EEGFGR: an Energy-Efficient Greedy-Face Geographic Routing for Wireless Sensor Networks

Tao Zi-Jin[1], Wu Yi[1], Gong Zheng-Hu[1]

[1]School of Computer, National University of Defense Technology, Changsha 410073, China
{taozj888@163.com, wuyi@nudt.edu.cn, gzh@nudt.edu.cn}

Abstract. The geographic routing technology is good for the self-organizing and large-scale Wireless Sensor Networks(WSNs), and the energy-limited sensor nodes require the energy used for routing to be minimum. In this paper a new energy-efficient geographic routing algorithm is proposed and it is distributed and based on the geographic routing, the topology characteristics of the network and the wireless communication energy model. The algorithm is based on the planarized graph (GG) of the network, it deals with the routing void by saving more face neighbors in every node, and selects the most energy-efficient nexthop node by using the energy-saving technologies including the Minimum Energy One-hop Neighbor Path Selection and the Optimal Face-neighbor Selection. The theoretical analysis and simulations show that the algorithm is feasible and more energy-efficient than many existed geographic routing algorithms. In the end , the means of face information maintenance are proposed.

Keywords: geographic routing; minimum energy; face neighbors; routing algorithm; Wireless Sensor Networks(WSNs)

1 1 Introduction

Recent advances in micro-electro-mechanical systems (MEMS) and low power and highly integrated digital electronics have led to the development of micro sensors. Such sensors are generally equipped with data processing and communication capabilities. The Wireless Sensor Networks (WSNs) are composed of many sensor nodes , which are self-organizing and easy deployable. The routing techniques of the WSNs are the key of highly-efficient networking of these nodes. Now the Geographic Routing Technique based on the position of the sensor nodes is a feasible solution to the routing problem of WSNs, and is attracting much more attention.

The network graph G<V,E> can be planarized by the methods in [4]. In this paper the face neighbor information as well as the one-hop neighbor information is added to the node based on the face properties of the planar graph, and the node selects the the node based on the face properties of the planar graph, and the node selects the routing path and the nexthop by the wireless communication model and the position of the neighbor stored in itself. The theoretical analysis and simulation show that the

energy consumption expended by the data communication is greatly reduced by the added neighbor information which can be used to deal with the routing void. So the algorithm is very proper for the WSNs which are long-lifetime and of large volume of data to be transmitted.

2   Related Works

Most geographic routing algorithms belong to the type which uses only the face routing or the combination of greedy routing and face routing.

1) Compass Routing II[3]

Completed based on the face routing, every time the forwarding node chooses the face which is intersected the line formed by the source s and destination d , and traversing the face, if meeting the line (u,v) which is intersected by the line (s,t), the packet is routed around another face. This approach is very energy inefficient for the face traversing greatly increases the routing hops and many hops have no uses with respect to making the packet closer to the destination.

2) GPSR[4]

An early geographic routing algorithm combines the greedy and face routing. In this algorithm, the greedy mode is used if it is feasible, and if not the perimeter mode is used in which the packet traverses the face using the right-hand rule until the node which is closer to destination and the packet leave the local optimal node. GPSR is highly more energy-efficient than the Compass Routing II.

3) GOAFR+[5]

When meeting a void, the GOAFR+ selects a random forwarding direction to search, but it doesn't continuously forward the packet around a face, but uses an ellipse to restrict the search bound. When it meets the ellipse, it will search on the opposite direction.

4) GPVFR[6]

In GPVFR , the node stores some planar face neighbors which is within several hops and the number of the neighbors is adjustable. By the increased neighbor information the best forwarding direction will be chosen and the blindly search is avoided. The simulation shows it is more efficient that the GPSR and GOAFR+. The main disadvantage of the algorithm is neighbor number stored in the node is determined beforehand and randomly, and its performance can be improved more.

The above algorithms conquer the routing void with their own methods but their performance can be improved further and the energy metric has not been considered, so the routing path formed is not likely to be the most energy-efficient path. At the next section a new routing algorithm based on the greedy-face mode is proposed which not only considers the best way to overcome the routing void but also the energy factor at every forwarding phase. The analysis and simulations show that the energy efficiency of the algorithm exceeds the above algorithms.

