Energy Efficient and Fault Tolerant GPSR in Ad Hoc Wireless Network

Jyotsana Jaiswal and Pabitra Mohan Khilar

Department of Computer Science and Engineering,
National Institute of Technology Rourkela, India
jyotsanaa.jaiswal@gmail.com, pmkhilar@nitrkl.ac.in

Abstract. Routing in wireless network is a key research area which establishes path between source and destination node pairs. In this paper, we have designed and evaluated an energy-efficient and fault tolerant Greedy perimeter Stateless routing (EFGPSR) protocol for wireless Ad hoc network. The proposed protocol is divided into four phases: Fault testing phase, Planarization phase, Energy efficient greedy forwarding phase and Energy efficient perimeter forwarding phase. In fault testing phase, all nodes come to know about their fault free neighbours. Next is planarization phase which is subset of perimeter forwarding phase and can be done reactively or proactively. Next are energy efficient greedy forwarding and perimeter forwarding phases. Both these phases try to maintain balance between the metrics to choose the next hop (i.e. distance from destination in greedy forwarding phase and minimum angle w.r.t the line connecting the forwarding node and destination in perimeter forwarding phase) and selection of node having highest energy among the neighbouring node to extend network lifetime. Evaluation and comparison of GPSR and EFGPSR is done through NS-2 simulator. Simulation shows that EFGPSR performs better in terms of increasing the network lifetime, successful packet delivery ratio with insignificant increase in number of hop count.

Keywords: Wireless network, geographic routing, fault tolerance, GPSR, planarization.

1 Introduction

In recent years, wireless and mobile communication technologies have grown rapidly. As wireless network evolves there is a trend towards decentralized, deployable and self organizing networks.

Routing plays a vital role in establishing a communication links between various nodes. Since last few years, geographic routing (GR) protocols, also known as georouting or position based routing for wireless networks has gained a significant attention. The idea behind GR protocol is that the source node sends a packet to destination node using the geographic location of destination instead of using network address. The essential requirements of GR are a) each node should be capable of determining its own location (geographic coordinates) and

D. Nagamalai, E. Renault, M. Dhanushkodi (Eds.): CCSEIT 2011, CCIS 204, pp. 683–692, 2011.

b) source should be aware of the location of destination. The main advantage of GR over traditional routing protocol is that each node require to maintain only location of itself and its neighbours for its functioning. However, in traditional routing protocols for wireless networks (e.g. AODV [1], DSDV [2]), nodes usually have to keep significant amount of routing information. Greedy Perimeter Stateless Routing (GPSR) [3] is a well known and most commonly used position-based routing protocol for wireless networks. In GPSR, source includes the location information of destination in the header of every packet. If the destination is not directly reachable, the source starts with greedy forwarding, i.e., the source node forwards the data packet to the neighbour that is closest to the destination in the coordinate space. Such greedy forwarding is repeated at the intermediate node, until the destination is reached. However, GPSR itself suffers from few drawbacks. Firstly, greedy forwarding over geographic coordinates may not be optimal due to unawareness of connectivity information of the network. Secondly, packet may get stuck in local minimum condition (a forwarding node could not find a neighbour that lies closer to the destination than itself) for sparse networks. To deal with above stated local minimum problem, the nodes switch to perimeter forwarding. In perimeter forwarding, the packet is forwarded to that neighbouring node which comes first in a planar sub graph of the network , when the line connecting the forwarding node and destination is rotated in the counter clockwise direction. The location of forwarding node where perimeter forwarding starts is recorded in the header of the data packet. Greedy forwarding is resumed when the data packet reaches a forwarding node which can find a neighbour node whose distance is smaller than the distance between the destination node and the node at which perimeter forwarding begun. The probability of finding a route between source destination node pairs is very high.

