, we mean that the tree contains only the nodes in a limited locality. Since correctness cannot be guaranteed, forwarding can sometimes fail using the local tree and in such a case, a node will switch to forwarding on one of the two original global hull trees, which is guaranteed to succeed.

A. Greedy Distributed Spanning Tree Routing

Greedy forwarding’s great advantage is its reliance only on knowledge of the forwarding node’s immediate neighbours. The state required is negligible and dependent on the density of nodes in the wireless network, not the total number of destinations in the network. 1 On networks where multi-hop routing is useful, the number of neighbours within a node’s radio range must be substantially less than the total number of nodes in the network. The position a node associates with a neighbour becomes less current between beacons as that neighbour moves. The accuracy of the set of neighbours also decreases; old neighbours may leave and new neighbours may enter radio range. For these reasons, the correct choice of beaconing interval to keep nodes’ neighbour tables current depends on the rate of mobility in the network and range of nodes’ radios. We show the effect of this interval on GPSR’s performance in our simulation results. We note that keeping current topological state for a one-hop radius about a router is the minimum required to do any routing; no useful forwarding decision can be made without knowledge of the topology one or more hops away.

B. Convex Hull Tree Creation

Each node in basic hull tree stores information about the convex hulls that contain the coordinates of all the nodes in sub trees associated with each of its child nodes. The convex hull information is aggregated up the tree. Each node computes its convex hull from the union of its coordinate and the points on the convex hulls of all its child nodes, and this information is communicated to the parent node. Consequently, the convex hull associated with the root node is the convex hull of the entire network and contains all the nodes in the network. The convex hull for a set of points is the minimal convex polygon that contains all the points; it is minimal because the convex hull will be contained in any convex polygon that contains the given points. The hull is represented as a set of points (its vertices), and this set could be arbitrarily.

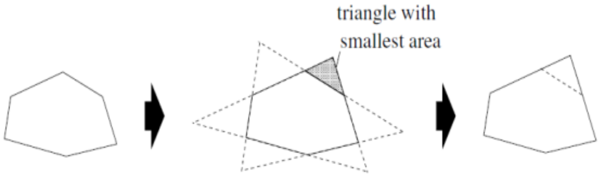

Fig 1.1 Procedure to reduce the size of a convex hull

To route packets on a hull tree, we forward a packet to child nodes that have a convex hull containing the destination. If none of the child nodes have convex hulls containing the destination, we know that the destination is not reachable down the tree, so we forward the packet up the tree.

DOI 10.17148/IJARCCE.2017.6230

C. Convex Hull with GDSTR Algorithm

In particular, because of the local tree building algorithm, we know that each local hull tree completely covers all the nodes in a given grid square. Hence, if the target region of a geocast message is completely contained in a grid square, we know that it will be broadcast correctly to all the required targets within the region. Because two forests are available, even if the target region of a geocast message is not completely contained within a grid square for one forest, it is likely to be contained within a grid square of the other. If indeed a suitable local tree cannot be found, correctness can be guaranteed by broadcasting on one of the global hull trees as before. Because the constraints for correctness are much stricter for planarization, it will in general require more effort to maintain a planar sub graph than a spanning tree. In fact, a distributed spanning tree has only two criteria for correctness:

1. Each node, except for the root node, has exactly one parent node.
2. Each node must be connected. We guarantee this by ensuring that every node has a common view of the root of the tree.

Both these conditions can be checked locally by a node by communicating only with immediate neighbours. On the other hand, the only known technique for detecting and eliminating non-planar edges in a connected graph requires non-local face traversals. The GDSTR routing algorithm will work correctly as long as we have a rooted spanning tree. In this section, we describe some algorithms that will produce rooted spanning trees. Given a hull tree with a specific root, GDSTR employs the following Minimal-Depth Spanning Tree algorithm. A node, n, chooses its parent node as follows:

Minimal-Depth Spanning Tree

Determine the set of neighbouring nodes that have minimal depth, i.e., are at the smallest number of hops from the root. If there is only one node in the set, choose that node as the parent.