3   Basic Idea and Model of EEGFGR

3.1   Basic Idea

As the above geographic routing algorithms, EEGFGR is also a distributive geographic routing algorithm in which the routing computation is completely dependent on the routing information stored in itself. EEGFGR added the face

neighbors[10] as the candidate nexthop, so the node will see more neighbors closer to the target and have more opportunities to select an better neighbor in a reasonable way.

EEGFGR combines the greedy routing, face routing and perimeter routing and at every phase the energy metric is considered at the same time. So in every step of the algorithm the minimum energy path is selected.

When the face routing is used, the node has two directions to forward the packet: clockwise and counter-clockwise. The minimum hops routing metric may not be the most energy-efficient because the distribution of the nodes may not be uniform.

When the node which is closer to the target than the current node can not be found yet by the face routing, the perimeter mode is used as the GPSR uses.

3.2   Model of EEGFGR

There are many algorithms which can be used to planarize the graph, in EEGFGR we use the the same means as in [8] (GG)and [9](RNG). The WSNs' topology is described as a Graph G<V,E,E′>, V is the node set ,E is the set of all the communication link , E′ is the set of the links which remains after planarization, $E' \subseteq E$. The communication link is represented as {v_1,v_2}, if {v_1,v_2} $\in$ E, then v_1,v_2 is a neighbor of v_2 and vice versa. If {v_1,v_2} $\in E'$, v_1 is a planar neighbor of v_2 and vice versa; if {v_1,v_2} $\in E - E'$, then v_1 is an non-planar neighbor of v2 and vice versa.

The face exploration usually uses the right-hand(left-hand) rule which can be represented as: if $v_1, v_2, v_3 \in V$, if the ray v_1v_2 rotate counter-clockwise around v1, when the ray meets the planar neighbor v_3, then right(v_1,v_2)=v_3. The concept face and face neighbor are given as following.

Definition 1: If F is a node sequence ($v_1,v_2,v_3,\ldots,v_{n-1},v_n$), if it satisfy the following condition , it will be called as a Face:

(1) {v_i,v_{i+1}} , {v_n,v_1} $\in E'$;

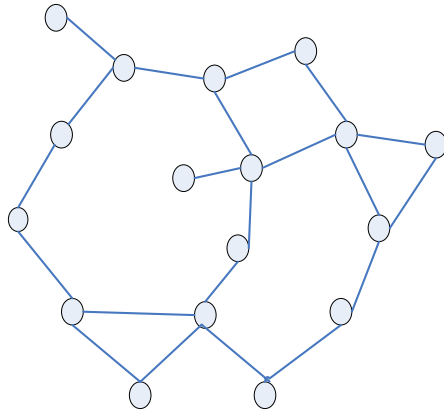(2) $right(v_{i+1},v_i)=v_{i+2}$, $right(v_n,v_{n-1})=v_1$, $right(v_1,v_n)=v_2$;$(1 \le i \le n-1)$。

Fig 1.   the sketch map of the face
In Fig1, (A,E,F),(F,K,J,I,Q，I,H,G,P,G,B,A),(D,C,N,L,G, B,A) are the faces.

Below, F is defined to be the set of all the faces in G.

Definition 2:

The face neighbors of the node v is a set of nodes that are contained in one face:

FN(v)={n| $\exists f \in F$, which satisfies $v \in f$ 且 $n \in f$ }

Definition 3:

The one-hop neighbors of node v is defined as :

HN(v)={n|{n,v}$\in E$ }-{v}

The routing void can be avoided by using the face neighbor information, so it will shorten the routing hops. As shown in Fig.1 if node L want to send a packet to the node G, if using the one-hop neighbor greedy , the node G will send the packet to node P, but node P sees that all the one-hop neighbors are further than itself , so the packet enter a routing void and the perimeter mode is used. Then the path is G-P-G-B-A-F-K-J-I-Q, which has 9 hops. But if we use the face neighbors, G will see that node I is the node closest to node Q, and the path is G-H-I-Q which has 3 hops.

