An energy efficient routing algorithm based on genetic algorithm for ad hoc networks

Marek Pilski[1*], Franciszek Seredyński[2,3]

[1]University of Podlasie, Computer Science Department, Sienkiewicza 51, 08-110 Siedlce, Poland
[2]Polish-Japanese Institute of Information Technology, Koszykowa 86, 02-008 Warszawa, Poland
[3]Institute of Computer Science, Polish Academy of Sciences, Ordona 21, 01-237 Warszawa, Poland

Abstract

Highly dynamic topology is an essential feature of mobile ad hoc networks. For this reason, maintaining a consistent state for routing purposes can be a very difficult task. The primary goal is correct and efficient route establishment between a pair of nodes. Although a more challenging goal is to provide energy efficient routing protocols. This paper presents a concept of an energy efficient routing algorithm based on applying genetic algorithm (GA). The aim of the algorithm is a prolongation of life time of the network. The life time of the network depends on nodes life, which is a function of battery energy nodes. A routing metric is a time measure from a moment when the network starts up, to the moment when the first battery in any nodes runs down. The purpose of the proposed algorithm is to maximize the life time of the network. To choose a path we propose to use one of a few different heuristics. In this paper we describe the GA-based approach to find a heuristic combination for solving power-aware routing problem. A problem solution is such a heuristic combination, which depends on an actual state of the network (energy of nodes) and will choose optimal paths.

1. Introduction

There are many situations where a data network is required in places where there is no fixed networking infrastructure and no time to create such an infrastructure. Examples of such situations occur in military operations [1], law enforcement, rescue operations and personal area networking (devices around the home/office). Ad hoc networking, the ability to form a network dynamically from scratch using wireless connections, addresses this need. When the nodes of the network are mobile, as is usually the case, the networks formed are called mobile ad hoc networks (MANETs). The dynamic nature of MANETs provides special challenges beyond those in standard data networks [2,3]. MANETs

---

* Corresponding author: *e-mail address*: pilski@ii3.ap.siedlce.pl

require special protocols, a review of a range of such protocols is given in [4]. Not only the network must be initiated, but it also needs to maintain itself as the nodes move around and as transmission properties change. With changing connectivity between nodes, including the entrance and exit of nodes, the network must have the ability to adapt its topology to reflect these changes. All nodes in such networks take two roles: producer and consumer in one of data packet streams, and routers for data packets destined for the other nodes. There exist currently more than a hundred routing protocols for MANETs [5].

Equally important feature in mobile ad hoc networks is limited battery capacity of the nodes (for example, the nodes in MANETs may be devices like: laptops, palmtops, mobile-phones). The limited battery capacity poses yet another challenge for the routing algorithms: to distribute the packets on multiple paths in such a manner that the battery of different nodes deplete at an equal rate, as a result, the life time of the network could be increased. This feature of the nodes is the reason for creating a new group of protocols called Energy Efficient Routing Protocols or Power-Aware Routing Protocols for MANETs [6,7]. The metrics for energy efficient routing are also introduced in [7], and it is evident that an energy aware routing algorithm is expected to degrade the traditional performance metrics of a routing algorithm i.e. throughput and packet delay. The real dilemma in MANETs is: how to to design a routing algorithm which is not only energy efficient but also provides the same performance as that of the existing state-of-the-art algorithms. The routing algorithms for MANETs can be broadly classified as proactive algorithms or reactive algorithms. Proactive algorithms periodically launch control packets which collect information about a new network state and update the routing tables accordingly. On the other hand, reactive algorithms find routes on-demand only. Reactive algorithms look more promising from the perspective of energy consumption in MANETs.

In this paper we consider the continuous process of choosing paths between communicating nodes in MANETs and power management. The aim of the proposed routing algorithm is a prolongation of life time of the network. A routing metric is a time measure from a moment when the network starts up, to the moment when the first battery in any nodes runs down. A problem solution is such a heuristic combination, which depends on an actual state of the network (energy of nodes) and will choose paths maximizing the routing metric. We propose the Genetic Algorithm-based approach [8] to find a heuristic combination for solving the power-aware routing problem.