- If there is more than one node in the set, choose the node that is closest in geometric distance to n as the parent. Closely related is the following Minimal-Path Spanning Tree:

Minimal-Path Spanning Tree

Determine the set of neighbouring nodes that have minimal path length to the root.

- If there is only one node in the set, choose that node as the parent.
- If there is more than one node in the set, choose the node that is closest in geometric distance to n as the parent.

These algorithms will produce minimal spanning trees (in terms of either path length or hops) rooted at external nodes. The expected advantage of these trees is that a packet is will be able to traverse the entire tree in a small number of hops or path length. The actual routing performance is related to D, the diameter of the network. The disadvantage of such trees is that when the network density is high, some intermediate nodes may end up with a large number of children. Since each child has an associated hull, the amount of state stored per node will therefore be proportional to network density, and not constant.

The results with local trees are marginally better than those with only two global trees. Surprisingly, the results with three and four global trees are worse. We suspect that the latter is due to an artefact in the experimental setup. The effective difference in the results is small: it translates to a difference in two or three packets for each geocast instance. The results for networks with obstacles. These results show that obstacles have a marginal effect on geocast performance. The fact that geocast with three global hull trees performs marginally worse than the rest is likely an artefact of the experimental set up, since the grid squares and target regions are squares that are aligned with the x and y axes. On the other hand, the rays that are used to choose the roots for the three global trees are not aligned in the same way. This is likely to have an effect on the orientation of the resulting hull trees.

Overall, geocast with local hull trees (GDSTR+) incurs 10% less overhead than geocast with only two global hull trees in sparse networks with large voids. All the variants seem to perform equally well in dense networks. These results also suggest that we can likely implement geocast using hull trees with no more than two times the minimum number of messages (since the Estimated Geocast Stretch is a loose upper bound).

D. Overall Process Of Algorithm

Step 1. Check for Geocast Mode: If p: mode = Geocast, follow step 6.

Step 2. Check Reached Broadcast Tree: If v has a child with a convex hull that intersects with R, follow step 5. Otherwise, follow step 3.

Step 3. Find Tree Mode: If p: mode = Find Tree:

- If v is the root node for p: Tree, algorithm terminates here.
- Otherwise, forward p to the parent node in P: Tree.

Step 4. GDSTR+ Routing: Route packet to destination t according to Algorithm. If packet is undeliverable, set p: mode: = Find Tree and follow step 3.

Step 5. Pick Hull Tree for Geocast:

- If R is contained in either of the grid squares of the local hull trees, set p: Tree as the local tree (in a grid square that contains R) with a root that is closest to the t.
- If the grid squares of the local hull trees do not completely contain R, set p: Tree as the global tree with a convex hull that contains R; if such a global tree does not exist, pick the global tree with a root that is closest to t. Follow step 6.

Step 6. Broadcast to Target Set: Determine target set B for message broadcast with respect to p: Tree according to the following rules:

DOI 10.17148/IJARCCE.2017.6230

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

- If p: mode = Geocast, the node from which geocast message was originally received is not to be included in set of targets.
- If p: Tree is a local tree, each neighbouring node that has an associated convex hull (from vs. perspective) that intersects R is added to the target set.
- If p: Tree is a global tree, each child node that has a convex hull that intersects R is added to the target set. If the convex hull of associated hull tree p: Tree fully contains R and none of the conflict hulls H intersects R, do not add the parent node to target set. Otherwise, add the parent node to the target set.
- If p: mode 6= Geocast, set p: mode: = Geocast. Broadcast p to all nodes in target set B.

The correctness of this algorithm follows from the correctness of the GDSTR geocast algorithm.

IV.EXPERIMENTAL RESULTS

This section evaluates the performance of the system can be evaluated by using following performance metrics. The performances of the proposed convex hull tree algorithm applied and compare the existing algorithm. To understand the effects of network density on routing performance and maintenance costs, we generated networks with 25 to 500 nodes randomly scattered over a 100x100 unit square. This process generated networks with average node degrees between 0.7 to 14.4. For each density, we generated 200 networks, and then routed 20,000 packets using each algorithm between randomly chosen pairs of source and destination nodes. The performance measurements presented are the average over the 200 times 20,000 data points. We also used these topologies to evaluate the effects of parameters like the number of hull trees and the value of r, the maximum size for the convex hulls.