Subjected to the paper space, the detailed face discovery process is omitted here.

For the planar graph, the number of face neighbors can be estimated by the face number and the average number of nodes in the faces. As demonstrated in [10], the average size of a face is $O(\frac{2n_e}{n_e - n_v + 2})$, the average node degree is $O(\frac{2n_e}{n_v})$, so the number of the face neighbors is $O(\frac{2n_e}{n_e - n_v + 2} \cdot \frac{2n_e}{n_v})$. To the planar graph the size of n_e and n_v are at the same level, so the number of the face neighbors is acceptable.

3.3 Energy Model of Wireless Communication

In this paper, we use the commonly accepted channel path loss model, $\rho = a \times \delta^\gamma + b$, $Pt = l\alpha_{11} + l\alpha_2(d)^2$, $P_r = l\alpha_{12}$

Pt is the power to send 1 bit of data to distance d , Pr is the power to receive 1bit of data. L is the packet length, α is a positive constant.

Each node, v, has a maximum communication range, range(v). We call the set of nodes within this range the neighborhood of v and denote as $N_v(\subseteq S)$.

$\alpha_{11} = 50 \times 10^{-9} J$, $\alpha_{12} = 50 \times 10^{-9} J$, $\alpha_2 = 10 \times 10^{-12} (\text{J/bit/m}^2)$ 。

4 Details of EEGFGR Algorithm

The EEGFGR is composed of three parts: the first one is Minimum Energy One-hop Neighbor Path Selection Algorithm(MEONPSA, the name of the algorithm is RouteInOneHopNeighbor for better memory), the second one is the Minimum Energy Face Path Selection Algorithm(MEFPA, the name of the algorithm is FacePath for better memory), and the EEGFGR full algorithm.

4.1 Minimum Energy One-hop Neighbor Path Selection Algorithm (MEONPSA)

The algorithm is used to select a minimum energy path to the neighbor which is within the node's radio range, i.e., to find a best path to the one-hop neighbor, instead of sending the packet directly to the node.

Table.1 MEONPSA

RouteInOneHopNeighbor (d)：
1. *if d is very close to s, then d is select to be the nexthop; return;*
2. *The radio range of s is R, its neighborhood set is the set of neighbors within the range and is represented as* N_{R_s} *.In order to limit the search range, a circle is constructed with the edge (s,d) as the diameter. The neighbors which is within the circle is represented as* N_{r_s} *, and for* $\forall v \in N_{r_s}$ *, we have* $(s,v)^2 + (v,d)^2 \leq (s,d)^2$ *, and if* $v_1, v_2 \in N_{r_s}, (v_1, v_2) \leq R$ *, i.e. there is an edge (v_1,v_2), and the set of the type of edges is represented as E_s.*
3. *For every edge* $e \in E_s$ *, label* l_e *represents the energy cost by the communication between two nodes.For any edge (v_1,v_2), the distance between v_1 and v_2 is represented as $d(v_1,v_2)$ and* $l_e \propto d(v_1,v_2)^2$ *.*
4. *In the neighborhood graph* $NGraph_s = (N_{r_s}, E_s, l_e)$ *, we set s to be the source, d to be the destination , and the edge e's weight to be l_e , run the Shortest Path Routing Algorithm(Dijkstra Algorithm)*
5. *Record the shortest path after the algorithm completes as $P_{s,d}=(s,v_1,v_2,...v_{n-1},d)$, the path is the minimum energy path.*
6. *Return the nexthop node v_1 in $P_{s,d}$ as MinE_nexthop, $P_{s,d}$ as MinE_Path。*

4.2   Dynamic Subdestination Adjust(Dyna_adjust)

After the node has got the minimum energy one-hop path with RouteInOneHopNeighbor, it send the packet to nexthop v_1, v_1 has two patterns to deal with the packet: one is to strictly send the packet to the next node which is denoted in $P_{s,d}$, this way is called Force mode; the other is v_1 recalculates a new minimum energy path and reselects a new subdestination based on its own neighborhood information, this way is called Dynamic Subdestination Adjust(Dyna_adjust) mode.

