Almost Delaunay Triangulation Routing for Energy Efficient Wireless Sensor Networks

Md. Bahlul Haider
Supervisor: Prof. Kokichi Sugihara

1. Introduction

The wireless sensor network is a new and promising style of sensing data from various kinds of environments, and hence is one of the most important research areas with many practical applications such as pollution monitoring, wildlife tracking, traffic monitoring etc. In particular, energy efficiency is the most important aspect in wireless sensor networks. Energy consumption could be reduced by an efficient routing topology of the sensor network. Further energy consumption could be reduced by taking localized routing decision. Here, we propose an energy efficient rouging for wireless sensor networks called almost Delaunay triangulation routing approach, which meets both the energy efficiency and localized routing criteria.

We propose a Delaunay triangulation based routing approach [1] for wireless sensor networks. We name it almost Delaunay triangulation routing. The almost Delaunay triangulation graph is a planar graph that contains all the Delaunay edges shorter than a threshold, and can be constructed and used efficiently by local communication.

2. Generation of the almost Delaunay triangulation

Suppose that many sensor nodes are distributed randomly in a certain domain. Each sensor node tries to recognize its neighbors by local communication. The relative neighbor relation is represented by a geometric graph called an almost Delaunay triangulation. This can be done by the next algorithm, where E_{MSTMAX} is the longest edge in the minimum spanning tree and C is some constant. $C \geq 1$ will ensure that the graph will be connected.

Step 1:

(a) Send a *hello* message within the distance $C \times E_{\mathrm{MSTMAX}}$ and wait for the reply.

(b) Upon receiving the *hello* message, acknowledge the sender alone with the location information of the current node.

(c) Update the neighborhood list according to the acknowledgements of the *hello* messages.

Step 2:

Calculate the Voronoi polygon formed by the perpendicular bisectors of each line segment (u,v_i), where v_i is the neighbor of u. Then connect u with the neighbor nodes that has common edge in the Voronoi polygon. Let this graph be G_1.

Step 3:

For each pair of two edges in G_1,whose distance is within the transmission range, if two edges cross each other, then remove the edge that violates local Delaunay property.

The resulting graph G is our almost Delaunay triangulation.

3. Complexity

Step 1 in the algorithm can be performed in O(n) time. It takes O($n$log$n$) time to generate the Voronoi polygon in step 2. To eliminate the edge crossing in step 3, it takes O(m^2), where m is the number of edges in graph G_1.

However, we need to make sure that the graph is connected. For this reason, we have to approximate the value of E_{MSTMAX} properly. If the estimated value of E_{MSTMAX} is smaller and the resulting graph is disconnected, the above method does not work. We constructed an additional algorithm by which, we check the connectivity, and if disconnected, we raise the energy for local communication and try to make an almost Delaunay triangulation again. We repeat this process until we get a connected graph. However, this is time and energy consuming, and hence good estimation of E_{MSTMAX} is important.

4. Routing in almost Delaunay triangulation

In each round of environment monitoring, one sensor node is chosen as the destination by using a token based head selection approach, and other sensors, if they sense information to report, send the report to the destination by finding a route by local communication. For this purpose, compass routing along with perimeter routing [2] and face routing are used. In compass routing, we always choose the next neighbor that is closest to the direction of the destination and the distance to the destination is less than that of the current node. If no such neighbor exists, we switch from compass routing to perimeter routing. In the case of perimeter routing, we choose either the right-hand side node or the left-hand side node to bypass the hole. When we move closer to the destination than the node where we were stuck for compass routing, we then again switch to compass routing. In perimeter routing mode we may come back to the starting point without moving closer to the destination. This may happen, because our underlying graph contains some edges that are not present in the Delaunay triangulation. In that case, we switch from perimeter routing to face routing. In face routing, we try to go to the closer face to the destination. Thus, we eventually reach the destination. Figure 1 shows an example of routing in the almost