Most existing designs of wireless network routing protocols are based on the assumption that every node in the network is fault free. However, such assumption usually does not hold in realistic environments. Various kinds of faults such as crash fault, transient fault, Byzantine faults can occur in a node. The occurrence of faults affects the routing process therefore fault tolerance is of increasing importance in applications where it is essential to maintain efficient routing. In routing, fault tolerance helps in controlling the overhead which is there due to the faulty node and thus helps in considering the reliable routes. Many fault tolerance routing protocols have been proposed in wireless network. In Wireless sensor network(WSN), fault tolerance mechanism in routing protocol has been classified into two schemes, retransmission and replication. In MANET, works that focus on fault tolerant routing problem are very less. Wireless network comprises of large number of energy constraint node, some node may run out of energy and die and shorten the lifetime of network. Therefore, many routing protocols in MANET and WSN have been proposed with the aim to reduce energy consumption. GPER protocol was proposed to provide power efficient geographic routing in WNS. Traditionally, greedy forwarding and perimeter forwarding consider only single metrics i.e. minimum distance from destination and minimum angle in counter clockwise direction with respect to the line connecting the forwarding node and destination. These parameters does not take into account the energy conservation for evaluating the routing performance. In this paper, an energy efficient and fault tolerant routing algorithm has been proposed. The rest of the paper is organized as follows: The system models are described in Section 2. The energy-efficient and fault tolerant GPSR routing protocol is presented in Section 3. Section 4 presents the simulation result and comparison with other existing protocol. Section 5 conclude the paper and lists the future works.

2 Preliminaries

2.1 System, Fault and Energy Model

System Model. The system is composed of homogeneous n nodes, each having a unique id. All nodes have similar computing, storage resources and identical communication range. A set of nodes with circular radio range r, can be seen as a graph: each node is a vertex, and edge (n, m) exists between nodes n and m if the distance between n and m, $d(n, m) \leq r$. Source node knows the location of final destination.

Fault Model. Wireless network consists of many nodes which may be either fault free or faulty. The following assumptions have been made about faulty nodes:

1. No new faults occur during the execution of the routing protocol i.e. faults are permanent (a faulty node remains faulty until it is repaired and/or replaced).
2. Faulty nodes can be either hard faulty or soft faulty.

Nodes become hard faulty due to two main reasons. First, a node may be damaged during deployment or immediately after that. Second, depletion of battery power. When a node becomes hard faulty it does not participate in any further communication. Soft faulty nodes are more subtle than hard faulty nodes, since a soft-faulted node continues to communicate with the other node in the system although with altered specifications i.e., the faulty nodes may produce some random results instead of expected results. In this work, both hard-faulted and soft-faulted nodes have been considered.

The proposed routing protocol apply the following testing model given in Table 1 to check whether the node is faulty or fault free. The testing model uses the test task to check the validity of the node. The test task is to find out two's complement of five bit number. Tester node knows the result of the test task. Tester node broadcast the test task along with its id and location information. On receiving the test task, all the neighbouring node unicast the result of the test task along with their id and location information. As shown in the table of test model, five cases are observed. First case, the status of the tester node and tested node are both fault free then test result is zero. This means that there is match between the expected result of the tester node and the actual result returned by the tested node. Second case, the tester node is fault free and the tested node is soft faulty then test result is 1. This means there is a mismatch between the expected result and actual result. Third case, the tester node is fault free or soft faulty and the tested node is hard faulty then test result is NULL because a hard faulty node can receive a beacon test message but cannot send reply to it. Fourth case, the tester node is soft faulty and tested node is soft faulty or fault free then test result is 1. Here also there is mismatch between the expected result and actual result. Fifth case, the tester node is hard faulty and tested node is also faulty (soft or hard) or fault free then test result is NULL because a hard faulty node cannot send a beacon test message.

When the test result is 0, the node is considered as fault free, otherwise faulty. The test model given in Table 1 is used to select fault free nodes while establishing the paths between the nodes pair.

Table 1. Test model

Status of Tester node	**Status of Tested node**	**Test Result**
Fault free	Fault free	0
Fault free	Soft faulty	1
Fault free	Hard faulty	NULL
Soft faulty	Hard faulty	NULL
Soft faulty	Soft faulty	1
Soft faulty	Fault free	1
Hard faulty	Soft faulty	NULL
Hard faulty	Hard faulty	NULL
Hard faulty	Fault free	NULL

2.2 Energy Model

In wireless network energy consumption at each node can be due to transmission and reception of message. The energy model proposed by Heinzelman et al [6] to transmit an n-bit message over a distance d or receive an n-bit message are as follows:

$$E_{TX}(m,d) = E_{TX-elec}(m) + E_{TX-amp}(m,d) \tag{1}$$

$$= m * E_elec + m * \varepsilon_{amp} * d^2 \tag{2}$$

$$E_{RX}(m) = E_{RX-elec}(m) = m * E_{elec} \tag{3}$$

Table 2 gives related parameters and their definations.