TABLE I PERFORMANCE UNDER THE GDSTR MODEL WITH CONNECTION STRETCH

Protocol	Connection Stretch							
	10	20	30	40	50	60	70	80
GPSR	2.0	8.4	9.1	9.7	10.9	12.5	15.4	18.6
CGR	1.8	4.1	6.2	7.9	8.1	10.0	12.1	14.2
GDSTR	1.2	1.4	1.4	1.5	1.7	1.8	1.9	2.1

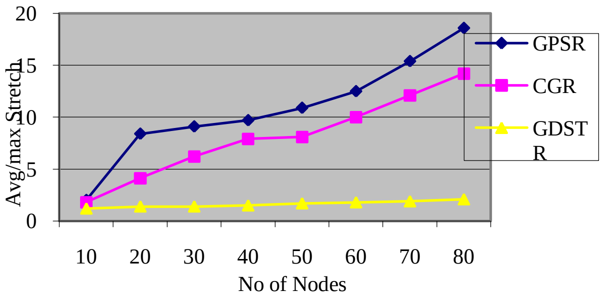

Fig. 1 Comparison of different connection Stretch with no of nodes

TABLE III PERFORMANCE UNDER THE GDSTR MODEL WITH AVG STATE PER NODE

Protocol	No of Nodes							
	10	20	30	40	50	60	70	80
GPSR	45	58	61	74	81	88	91	94
CGR	20	37	48	52	69	74	82	85
GDSTR	10	17	24	32	41	47	53	59

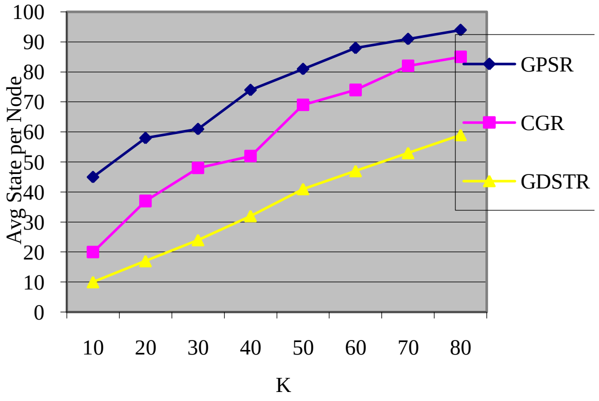

Fig. 2Comparison of different state per node with k

TABLE IIIII PERFORMANCE UNDER THE OBSTACLE MODEL WITH STRETCH

Protocol	No of Obstacle							
	2	4	6	8	10	12	14	16
GPSR	43	48	49	52	54	57	59	61
CGR	21	25	28	31	35	38	41	44
GDSTR	12	15	18	21	26	28	32	36

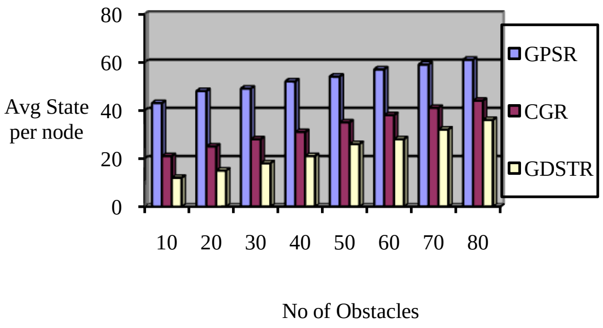

Fig. 3Comparison of different obstacles state

TABLE IVV PERFORMANCE UNDER DIFFERENT TRAFFIC LOAD

Protocol	No of Obstacle				
	2	4	8	16	32
GPSR	643	1043	2264	2866	3873
CGR	554	887	1056	1164	1257
GDSTR	321	534	755	1078	1135

