Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 31 (2008) 858–869
computer communications
www.elsevier.com/locate/comcom

A genetic algorithm for energy-efficient based multicast routing on MANETs

Yun-Sheng Yen [a], Yi-Kung Chan [a], Han-Chieh Chao [b,*], Jong Hyuk Park [c]

[a] Department of Electrical Engineering, National Dong Hwa University, Taiwan, ROC
[b] Department of Electronic Engineering, National Ilan University, Taiwan, ROC
[c] Department of Computer Science and Engineering, Kyungnam University, Kyungnam, North Korea

Available online 24 October 2007

Abstract

In ad hoc networks, mobile node battery energy is finite and represents one of the greatest constraints for designing multicast routing protocols. In regards to the battery lifetime limitation in supporting multicast routing, some studies have investigated a power saving network layer. These proposed methods have always considered several techniques such as a route load for relaying, battery lifetime in route selection, decreasing the frequency of sending control messages, optimizing the size of control headers, and efficient route re-configuration techniques. This paper discusses the multicast routing problem with multiple QoS constraints in MANETs. It is also an *NP-Complete* problem that deals with the various constraints. We propose an energy-efficient genetic algorithm mechanism to resolve these problems. Furthermore, we design a source-tree-based routing algorithm and build the shortest-path multicast tree to minimize delay time by using a small population size in the genetic algorithm. Only a few nodes are involved in the route computation. We also improve the genetic sequence and topology encoding and prolong the lifetime of mobile nodes that calculate the residual battery energy of all nodes in a multicast tree. The simulation results show that our proposal method is an efficient and robust algorithm for multicast route selection.

Keywords: Mobile Ad Hoc networks (MANETs); Multicast routing; Service of quality (QoS); Genetic algorithm (GA); Sequence and topology (ST) encoding

1. Introduction

Mobile Ad Hoc Networks (MANETs) are groups of self-organizing mobile nodes in dynamic topology networks. MANETs do not need infrastructure units such as base stations or access points in advance. Each movable node in the MANETs has a routing function whereby it communicates by forwarding datagrams via intermediate nodes. If two movable nodes are located within the forwarding range, they communicate with each other directly. Otherwise, they need another node to forward their datagrams and require a datagram forwarding operation using a multi-point hopping method. MANETs [16] are characterized by non-restricted mobility and easy deployment, which makes them very popular.

In general, the design of a QoS multicast routing protocol [7] with multi-constrained metrics has not always taken into consideration the consumption of battery energy. Thus, upon operation of the whole network, some mobile nodes can have problems with energy overhead due to a lack of balance in their battery energy consumption. Once these selected nodes in the multicast tree run out of residual battery energy [21], an interruption condition arises that is generated in the link during packet forwarding. Thus, there are numerous interruptions that can occur in selected packet forwarding routing paths. The multicast service lifetime will not be maintained continuously until the comple-

* Corresponding author.
E-mail addresses: d9223008@em92.ndhu.edu.tw (Y.-S. Yen), m9223511@em92.ndhu.edu.tw (Y.-K. Chan), hcc@niu.edu.tw (H.-C. Chao), jhpark1@kyungnam.ac.kr (J.H. Park).

0140-3664/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2007.10.033

tion of packet forwarding. The idea is to maximize the duration so that all mobile nodes are up until one of them is drained of energy. As mentioned above, the QoS items that are called *metrics* include available bandwidth, end-to-end delay, probability of packet loss, delay variance (jitter), expense, and so on. Since different items have different properties, they could be classified into three categories, namely additive, multiplicative, and minimal properties. Many relevant references which have been cited in multicast routing topics usually tackle some of the general QoS metrics such as the bandwidth, packet loss rate, and propagation delay.

A genetic algorithm (GA) is a searching algorithm that utilizes the genetic operators, for examples, crossover and mutation [2–4,10,19,22]. It emulates the evolution idea using natural selection and the survival of the fittest concept (i.e., as shown in Fig. 1). In each generation, a new population of solutions is created by exchanging and combining the information obtained from the solutions through the previous generation. Crossover is one of the genetic operators in which genes from two chromosomes are exchanged and the genotypes of two selected parents are merged to make two new offspring. Crossover is also referred to as recombination and sometimes called mating. Two chromosomes with greater fitness values are picked from the chromosome pool. The starting point and length of the portion to be exchanged are randomly selected. The two new offspring are created and put back into the chromosome pool. Mutation is another genetic operator in which new genetic structures are inducted into the population by randomly modifying some of the genes. Mutation also maintains the search algorithm to escape from local optimum and prevents converging too fast. In other words, mutation operation gives the genetic algorithm an opportunity to search for new and more feasible chromosomes in new areas of solution space. After the mutation operation, the multicast tree will be modified because the mutation operator can destroy the tree structure and outgoing degree constraints. The other genetic algorithm operations include encoding, initial population, evaluation, reproduction, crossover and mutation.