Table 2. Radio parameters

Parameter	**Definition**	**Unit**
E_{elec}	Energy dissipation rate to run the radio	$50nJ/bit$
ε_{amp}	Energy dissipation rate to run transmit amplifier	$100pJ/bit/m^2$

3 The Proposed Algorithm

The proposed algorithm consists of four different but dependent phases which are as given below:

3.1 Fault Testing Phase

In this phase, all nodes come to know about the location information and residual energy information of their fault free neighbours at regular interval. Each tester node broadcast a beacon-test message containing its own id (e.g. IP address), location information and test task. Information regarding the test task in given in 2.1. Each tester node knows the result of the test task. On receiving the beacon-test message, all the neighbouring node execute the test task and unicast the reply message. The reply message contains the result of the test task, their unique Id, location information and residual energy information of the node. In this phase, tester node checks whether the neighbouring nodes present are fault free or faulty (i.e., soft or hard) by comparing the result of test task using the test model given in Table 1. Each node maintains the location information and residual energy information of all fault free neighbouring nodes in their range.

3.2 Planarization Phase

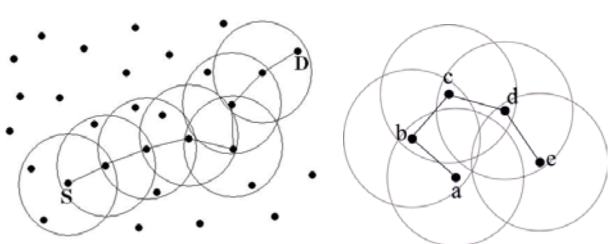
After fault testing phase, planarization phase starts. This phase can be done reactively or proactively. In proactive version, each node begins the planarization phase immediately after the fault testing phase. In reactive version, each node begins the planarization phase when local minimum condition occur. The main reason for using the planarization phase is that the right-hand rule does not work properly on full connected graphs with crossing edge. The right hand rule is used by perimeter forwarding phase. Each node should run the planarization algorithm in a distributed fashion. The Relative Neighbourhood Graph (RNG) and Gabriel Graph (GG) are two well known planar graph [7]. RNG planarization has been considered in this paper. During this phase one important property to be taken care is that, while removing edges from the graph to reduce it to RNG must not disconnect the graph.

3.3 Energy Efficient Greedy Forwarding Phase (EEGF)

In this phase, a forwarding node F uses full graph in choosing next hop. The optimal choice of next hop considers two metrics first minimum distance from destination(D) and second the energy. Algorithm 1 explains the pseudo code for the proposed energy efficient greedy forwarding used at a forwarding node. In traditional GPSR, the distance between forwarding node as well as its neighbour with respect to destination is calculated. The forwarding node selects that neighbour as the next hop that lies closest to the destination. If the forwarding node F could not find a neighbour node that lies closer to the destination than itself, then node switches to perimeter forwarding. However in EEGF, forwarding node first form a set of Selected-Neighbour(F) which is a subset of the Neighbour-list(F). Each node I ε Selected-Neighbour(F), if the distance between the neighbour node I and the destination node D is less than the distance between the forwarding node F and D. After this, $Cost(I)$ of every node I ε Selected-Neighbour(F) is calculated. $Cost(I)$ is defined as

$$Cost(I) = \frac{ResidualEnergy(I)}{Distance(F, I)^2} \tag{4}$$

where, $ResidualEnergy(I)$ is the $AvailableEnergy(I)$ divided by $InitialEnergy$ (I) and $Distance(F, I)$ is the difference in the distance between F And D and the distance between I and D divided by the distance between F and D. The reason for calculating this cost is that relation between energy and distance are inversely proportional to each other because from energy balance point of view, the node with more residual energy should be selected as the next hop and from distance point of view, the node which is closer to the destination should be selected as the next hop. Thus, $Cost(I)$ balances the two metrics energy and distance and helps in finding the optimal next hop. The forwarding node selects that neighbour I ε Selected-Neighbour(F) as next hop, whose $Cost(I)$ is maximum. If the forwarding node F could not find a neighbour node that lies closer to the destination than itself, that is Selected-Neighbour(F) is empty then node switches to perimeter forwarding.