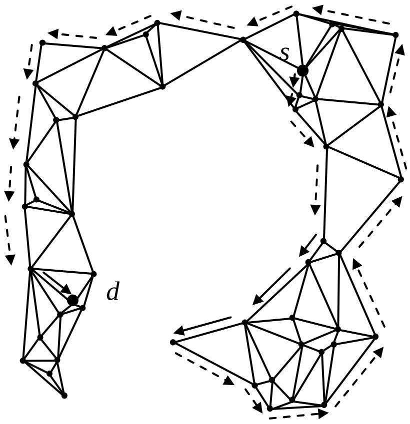

Figure 1. Routing in almost Delaunay triangulation.

Delaunay triangulation. In Figure 1, *s* represents the source node and *d* represents the destination node. The solid-line arrows represent the compass routing, whereas the broken-line arrows represent the perimeter routing. In perimeter routing mode, we may cross the line *c-d* without going closer to the destination. The closest such edge to the destination on the perimeter is called the critical edge. In case of face routing, we move to the other face when we reach the critical edge. Figure 2 shows the flowchart of our routing algorithm. We start at the compass routing phase and may reach the destination from the compass routing phase or from the perimeter routing phase.

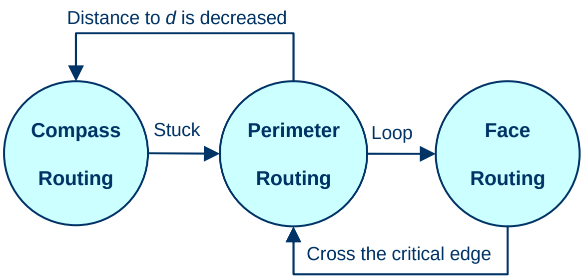

Figure 2. Flowchart of routing.

5. Guaranteed delivery

Our proposed routing technique guarantees the delivery in the almost Delaunay triangulation. In the compass routing mode, at each step we move closer to the destination because we taking greedy decision. In the case of perimeter routing, we also move closer to the destination if we have nodes that are closer to the destination on the perimeter. If there is no node closer to the destination on the perimeter then we switch to face routing. In the case of face routing, we change the face in the critical edge and switch to perimeter routing. When we change the face, we always go to a face that is closer to the destination on the line *c-d*. As there are finite number of faces and nodes in the almost Delaunay triangulation graph, we always reach the destination in finite steps. Compass routing and face routing are sufficient to guarantee the delivery in almost Delaunay triangulation. However, we have used perimeter routing to improve the path quality.

6. Experimental result

Face routing is rarely used in almost Delaunay triangulation. Experimental result shows that the average path length found by using our routing technique is about 1.5 times longer than the average minimum cost path in the routing trees if there is an open hole. If the hole is closed or there is no hole at all then the average path length in our approach is about 1.3 times longer than the average minimum cost path.

7. Conclusion

We have presented a simple but efficient routing approach for wireless sensor networks. The new concept called the almost Delaunay triangulation is introduced and used in our approach which is very much easy to generate by the nodes in a localized manner. In our routing, we have combined the compass routing, the perimeter routing and the face routing, by which we guarantee that we always reach the destination. Also the quality of the resulting route is satisfactory.

Other routing approaches could be used efficiently in the same graph, as our underlying graph is a planar graph. In our proposed almost Delaunay triangulation routing in sensor network, we have used only one cluster. Further research could be continued to implement the same algorithm with multiple clusters. Multiple clusters could use the energy of the sensor nodes more efficiently. Further work could be done to find the optimal value of *C* in terms of sensing area *A* and number *n* of nodes.

References

[1] M. B. Haider and K. Sugihara, An Energy Efficient Routing Approach for Wireless Sensor Networks, *Proceedings of 9th International Conference on Computing CORE*, Mexico City, Mexico, 2008, pp. 261-276.

[2] B. Karp and H. T. Kung, GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking MobiCom'00*, ACM Press, 2000, pp. 243-254.