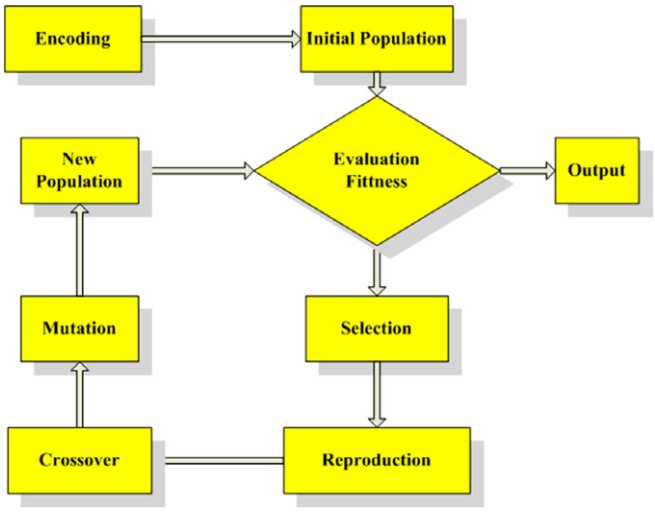

Fig. 1. The flowchart for a typical genetic algorithm.

Basically, a genetic algorithm (GA) is a search technique used in computer science to find approximate solutions for optimization. Our proposal is based on a multicast spanning tree that can be pre-built with minimum delay constraint and that prolongs the lifetime of multicast service. Moreover, making all mobile nodes share the multicast tree total energy is applicable for nodes with much greater energy. In this way, the lifetime of the multicast service is prolonged because no interruption occurs during the packet forwarding process due to shortages in battery energy. We present an extended sequence and topology encoding method that is an improved version of the sequence and topology adopted from [1]. It will be more efficient in controlling the outgoing node degree and can simplify the genetic operation. We utilize an improved mating mechanism and energy-efficient mutation as the repair function. The proposed method is applicable in reducing the degree of nodes by replacing a node with lower energy by another with higher energy. Hence, distributing the total battery energy consumption for the multicast service is carried out using higher residual battery energy (RBE) nodes. The simulation results show that our proposal is an efficient and robust algorithm for multicast route selection.

The remainder of this paper is organized as follows: Section 2 surveys the conventional genetic encoding. Section 3 describes the multicast model and extended tree encoding. Section 4 introduces our proposal. Section 5 discusses the analysis of convergence for the genetic algorithm and shows the simulation results. The last section presents our conclusions.

2. Conventional genetic encoding

The genetic encoding evolution process is designed to find optimal network. A better encoding approach would make the crossover and mutation operations more efficient. There are three major encoding scheme categories in the tree-based topology [21]: (i) edge encoding, which is generally unsuitable for a spanning tree; (ii) vertex encoding, which is more popular and often employs the Prüfer number encoding method [8]; (iii) edge and vertex encoding. Prüfer number encoding [8] a named tree is a unique sequence associated with the tree. The sequence for this tree on n vertices has length $n-2$, and can be generated using a simple iterative algorithm. It can generate a named tree's Prüfer number by iteratively removing vertices from the tree until only two vertices remain. Consider a named tree T with vertices $\{1,2,\ldots,n\}$. At step i, remove the leaf with the smallest label and set the ith element of the Prüfer number as the label for this leaf's neighbor. An example as shown in Fig. 2, we give a string P_n of Prüfer number and its leaf node string P_l. In [8], we can get below to illustrate

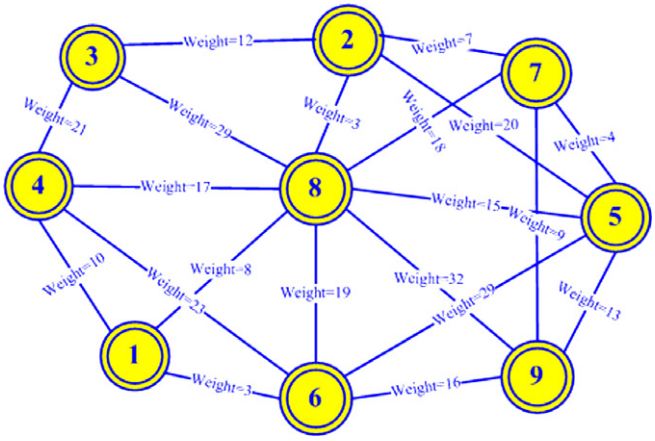

Fig. 2. An example of the network topology.

the Prüfer number encoding and decoding results with more details.