The paper is organized in the following way. The next section presents an idea of a new energy efficient routing protocol. Section 3 describes representation of the network state and section 4 shows how information about the routes is collected. Section 5 presents five heuristics, which we used and

section 6 shows genetic algorithm which uses these heuristics to establish connections between nodes in such a way as to optimize accepted criterion of the network life. The last section contains conclusion.

2. The idea of a new energy efficient routing protocol

There exist a number of power-aware routing protocols, which perform well in specific network situations. However, a network state may change dynamically in time, and there is a need to use properly these protocols depending on the network state. Such a mechanism can be provided by a global mechanism which uses some meta-heuristics. The purpose of such a meta-heuristic is to use a combination of available protocols (heuristics) in order to solve a current problem in the network. In this paper we will use evolutionary-based hyper-heuristic [9] to discover a combination of protocols (heuristics) to solve power-aware routing problem. The problem solution procedure based on applying a meta-heuristic is the following:

– given a state *p* of the problem,
– find currently the best heuristic *h(p)* for a given state of the problem and apply it,
– this transforms the problem state, to state *p'*,
– repeat this until the problem has been solved.

The purpose of GA is to define key-situations in the network states and associate with them combination of heuristics (communication protocols) which can be applied to provide efficient functioning of the network.

3. Representation of the network state

We assume that the network state is defined by four variables. The variables describe how many nodes of the network are into each of the four categories at battery energy level:

– huge (*H*) stockpile of energy ($e_{best}/2 < n_i$),
– large (*L*) stockpile of energy ($e_{best}/3 < n_i \le e_{best}/2$),
– medium (*M*) stockpile of energy ($e_{best}/4 < n_i \le e_{best}/3$)
– small (*S*) stockpile of energy ($n_i \le e_{best}/4$),

where: n_i – a node in mobile ad hoc networks and e_{best} – an energy of the best node (the node with the greatest stockpile of energy).

For example, the network state in time *t(1)* is described as follows: *H=100%, L=0%, M=0%* and *S=0%* (assumption: when the network starts up all nodes have the same energy in batteries). The example shows that all nodes are inside the *H* category (100% nodes), there are not nodes in *L*, *M* and *S* categories. For example, the network state in time *t(n)* may be described as follows: *H=55%,*

L=40%, M=3% and *S=2%*. It is worth noting that this representation gets rid of many details.

4. Collecting information about the routes

When the node *S* (source) wants to send data-packets to the node *D* (destination), it must know the route. To establish a connection between two hosts not directly connected, messages must be routed by the intermediate node. The traditional broadcasting method used to discover routes relays on dissemination of a message called a *Route Request*. When the node *S* wants to find a route to the node *D*, it sends a route discovery message to its neighborhood (on-demand). Each of its neighbors adds its name to the message and re-emits it. Every other node in the network does so and finally, the searched node *D* receives the message with the chain of nodes followed from the source node *S*. The final step of this process is the emission of a message called a *Route Reply* by the node *D* to inform *S* that a route has been found. This procedure is the basis for an IETF (Internet Engineering Task Force) as standardized protocol [10]. The broadcasting task is a fundamental mechanism in route discovery. Therefore, it is unceasingly the subject of the study [11,12]. The broadcasting method is illustrated in Fig. 1.
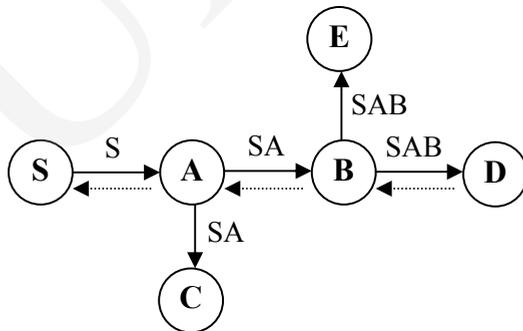

Fig. 1. The broadcasting task – the route request from node *S* to node *D* (continuous arrow) and route replay from node *D* to node *S* (dotted arrow)