DOI 10.17148/IJARCCE.2017.6230

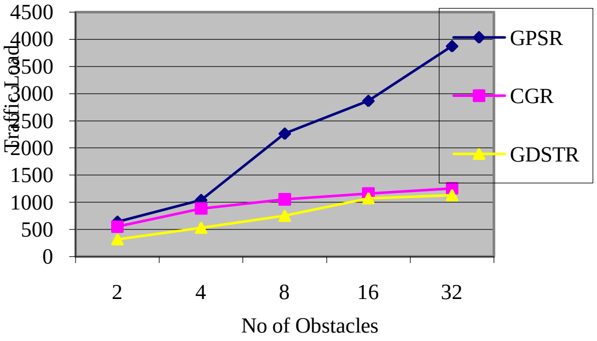

Fig. 4Comparison of different traffic load

TABLEVV PERFORMANCE TRANSMISSION MESSAGE COST

Proto-col	**No of Nodes**							
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**
GPSR	153	175	187	196	254	289	297	321
CGR	141	154	164	176	196	234	265	298
GDSTR	123	134	157	163	173	186	197	211

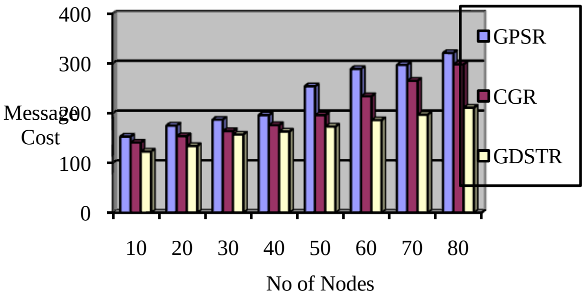

Fig. 5Comparison of different Message Cost

V. CONCLUSION

Improved GDSTR forwards packet greedily until a local minimum is encountered. When a packet ends up in a local minimum, IGDSTR uses hull trees to route the packet around the void in a deterministic way. A hull tree is a spanning tree where each node has an associated convex hull that contains within it the locations of all its descendant nodes in the sub tree rooted at the node. IGDSTR switches back to greedy forwarding once it finds a neighbor closer to the destination than the local minimum.

REFERENCES

[1] J. Bruck, J. Gao, and A. Jiang, “MAP: Medial axis based geometric routing in sensor networks,” in Proc. MobiCom, 2005, pp. 88–102.

[2] M. Caesar, M. Castro, E. Nightingale, G. O'Shea, and A. Rowstron, “Virtual ring routing: network routing inspired by DHTs,” in Proc. ACM SIGCOMM, 2006, pp. 351–362.

[3] L. Cowen, “Compact routing with minimum stretch,” J. Algor., vol. 38, no. 1, pp. 170–183, 2001.

[4] T. Eilam, C. Gavoille, and D. Peleg, “Compact routing schemes with low stretch factor,” in Proc. ACM PODC, 1998, pp. 11–20.

[5] R. Fonseca et al., “Beacon vector routing: Scalable point-to-point routing in wireless sensor nets,” in Proc. NSDI, 2005, vol. 2, pp. 329–342.

[6] B. Karp and H. T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” in Proc. MobiCom, 2000, pp. 243–254.

[7] R. Kleinberg, “Geographic routing using hyperbolic space,” in Proc. IEEE INFOCOM, 2007, pp. 1902–1909.

[8] S. S. Lam and C. Qian, “Geographic routing in d-dimensional spaces with guaranteed delivery and low stretch,” in Proc. SIGMETRICS, 2011, pp. 217–228

[9] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shankar, and I. Stoica, “Geographic routing without location information,” in Proc. MobiCom, 2003, pp. 96–108.

[10] G. Tan, M. Bertie, and A-M. Kermarrec, “Convex partition of sensor networks and its use in virtual coordinate geographic routing,” in Proc. IEEE INFOCOM, 2009, pp. 1746–1754.

DOI 10.17148/IJARCCE.2017.6230