The Dyna_adjust will increase the energy efficiency of the routing algorithm, for every routing step the amount of energy consumed is less than that of the previous node calculates. But the Dyna_adjust may cause routing loop. But we have the follow theorem:

Theorem 1. If the Dyna_adjust satisfies $P_{s,d} \cap P_{s,d}' = \phi$, then the greedy algorithm with this method will not result in a routing loop.

PROOF.   If the Dyna_adjust will cause a routing loop, for $P_{s,d} \cap P_{s,d}' = \phi$, so the loop can only be caused after running the Dyna_adjust 2 or more times, and after running the algorithm many times, the routing path may have the same node. But in the greedy mode, the new subdestination will be closer than the previous one to the target and every node in the minimum energy path will be closer than the current node to the target, so these nodes must not have duplicate nodes. This contradicts the assumption.

4.3   Minimum Energy Face Path Selection Algorithm(MEFPSA)

When the packet is in the face routing mode, the minimum energy face path must be calculated when the packet traverses the face.

There are two ways for the node to select a nexthop from the face neighbors.

Mode 1: (The Most Greedy Mode, all-face-greedy)

The current node selects the face neighbor which is closest to the destination in all the face neighbors as the subtarget node.

Mode 2: (the face which is in the same direction with the line (u,t) fisrt, dir-face-greedy)

In this mode, the node u make a virtual line (u,t), if the face is intersected by the line, the face is called the face in the same direction(dir-face).If there exists that type of faces, the current node first selects the node which is closest to the destination in that face. If there are many that type of faces, then the first face that is intersected by the line (u,t) is selected.

It can be demonstrated that in GG(or RNG) graph, the dir-face must have node which is closer to the target than current node , and it is shown in below.
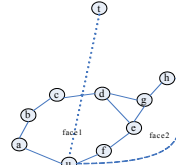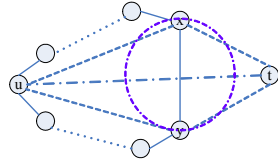

Fig. 2 The face which intersects the line (u,t) must has the node which is closer to the target

Fig. 3 The faces with and without the same direction with line (u,t) than the current node.

Theorem 2. In GG graph, if there is a face which is intersected by the line (u,t), then that face must have a node which is closer than the current node to the target t.

PROOF: (Omitted)

The reason to choose the dir-face is that generally the cost of traversing the face is high, so the face traversing should be finished as soon as possible. If the face is non-dir face, the face deviates from the target, even if the face has a better neighbor, the cost of traversing that face is higher than that of the dir-face. The dir-face example is shown in Figure 6, where node h is closer to t than the node d but the cost to get to node h is much higher than the cost to get to d.

Table 2.   MEFPSA

FacePath-all(v):

the node v constructs the face set which contains the node v, $FaceContain(v)$, $FaceContain(v) \subseteq Face(u)$, For each edge $e \in f$, $f \in FaceContain(v)$, computes the energy cost of every edge e, denoted as l_e;

For each face $f \in FaceContain(v)$, compute the energy cost Ep_i of every path P_i from u to v in f ,

$Ep_i = \sum_{j \in path_i} e_j$ ， if $Ep_k \leq (\forall i)Ep_i, i = 1, 2, ...n$ ， then the minimum energy face path MinEFacePath is P_k.

If we adopt the dir-face first mode, then in the above algorithm we set the *FaceContain(v)* only contain the nodes in the set of the dir-face.