(a) (b)

Fig. 1. (a) Greedy forwarding and (b) Local minima condition

Require: Destination D, Forwarding node F, Neighbour-list(F)
Ensure: Next-Node // if Energy efficient greedy forwarding is successful
Switch to perimeter forwarding // if Energy efficient greedy forwarding is not successful
Intialization: Next-Node=NULL; Maximum-Cost=0.0;
Selected-Neighbour(F)=0

Dist(F-D) $= \sqrt{(x_F - x_D)^2 + (y_F - y_D)^2}$
while neighbour-node $I \; \varepsilon$ Neighbour-list(N) **do**
  Dist(I-D) $= \sqrt{(x_I - x_D)^2 + (y_I - y_D)^2}$
  if Distance(I-D) < Dist(F-D) **then**
    Selected-Neighbour(F) $\rightarrow$ Selected-Neighbour(F)$\cup$(I)
  end if
end while
while neighbour-node $I \; \varepsilon$ Selected-Neighbour-list(N) **do**
  $ResidualEnergy(I) = \frac{AvailableEnergy(I)}{InitialEnergy(I)}$
  $Distance(F, I) = \frac{Dist(F-D) - Dist(I-D)}{Dist(F-D)}$
  $Cost(I) = \frac{ResidualEnergy}{Distance(F,I)^2}$
  if Maximum-Cost < Cost(I) **then**
    $Maximum - Cost = Cost(I)$
    $Next - Node \leftarrow I$
  end if
end while

Algorithm 1. Energy-Efficient Greedy Forwarding Algorithm

3.4 Energy Efficient Perimeter Forwarding

Energy efficient perimeter forwarding uses right hand rule/left hand rule as given below:

Right Hand Rule. The right-hand rule to traverse a graph is shown in Figure2 (a). This rule states that when arriving at node b from node a, the next edge traversed is the next one sequentially counterclockwise about b from edge (a, b). On graphs with edges that cross, the right-hand rule may take a degenerate tour of edges that does not trace the correct path as shown in figure 2 (b). Now on removing the crossing edges right hand rule traces the correct path (Figure 2 (c)). In this context planarization comes into picture. The left hand rule also works in a similar fashion the only difference is that the next edge traversed is in clockwise direction

Energy Efficient Perimeter Forwarding. In this phase, complete EFGPSR is described which combines fault testing phase, energy efficient greedy forwarding phase and energy efficient perimeter forwarding phase. Greedy forwarding phase is executed on the full network graph whereas energy efficient perimeter forwarding phase on the planarized network graph where greedy forwarding

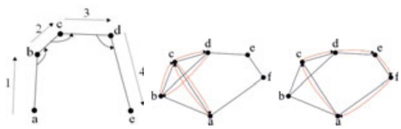


Fig. 2. (a) Right Hand Rule (RHR), (b) RHR with crossing edge, (c) RHR without crossing edge

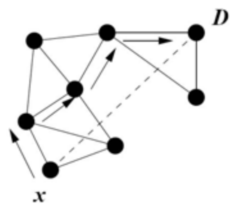


Fig. 3. Perimeter forwarding example

fails. A flag is used in the packet header of EFGPSR which indicate whether the packet is in greedy mode or perimeter mode. Initially, source node marks all data packets as greedy-mode. On receiving greedy-mode packet, a forwarding node uses the energy efficient greedy forwarding algorithm to find the optimal next hop. If no optimal node is found i.e. greedy forwarding fails, the node marks the packet into perimeter mode. In traditional perimeter forwarding as shown in figure 3 the packet is forwarded to that neighbour node that comes first in a planar sub graph of the network , when the line connecting the forwarding node and destination(D) is rotated in the counter clockwise direction. The location of forwarding node where perimeter forwarding starts is recorded in the header of the data packet. Greedy forwarding is resumed when the data packet reaches a forwarding node or a node which can find a neighbour node whose distance is smaller than the distance between the destination node and the node at which perimeter forwarding begun. However, in energy efficient perimeter forwarding, the forwarding node select the next hop using the following algorithm:

4 Simulations and Results

In this paper, the performance of GPSR and EFGPSR is evaluated based on NS-2 network simulator. The network dimension used for simulation is 1000$\times$1000meter square. The transmission range of each node is assumed to be 250m. For traffic source 20 traffic flow originated by 20 sending nodes has been considered. The MAC layer protocol used is IEEE 802.11. Simulation results are shown in figure 4 and figure 5.