The node *S* wants to find a route to *D*, so it sends a route request message to its neighborhood with its *ID* as the initial chain of hosts. The node *A* receives this message, writes its name in the chain and re-emits it. *S* gets the packet, but ignores it (it already knows it and furthermore, it is the one that first emitted it). *B* does the same as *A* (as well as *C* and *E*), and finally *D* receives the request. Every node in the network retransmits once the message, upon receiving the first copy of it, and will consequently ignore further copies of the same message. It then just has to reply to *S* by emitting *Route Reply*. This last message is addressed directly to *S*, because *D* knows the route to it. This one is simply

written in the route request message, formed by the chain of hosts that relayed the message. In the example, the chain received by *D* was *SAB*, so it sends the route reply with BAS as the route to follow. Route discovery is performed by a broadcasting task, while the *Route Reply* is an unicast routing operation. Each answer (*Route Reply*) returned from *D* contains: the route (e.g. *SAB*) and the energy of each node on the route: energy of node *A* and energy of node *B*.

5. The set of heuristics used

We consider a simple ad hoc network with six nodes (Fig. 2). Let the energy values of batteries in nodes be the following: the energy in node *A=70%*, *B=80%*, *C=40%*, *E=90%*. We want to choose the path between the node *S* and the node *D*.
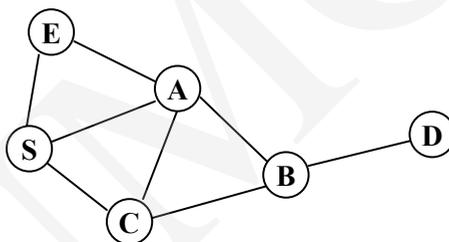
Fig. 2. The example of ad hoc network

Our GA-based meta-heuristic will use five basic heuristics:

5.1. Heuristic h1

Choose the path as follows:
– Step 1. For each path mark the weakest node (with the smallest energy in battery on the path).
– Step 2. Select a path from marked nodes which have the biggest battery energy. If there exists more than one path, select the shortest path.

> S-E90%-A70%-B80%-D
> S-A70%-B80%-D {selected path}
> S-C40%-B80%-D
> S-C40%-A70%-B80%-D
> S-A70%-C40%-B80%-D
> S-E90%-A70%-C40%-B80%-D

Fig. 3. The example of choosing a path by heuristic h1

5.2. Heuristic h2

Let *T* be a set of all paths. Choose the path as follows:

- Step 1. The set *T* of paths is sorted according to the weakest node on the path.
- Step 2. Select subset *T1* from *T*, such that it contains a node with the highest value of energy of the weakest node.
- Step 3. Randomly select a path from the set *T1*.

The size of set *T1* is a parameter of the algorithm.

```
S-A70%-B80%-D
S-E90%-A70%-B80%-D {selected path}
S-C40%-B80%-D
S-C40%-A70%-B80%-D
S-A70%-C40%-B80%-D
S-E90%-A70%-C40%-B80%-D
```
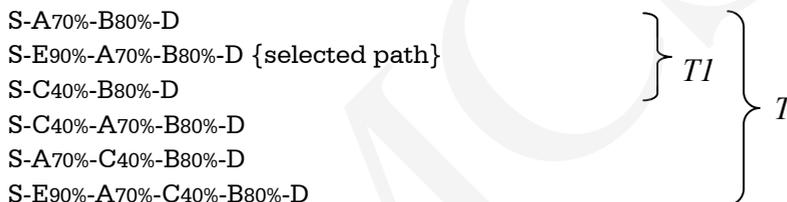
Fig. 4. The example of choosing a path by heuristic h2

5.3. Heuristic h3

Heuristic h3 is similar to heuristic h2. These heuristics differ in the end step.
Let *T* be a set of all paths. Choose the path as follows:

- Step 1. The set *T* of paths is sorted according to the weakest node on the path.
- Step 2. Select the subset *T1* from *T*, such that it contains a node with the highest value of energy of the weakest node.
- Step 3. Randomly select a path from the set T1.