4.4   Detailed Description of EEGFGR

4.4.1   Symbols used in EEGFGR

target: the destination node

pack：the packet received

e: in the Perimeter mode,   the first edge that the packet enter the face, it is used to check whether the destination is unreachable.

mode：the mode that the packet is in，3 value is :   EEGFGR-GREEDY, EEGFGR-FACE, EEGFGR-PERIMETER，the initial value is EEGFGR-GREEDY;

n_1: the node which is closest to target   in the one-hop neighborhood set

n_2: the node which is closest to the target in the face neighbors

goal: the goal which EEGFGR-FACE mode want to arrive

path：an array which elements are a lot of nodes, the index begins from 0

now：the initial value is 0

Neigh(u): the set that is composed of node u's one-hop neighbors

Face(u): the faces which are stored in node u;

FaceNeigh(u): the node set which is composed of the face neighbors of node u

  node(f)：the node set which is composed of the nodes in face f, and edge(f) is used to represent the edges in face f;

the distance between node u and node v is represented as d(u,v), the distance between the node u and the face f is represented as d(u,f)=min(a,v), ($v \in node(f)$), and set the v_{uf} to be the node closest to the node u in the face f.

If $v_1, v_2 \in node(f)$, we define the Face distance of v_1,v_2, $D(v_1, v_2, f)$ to be the length of the minimum energy path in face f ;

Face_Subgoal: the subdestination which the node u used to traverse the face;

MinEFacePath: the minimum energy face path $(fv_0,fv_1,…,fv_n)$ which is computed by the FacePath();

RouteInOneHopNeighborNexthop: the nexthop node which is computed by RouteInOneHopNeighbor() algorithm

RouteInOneHopNeighborPath: the minimum energy path $(v_0,v_1,v_2…,v_n)$ which is computed by RouteInOneHopNeighbor()

ForcedRoutingFlag：used to indicate the packet is in forced routing mode or dyna_adjust mode, 0 is for Dyna_adjust, 1 is for ForcedRouting.

4.4.2   Detailed Process of the Algorithm

1. if u ==d，then the routing is successful, return;

2. if ForcedRoutingFlag==1

If u != pack.RouteInOneHopNeighborNexthop   /*Forced Mode, the current node is not the last hop in the Forced Path   */

$u \in RouteInNeighPath$, *if u =v_i, then nexthop=v_{i+1};goto 9;*

else   /* is the last hop in the Forced Path */

 pack.ForcedRoutingFlag=0; goto 3；

3. if pack.status = EEGFGR-PERIMETER and (d(n_1,t) < d(s,t) or d(n_2,t) < d(s,t))，then pack.status:= EEGFGR-GREEDY; pack.ForcedRoutingFlag=0; goto 5;

4. if pack.mode = EEGFGR-FACE and u= pack.face_subgoal, then pack.mode:= EEGFGR-GREEDY；select step 5 or 6 or 7 based on the mode;

5. EEGFGR-GREEDY: /* the user may select 5.a or 5.b based on the greedy manner */

5.a)（one-hop neighbor first greedy）if d(n_1,t) < d(u,t), then n_1 is set to be subdestination node; pack.subdest=n_1, nexthop = RouteInOneHopNeighbor (n_1)，ForcedRoutingFlag =0;

Else, if d(n_2,t) < d(u,t)，/* the closer node is only situates in the face neighbors */，then select 5.a.1 or 5.a.2 according to the face greedy mode:

5.a.1) FacePath-all(n_2);

5.a.2) FacePath-dir(n_2);

Then we get the minimum energy path to n_2 MinEFacePath =(fv_0,fv_1,…,fv_k）, fv_k=n_2;

And the node u set:

ForcedRoutingFlag =1，

pack.mode=EEGFGR-FACE,

pack.path=MinEFacePath,

nexthop= RouteInOneHopNeighbor(MinEFacePath[1]),

and set the packet's path: pack.RouteInOneHopNeighborPath ;

pack.face_subgoal= n_2,   pack.subdest= MinEFacePath[1],

pack.hops=1; /* the first hop when traversing the face */

else /* can't find closer node */

set pack.mode = EEGFGR-PERIMETER;   goto 8;