Require: Forwarding node F, Neighbouring-node-list containing *index*, *angle* and *energy* of each node
Ensure: Next-Node

```
Intialization: Next-Node=NULL; Temp=0; Max=Max-energy;
Min=Min-energy; Mid=Mid-energy
Sort the Neighbouring-node-list in increasing order of angle
for all Neighbouring-node
Temp=First nodes after sorting
if Temp[energy] < Mid and Temp[energy] >= Max then
   Select node index
   break
end if
End for
```

Algorithm 2. Energy-Efficient Perimeter Forwarding Algorithm

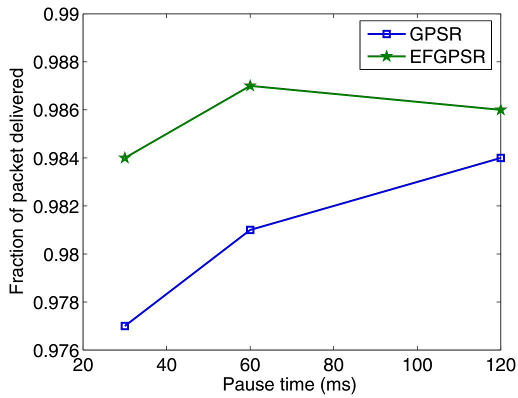


Fig. 4. Packet Delivery Success Rate. GPSR and EFGPSR with becon interval 1. 50 nodes

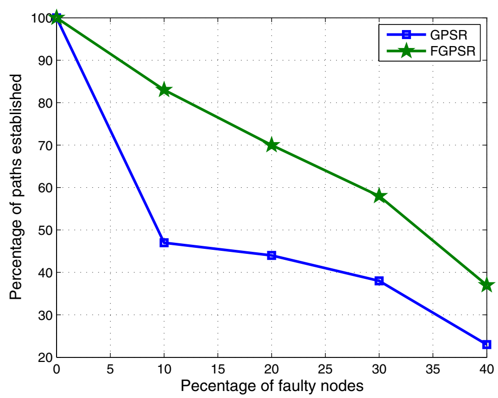


Fig. 5. GPSR vs EFGPSR

5 Conclusions

In this paper, fault tolerant GPSR routing algorithm has been proposed. EFGPSR and GPSR have been simulated for the 1000×1000-sized static networks of 50 node. 20 traffic flow originated by 20 sending nodes has been considered. The result carried out shows that EFGPSR provides higher no. of path between source and destination out of the total no. of paths (i.e., 20) than GPSR due to the fact that GPSR does not check the faulty node which causes long detouring paths and may not establish path to the destination shown in figure 5. And also EFGPSR try to maintain balance between the metrics to choose the next hop (i.e. distance from destination in greedy forwarding phase and minimum angle with respect to the line connecting the forwarding node and destination in perimeter forwarding phase) and selection of node having highest energy among the neighbouring node to extend network lifetime. Evaluation and comparison of GPSR and EFGPSR shows that EFGPSR performs better in terms of increasing the network lifetime, successful packet delivery ratio with insignificant increase in number of hop count.

References

1. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing, RFC Editor (2003)
2. Charles, E., Perkins, B.P.: Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. SIGCOMM Comput. Commun. Rev. 24, 234–244 (1994)
3. Karp, B., Kung, H.: GPSR: greedy perimeter stateless routing for wireless networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, pp. 243–254 (August 2000)
4. Wu, S., Candan, K.: GPER: Geographic Power Efficient Routing in Sensor Network. In: Proceeding of Int'l Conference Network Protocols (ICNP), pp. 161–172. IEEE, Los Alamitos (2004)
5. Melodia, T., Pompili, D., Akyildiz, I.: Optimal Local Topology Knowledge for Energy Efficient Geographical Routing in Sensor Network. In: Proceeding of INFOCOM. IEEE, Los Alamitos (2004)
6. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proceedings of the 33rd Hawaii International Conference on System Sciences, vol. 8. IEEE Computer Society, Hawai (2000)
7. Agarwal, P., Matausek, J.: Relative neighborhood graphs in three dimensions. In: Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 58–65. Society for Industrial and Applied Mathematics, Philadelphia (1992)