The size of set *T1* is a parameter of the algorithm.

```
S-E90%-A70%-B80%-D
S-A70%-B80%-D {selected path}
S-C40%-B80%-D
S-C40%-A70%-B80%-D
S-A70%-C40%-B80%-D
S-E90%-A70%-C40%-B80%-D
```
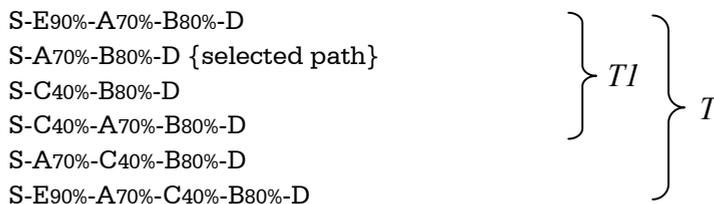
Fig. 5. The example of choosing a path by heuristic h3

5.4. Heuristic h4

Heuristic *h4* behavior is similar to many other routing protocols which select the shortest path between two nodes in the network.

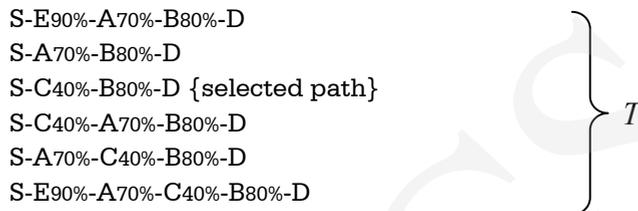Let T be a set of all paths. Choose the path as follows: select the shortest path from the set T.

S-E90%-A70%-B80%-D
S-A70%-B80%-D
S-C40%-B80%-D {selected path}
S-C40%-A70%-B80%-D
S-A70%-C40%-B80%-D
S-E90%-A70%-C40%-B80%-D

$\left.\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\right\} T$

Fig. 6. The example of choosing a path by heuristic h4

5.5. Heuristic h5

Heuristic *h5* with a randomly selected path we use in order to check what is the influence of random choice of path. The strategy usually allows to leave the local optimum. In this case we will omit the most frequently used paths.
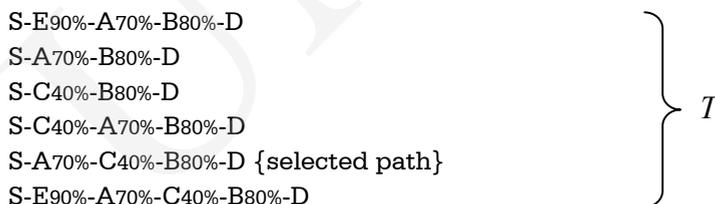Let T be a set of all paths. Choose the path as follows: randomly select a path from the set T.

S-E90%-A70%-B80%-D
S-A70%-B80%-D
S-C40%-B80%-D
S-C40%-A70%-B80%-D
S-A70%-C40%-B80%-D {selected path}
S-E90%-A70%-C40%-B80%-D

$\left.\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\right\} T$

Fig. 7. The example of choosing a path by heuristic h5

6. Genetic algorithms

Genetic Algorithms [8] are adaptive heuristic search algorithms premised on the evolutionary ideas of natural selection and genetics. The basic concept of GA is designed to simulate processes in the natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

We use the genetic algorithm-based approach to find a heuristic combination for solving a routing problem with taking battery energy into account.

6.1. The algorithm

The genetic algorithm can be described as follows. Firstly, we generate random population P of n chromosomes (suitable solutions of the problem).

Next, we evaluate each chromosome *x* in the current population *P* according to the fitness function *f(x)*. In each iteration of the algorithm called a generation, we sequentially perform three basic genetic operators:

- selection – this operator selects chromosomes from a population according to their fitness,
- crossover – this operator with a crossover probability cross over two parent chromosomes and forms a new offspring,
- mutation – this operator with a mutation probability mutates a new offspring at each locus (position in chromosome).