5.b) (all-neighbor-greedy) if $d(n_1,t) \leq d(n_2,t)$ && $d(n_1,t) < d(u,t)$ set $n_1$to be subdestination:

pack.subdest=n₁，nexthop = RouteInOneHopNeighbor (n₁)，ForcedRoutingFlag =0;

else if $d(n_2,t) \leq d(n_1,t)$ *&&* $d(n_2,t) < d(u,t)$ *，/* there is a closer node in face*

neighbors *，then select 6.a or 6.b according to the face selection mode.;goto 6;

else /* there is no closer node in all neighbors */

pack.mode=EEGFGR-PERIMETER;   goto 8;

6.   Computing the minimum energy face path based on the face selection strategy, and record the result in the packet:

6.a) FacePath-all(n₂);

6.b) FacePath-dir(n₂);

and get a minimum energy face path to n₂ MinEFacePath =(fv₀,fv₁,...,fvₖ）, fvₖ=n₂;

the node u set the following values:

Pack.ForcedRoutingFlag ＝1，

pack.mode=EEGFGR-FACE,

pack.path=MinEFacePath,

nexthop= RouteInOneHopNeighbor (MinEFacePath[1]), and set pack.RouteInOneHopNeighborPath;

pack.face_subgoal= n₂， pack.subdest= MinEFacePath[1],

pack.hops=1; /* the first hop when traversing the face */

else /* can't find a closer node */ pack.mode=EEGFGR-PERIMETER; goto 8;

7. EEGFGR-FACE:

pack.hops = pack.hops+1,

pack.subgoal = pack.path[pack.hops];

nexthop= RouteInOneHopNeighbor (pack.subgoal), and set the pack.RouteInOneHopNeighborPath;

goto 8;

8.   EEGFGR-PERIMETER: is the same as the PERIMETER mode in GPSR, ForcedRoutingFlag ＝ 1，using the right-hand rule to compute the nexthop: pack.subdest = RightHand_nexthop, nexthop= RouteInOneHopNeighbor (pack.subdest)；and set the pack.RouteInOneHopNeighborPath;

9. send the pack to the nexthop。

5   SIMULATION RESULTS AND EVALUATION

5.1   Simulation Environment

In order to learn the performance of EEGFGR, simulations have been run on the ns2 simulator, and Monarch wireless communication model[12] is used. GPSR and GPVFR are programmed according to [4] and [6]. The number of simulated nodes is 200-500, the increment is 50, and the nodes are randomly deployed in a 500m*500m area. For every node density, ten graphs are produced, the radio range of the node is 50m.

We randomly set a node pair in each graph which has a distance of 300m, and the length of the packet is set to 250 bytes. The energy model described in 3.3 are used. The total energy consumption are recorded in the packet header, and the mean value of the 10 results are used as the final result. The WSNs are usually powered by the battery, we assume each node has an initial energy of 2 Joule.

5.2   Energy Used for Face Neighbor Discovery

Compared with GPSR and GPVFR, EEGFGR has a complete face neighbor discovery process. In order to illustrate the feasibility of this method, we record the energy cost used for the face exploration, i.e., the total energy used by all face exploration messages.

It is shown in figure 9 that under each network density the energy used for face exploration is very low and on the whole it is below 0.0043 Joule which is a very little part of the node's initial energy. The reason is when the network is sparse, and node

number of each face is high but the number of the face is high; when the network is dense, the number of the face is increased but the number of nodes in each face is decreased. At the same time, the face exploration is based on the one-hop neighbor information and avoids the blind broadcasts.
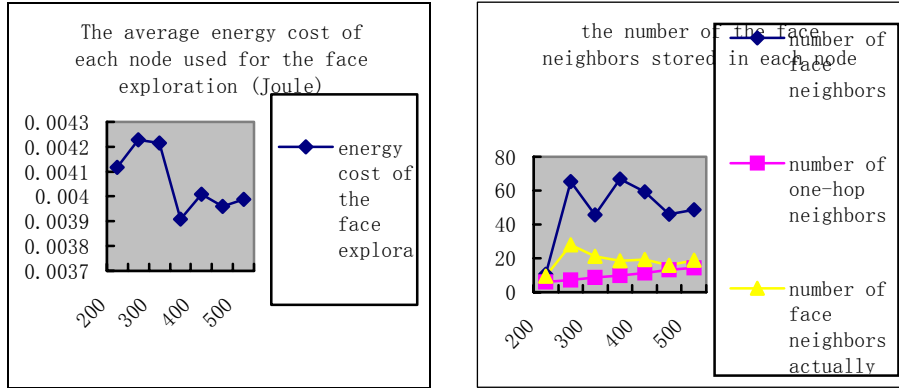