2.1. Encoding of Prüfer number

(1): Marking an identification code onto each node in an extending multicast tree T with n nodes.
(2): Selecting the leaf node u with a minimum identification code.
(3): Finding a unique node v that is connected to a certain node u, taking node v as a Prüfer number and then decoding the first bit located at the leftmost end.
(4): Deleting node u and the link connecting node u and node v so that the remaining number of nodes in the extended tree T is $n-1$ node.
(5): Operating steps 2, 3 and 4 repeatedly from left to right in a sequential manner to acquire each Prüfer number until there is only one side left. In this way, we acquire a Prüfer number encoding string P_n.
(6): Marking an identification code for the remaining nodes that are not shown in P_n, and then acquiring a leaf node string P_l by sorting sequentially from small to large. u represents the node located at the last position in the whole tree topology, i.e., there is no other node connected to it that has a minimum identification code. v represents the non-leaf node connected to node u. *Link* represents the connecting portion in between two nodes.

Below is an example that illustrates the Prüfer number encoding with more detail. (i)$P_n=\{2\}$, $P_l=\{3\}$; (ii) $P_n=\{2,1\}$, $P_l=\{3,4\}$; (iii) $P_n=\{2,1,7\}$, $P_l=\{3,4,5\}$; (iv) $P_n=\{2,1,7,1\}$, $P_l=\{3,4,5,6\}$; (v) $P_n=\{2,1,7,1,8\}$, $P_l=\{3,4,5,6\}$; (vi) $P_n=\{2,1,7,1,8,2\}$, $P_l=\{3,4,5,6\}$; (vii)$P_n=\{2,1,7,1,8,2,7\}$, $P_l=\{3,4,5,6,9\}$. As shown in Fig. 3, the Prüfer number are $P_n=\{2,1,7,1,8,2,7\}$ and $P_l=\{3,4,5,6,9\}$.

2.2. Prüfer number decoding

(1): Given a string P_n of Prüfer numbers and its leaf node string P_l.
(2): Let x be the least bit in string P_l, and let y be the leftmost bit in P_n; add an edge to connect nodes x and y, and then remove x from P_l and remove y from P_n.
(3): Check if is y in P_n or not. If y longer exists in P_n, then add y into P_l, sort P_l and rearrange it from small to large; repeat the above-mentioned steps until not a single bit exists in P_n.
(4): By the time there is no single bit in P_n, the only nodes remaining in P_l should be nodes r and s and hence, these two nodes are connected to form a tree topology.
X: Representing the bit with the minimum number known in P_l.

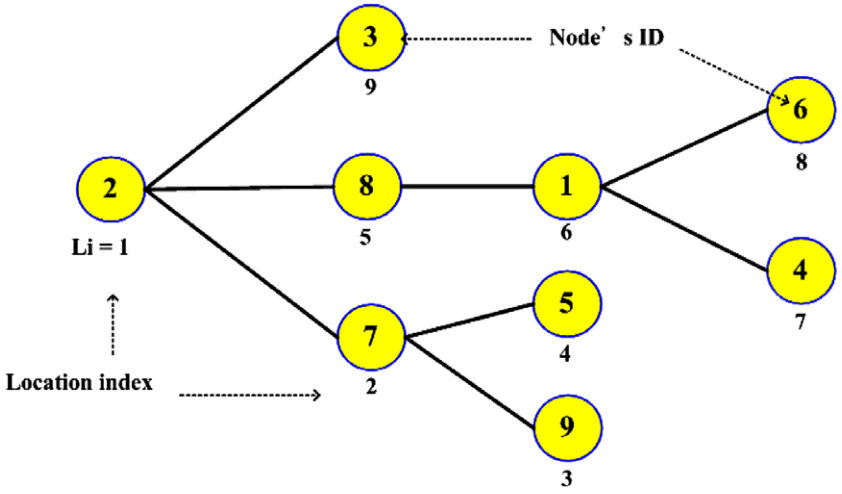

Fig. 3. An example of a multicast tree – MTree 1.