Then, the two children replace two members of the current population *P*. Next, we again compute the fitness *f(x)* of each chromosome *x* in the population. The process is repeated a fixed number of times. If the algorithm stops, we return the best solution (chromosome) in the current population *P*.

```
Chromosome: x=(b1, b2, ..., bn) {structure representing a solution};
Optimization criterion: maximizing the routing metric;
Initialize population: P() {set of chromosomes};
t = 0;
Evaluate P(t);
WHILE termination condition NOT TRUE
t = t + 1;
Select P(t) from P(t–1) to reproduce;
Crossover in P(t);
Mutate in P(t);
Evaluate P(t);
DO
Problem_solution = the best chromosome from P(t).
```

Fig. 8. The pseudo-code of the genetic algorithm

6.2. Chromosome representation

A chromosome is composed of blocks. Each block *j* contains five numbers: *Hj, Lj, Mj, Sj, hj*. The 1st four numbers represent the ad hoc network state, all in the range 0 to 1. The 5th number, *hj*, is an integer in the range 1 to 5 indicating which heuristic is associated with this block. An example of chromosome can be seen in Fig. 9.

The network state in time *t* is described by four numbers. The numbers describe how many nodes are into each of four categories: *L, H, M, S*. The current network state (*CNS*) is a point inside the 4-dimensional space. On the other hand, the blocks in the chromosome also represent a number of points inside the 4-dimensional space, and each point is associated with the heuristic *h*.

The currently operating ad hoc network changes its states represented by *CNS* points. Each node in MANETs has the same chromosome which is the problem solution. If the node S wants to send data-packets to the node D, it must choose a point (block) from the chromosome closest to that describing the actual network state and use associated heuristic with the chosen block to routing.

H	L	M	S	h	H	L	M	S	h	H	L	M	S	h
0.90	0.10	0.00	0.00	5	0.70	0.20	0.08	0.02	3	0.10	0.10	0.35	0.45	1

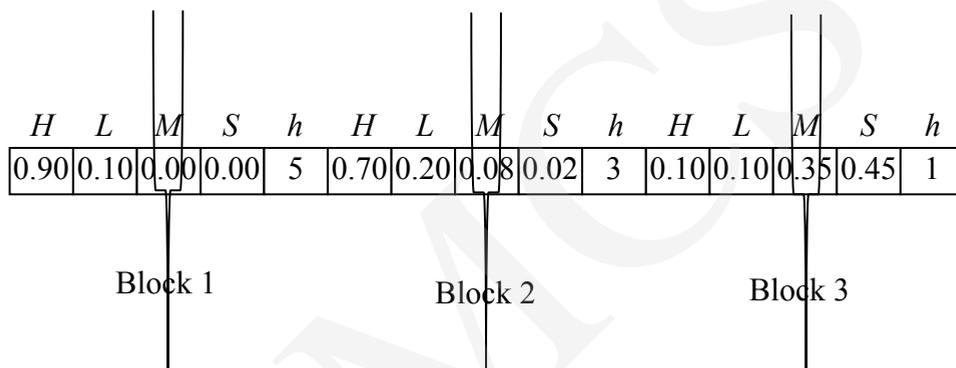Block 1   Block 2   Block 3

Fig. 9. The example of chromosome consisting of three blocs

We do not know how many blocks are needed. Therefore we use a variable-length chromosome.

6.3. Fitness function

The fitness describes how well a chromosome solves a specified problem. In this case it is a prolongation of life time of the ad hoc network. We have two methods to calculate the fitness function. The first method is based on chromosomes ranking in relation to the best known solution. The metric is a time measure from a moment when the network starts up, to the moment when the first battery in any nodes runs down. The second method is based on the ranking which describes how better result returns a given chromosome than any single heuristic while solving the problem.

6.4. Selection

We have implemented a number of selection operators. These are the following operators:
- proportional selection,
- tournament selection,
- ranking selection.

6.5. Crossover

We use two-point crossover. We have variable length chromosomes, therefore firstly two points are selected on the shortest parent. The same points are selected on the second parent. Everything between the two points is swapped between the parents, rendering two children chromosomes.

6.6. Mutation

We have three mutations.