Figure 4. The average energy cost of each node used for the face exploration
Figure 5. the number of one-hop neighbors and face neighbors stored in each node

5.3   Number of Neighbors Stored in Each Node

In order to learn the storage cost for the face neighbors, we count the number of the face neighbors under each network density. In the topology, the outer face is composed of the nodes which locates at the network boundary and it has a great number nodes in it. But the outer face helps little to the routing, so the neighbors of the outer face may be discarded to decrease the storage cost.

It can be seen from the simulation that the number of face neighbors actually stored in each node is low and the peak value is reached when there are 250 nodes in the network.

5.4   Routing Performance of EEGFGR

The EEGFGR routing algorithm has many options, and in order to demonstrate the efficacy of the Minimum Energy One-hop Neighbor Path Selection strategy, we have used the strategy in GPSR. The following table illustrates the options of the algorithms.

Table 3. The Options of Each Algorithm

Name	Explanation
EEGFGR	MEONPSA(Dyna_adust), one-hop neighbor greedy, dir-face first
EEGFGR-II	MEONPSA(Dyna_adust), all-neighbor greedy, dir-face first
EEGFGR-III	MEONPSA(Dyna_adust), one-hop neighbor greedy
EEGFGR-IV	MEONPSA(Dyna_adust), all-neighbor greedy
GPSR-II	To improve the GPSR with MEONPSA, when the packet is in the greedy mode, the Dyna_adjust is used, and when the packet is in Perimeter mode, Forced mode is used

The experiments show that:

1. The energy cost is decreased as the network density is increased for each algorithm;

2. The energy efficiency of EEGFGR and its variations are much higher than GPSR, especially at the case the network is sparse or the network has large routing voids and in that case GPSR must frequently enter the Perimeter mode to blindly search a closer node to the target, but in EEGFGR the packet is usually in the Greedy mode(including

one-hop-neighbor greedy and all-neighbor greedy) because of the more neighbors stored, especially when the network has a large routing void its performance degrades little because of the help of face neighbors.
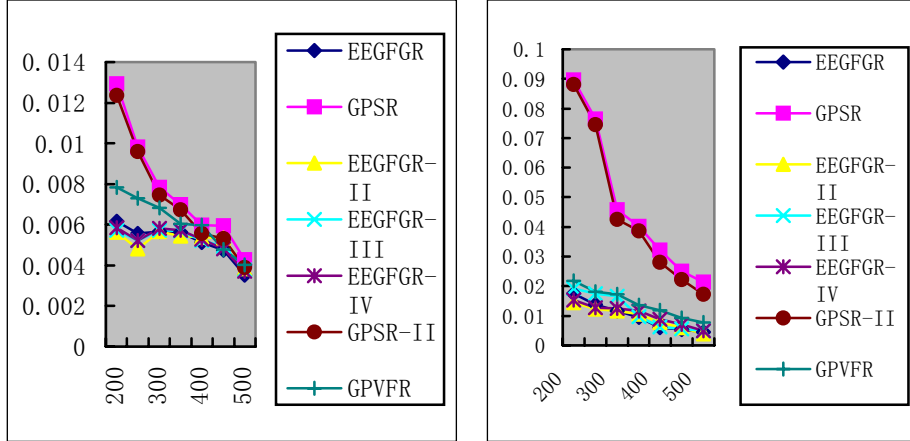

Figure 6. The energy cost (Joule) of transferring the data from the source to destination when the nodes are normally distributed in the network.

Figure 7. When the network has been added an artificial routing void (100m*200m), the total energy cost (Joule) used for transferring the data from the source to the destination.