– Add-block mutation. The mutation generates a new block and adds it to the chromosome. The first four numbers (four categories which describe the network state) of a block are randomly generated in the range 0 to 1. The fifth number (the heuristic, an integer in 1 to 5) is generated randomly too.
– Remove-block mutation. The mutation removes a random block from the chromosome.
– Normal mutation. The mutation changes a locus (only one element from a block). In randomly chosen locus a suitable value is inserted. The value depends on the position in the block. For the first four cells in the block we inserted a value in the range 0 to 1. For the fifth cell we inserted an integer in the range 1 to 5.

A child has a chance of being mutated with a certain probability. If it is to be mutated, then one type is chosen. The probability of mutation and the type of mutation are parameters of the algorithm.

Conclusion

This work presents an approach to developing the algorithm of an energy efficient routing protocols in MANETs. The problem is solved by continuous process of choosing paths between communicating nodes in MANETs and power management. The purpose of the proposed algorithm is to maximize the life time of the network. The life time depends on nodes life, which is a function battery energy of nodes. A routing metric is a time measure from the moment the network starts up, to the moment when the first battery in any nodes runs down. We proposed the genetic algorithm-based approach to find a heuristic combination for solving a power-aware routing problem. A problem solution is such a heuristic combination, which depends on an actual state of the network and will choose optimal paths. Currently the algorithm is the subject of experimental study.

References

[1] Brown W.W., Marano V., MacCorkell W.H., Krout T., *Future Combat System-Scalable Mobile Network Demonstration Performance and Validation Results*. Military Communications Conference. MILCOM. IEEE, 2 (2003) 13; 2 (2003) 1286.

[2] Basagni S., Conti M., Giordano S., Stojmenović I., *Mobile ad hoc Networking*. Published Wiley-IEEE Press, (2004).

[3] Ramanathan R. Redi J., *A Brief Overview of ad hoc Networks: Challenges and Directions*. IEEE Communications Magazine, 40(5) (2002) 20.

[4] Abolhasan M., Wysocki T.A., Dutkiewicz E., *A Review of Routing Protocols for Mobile ad hoc Networks*. Elsevier, Ad Hoc Networks, 2(1) (2004) 1.

[5] http://en.wikipedia.org/wiki/Ad_hoc_protocol_list (visited: January 2008).

[6] Cui Y., Xue Y., Nahrstedt K., *A Utility-Based Distributed Maximum Lifetime Routing Algorithm for Wireless Networks*. Vehicular Technology, IEEE Transactions on vehicular technology, 55(3) (2006) 797.

[7] Wedde H.F., Farooq M., Pannenbaecker T., Vogel B., Mueller C., Meth J., Jeruschkat R., *BeeAdHoc: An Energy Efficient Routing Algorithm for Mobile ad hoc Networks Inspired by Bee Behavior*, Proc. of Genetic and Evolutionary Computation Conference, (2005) 153.

[8] Goldberg D.E., *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, (1989).

[9] Ross P., Marın-Blazquez J.G., Schulenburg S., Hart E., *Learning a Procedure That Can Solve Hard Bin-Packing Problems: A New GA-Based Approach to Hyper-heuristics*. Genetic and Evolutionary Computation – GECCO 2003, Springer-Verlag Berlin Heidelberg, LNCS 2724, (2003) 1295.

[10] Johnson D.B., Maltz D.A., Y.-C. H., *The Dynamic Source Routing Protocol for Mobile ad hoc Networks (DSR)*. http://tools.ietf.org/id/draft-ietf-manet-dsr-10.txt, 2004. (visited: January 2008).

[11] Tseng Y.-C., Ni S.-Y., Chen Y.-S., Sheu J.-P., *The Broadcast Storm Problem in a Mobile ad hoc Network*. Wireless Networks, Short version in MOBICOM 99, 8 (2002) 153.

[12] Ingelrest F., Simplot-Ryl D., Stojmenovic I., *Energy-Efficient Broadcasting in Wireless Mobile ad hoc Networks*. Resource Management in Wireless Networking, Springer, (2005) 543.