3. When the network is sparse (the number of nodes is low, <=250), there more routing voids in the network and the all-neighbor greedy mode is superior than the one-hop-neighbor greedy mode; when the network is dense and the one-hop-neighbor greedy mode is better. The reason may be that when the network is sparse, the all-neighbor greedy mode may lead the node to see "further" than the one-hop greedy mode which is good for overcome the routing voids. When the network is dense, one-hop neighbor greedy is better because it leads to less face traversing manner which is energy expensive.

4. The Minimum Energy One-hop Neighbor Path Selection technology has some obvious effect in decreasing the energy consumption, which is shown in the graph when the GPSR has used that technology (GPSR-II), especially when the network is dense. About 5%-10% energy may be saved by the method.

5. Using the dir-face first technology is more energy efficient than not using that technology, which has an effect of decreasing the energy consumption by 3%-7%.

6. The energy efficiency of GPVFR is similar as the EEGFGR, but the number of neighbors stored in GPVFR is less than EEGFGR. When the network is sparse more neighbors are good for avoiding the routing voids, so the energy efficiency of EEGFGR is still higher than the GPVFR.

6 Conclusion

We have presented EEGFGR, a new energy-efficient geographic routing algorithm based on the greedy-face technology, which achieves small per-node routing state, small routing protocol message complexity and good delivery guarantees as previous promoted geographic routing algorithms. Theoretic analysis and experiments show it

has much higher energy-efficiency than previous ones, especially when the network has some large routing voids.

In the future work, we will check its performance in other planar graph, such as the UDel[13] graph, and improve the algorithm to fit for the practical communication environment[14].

References:

1. I.F. Akyildiz et al．Wireless sensor networks: a survey．Computer Networks，Vol.38，pp. 392-422．March 2002.

2. Chen Shigang,Fan Guangbin,Cui Jun-Hong, Avoid "Void" in Geographic Routing for Data Aggregation in Sensor Networks. International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC),Special Issue on Wireless Sensor Networks,2006,2(1)

3. E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in Proceedings of the 11th Canadian Conference on Computational Geometry, Vancouver, August 1999, pp. 51–54.

4. B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in Proceedings of the 6th annual international conference on Mobile computing and networking (Mobicom 2000), Boston, Massachusetts, August 2000, pp. 243–254.

5. Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger, Worst-case optimal and average-case efficient geometric ad-hoc routing. In Proceedings of the 4th ACM International Symposium on Mobile Computing and Networking (MobiHoc 2003), 2003.

6. Ben Leong, Sayan Mitra, and Barbara Liskov. Path vector face routing: Geographic routing with local face information. In Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP 2005), 2005.

7. HE Peng, LI Jian-Dong, CHEN Yan-Hui, ZHOU Lei, A Routing Algorithm for Ad Hoc Networks Based on Delaunay Triangulation, Journal of Software(Chinese), Vol.17, No.5, May 2006, pp.1149-1156

8. Prosenjit B, Luc D, William SE, David GK. On the spanning ratio of Gabriel graphs and beta-skeletons. In: Sergio R, ed. Proc. of the 5th Latin American Symp. on Theoretical Informatics. London: Springer-Verlag, 2002. 479.493.

9. TOUSSAINT, G. The relative neighborhood graph of a finite planar set, Pattern Recognition 12, 4 (1980), 261–268

10. Huang Qingfeng, Lu Chenyang, Roman G-C. Reliable Mobicast via Face-Aware Routing, INFOCOMM 2004

11. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660-670, October 2002.

12. THE CMU MONARCH GROUP. Wireless and mobility extensions to ns-2. http://www.monarch.cs.rice.edu.

13. Xiang-Yang Li, Gruia Calinescu, and Peng-Jun Wan, Distributed construction of a planar spanner and routing for ad hoc wireless networks. In Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM '02), volume 3, pages 1268–1277. IEEE Computer Society, June 23–27 2002.

14. Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In Proceedings of USENIX Symposium on Network Systems Design and Implementation, Boston, Massachusetts, USA, May 2005.