Wireless Pers Commun (2015) 85:2485–2505
DOI 10.1007/s11277-015-2916-y

A Hybrid Algorithm for Preserving Energy and Delay Routing in Mobile Ad-Hoc Networks

Mitra Ahmadi[1] · **Mohammad Shojafar**[2] ·
Ahmad Khademzadeh[3] · **Kambiz Badie**[4] ·
Reza Tavoli[5]

Published online: 23 July 2015

Abstract The Quality of Service (QoS) routing protocol plays a vital role in enabling a mobile network to interconnect wired networks with the QoS support. It has become quite a challenge in mobile networks, like mobile ad-hoc networks, to identify a path that fulfils the QoS requirements, regarding their topology and applications. The QoS routing feature can also function in a stand-alone multi hop mobile network for real-time applications. The chief aim of the QoS aware protocol is to find a route from the source to the destination that fulfils the QoS requirements. In this paper we present a new energy and delay aware routing method which combines Cellular automata (CA) with the Genetic algorithm (GA). Here, two QoS parameters are used for routing; energy and delay. The routing algorithm based on CA is used to identify a set of routes that can fulfill the delay constraints and then select a reasonably good one using GAs. The results of Simulation show that the method proposed produces a higher degree of performance than the AODV and another QoS method in terms of network lifetime and end-to-end delay.

Keywords Mobile ad-hoc networks · Quality of Service · Routing · Cellular automata · Genetic algorithm

✉ Mohammad Shojafar
m.shojafar@yahoo.com; mohammad.shojafar@uniroma1.it

[1] Department of Computer Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran

[2] Department of Information Engineering, Electronics and Telecommunication (DIET), "Sapienza" University of Rome, Rome, Italy

[3] Education and International Scientific Cooperation Department, Iran Telecommunication Research Center, Tehran, Iran

[4] IT Faculty, Iran Telecommunication Research Center, Tehran, Iran

[5] Department of Mathematics, Islamic Azad University, Chalous Branch (IAUC), Chalous, Iran

1 Introduction

A mobile ad-hoc network (MANET) has a dynamic topology without using any fixed infrastructures, and each node operates both as a host and as a router. Their most important features include Autonomous and no infrastructure, multi-hop routing, dynamic network topology, device heterogeneity, energy-constrained operation, bandwidth-constrained variable capacity links, and limited physical security.

The features of MANET have been used in many places, although it has many constraints. First of these features is high ability in situations where fixed infrastructure is not available. Another one is needless to operate in a stand-alone fashion, but can be attached to the Internet and integrating many different devices and making their services available to other users. MANET has been used in several applications. In tactical networks applications, we have military communications and operations. In emergency services, MANETs are used in search and rescue operations, policing and firefighting, supporting doctors and nurses in hospitals. Further, universities and campus settings, virtual classrooms and ad-hoc communications during meetings or lectures are MANET's usage in education field.

In MANET, it is very important to find a path from source to destination for delivering data packet that satisfies the Quality of Service (QoS) requirements, such as throughput, end-to-end delay, energy and so one.

Two major categories of MANETs routing protocols are *pro-active* (*Table-Driven*) *routing protocols* and *reactive* (*On-Demand*) *routing protocols* [1]. The *first one* maintains routing information in one or more tables, so it is called *Table-Driven*. This kind of protocol keeps the routes between each pair of nodes and up-to-date by using periodic broadcasts. It incurs additional overhead cost due to maintaining up-to-date information and as a result, throughput of the network may be affected. The most common protocol used for this routing is Highly Dynamic Destination-Sequenced Distance Vector Routing Protocol (DSDV) [2, 3]. In the *second one*, discovery process of a route is performed when it is required. Therefore, it does not store any information about the network, but makes a higher latency waiting. The well-known protocol used for this routing is Ad hoc *On Demand Vector* (*AODV*) [1] *and Dynamic Source Routing* (*DSR*) [4] that node only gather routing information on demand: when a data session to a new destination starts, or when a path which is in use fails. There are some protocols, which are the combination of proactive and reactive protocols taking the best features from both protocols. They are called *hybrid routing protocols*. The Temporally Ordered Routing Algorithm (TORA) [5] and Zone Routing Protocol (ZRP) [6], ANtHocNet [7] are examples for this type of protocols.

Metaheuristic algorithms provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information or limited computation capacity. Metaheuristic algorithms are used for combinatorial optimization in which an optimal solution is sought over a search-space such as graphs and networks. Nowadays, several scholars work on these optimizations on different networks, Wireless sensor networks, MANETS, peer-to-peer, and Grids such as [8–15] focusing in routing and saving the energy. Specifically, authors in [10] proposed a routing technique based on Ant-Colony optimization (ACO) to consider file types in order to search for file container nodes with a high probability of success and low-energy consumption QoS in WSN. In addition, maximizing lifetime of the sensors, is one the most important QoS parameters in the WSN which is the focus of the [9]. In detail, Authors present an efficient cover set algorithm based on Imperialist competitive algorithm to determine the sensor nodes that must be selected in different cover sets. As the presented algorithm proceeds, the cover sets are generated to monitor all deployed targets [9]. Authors in [11, 12] proposed soft-based methods based on joint of Fuzzy and imperialist competitive clustering algorithm which enhances the accuracy of malicious detection in WSN and deployed at the Intel Berkeley Research Lab.

Recently, there are numerous classical and artificial routing protocols proposed for mobile ad hoc networks. Most of them are unable to satisfy support of QoS requirements. As a result, QoS should be take into account in MANET. We are able to provide QoS routing protocols by expanding these protocols (i.e., Pro-active, reactive and Hybrid). In [16] authors added two extensions and a QoS flag to RREQ and RREP messages in AODV protocol to enhance QoS and load balancing features. In [17], EDTORA is an energy and delay aware protocol based on TORA that has a higher performance than TORA in terms of network lifetime, packet delivery ratio and end-to-end delay. Also, authors in [18] proposed energy-saving dynamic provisioning of routing in WSN named DDLA that combine directed diffusion with learning automata. They considered fault tolerance and rejoining based on their proposed workbenches to improve QoS. Authors in [19] proposed risk-aware approach to give clear selection criteria for practical application on industrial networks. Also, their approach is promising but lack of management energy and latency. Due to the fact that the nodes that take-part in transmitting have mobility and movement, this makes the MANET system be complex for routing. Briefly, MANET's routing is an NP-complete problem; therefore, it cannot be solved in polynomial time in any known way. This is the focus of this paper propose joint technique to enhance routing in MANET by hybrid metaheuristic approaches. In this paper, we propose a parallel and distributed routing algorithm based on Genetic algorithms (GAs) and cellular automata (CA) to discover a route that guarantees two QoS requirements, delay and energy. CA and GA are emerged as powerful tools for solving NP-complete constrained optimization problems by their characteristics. GA has the characteristics of high convergence speed while it mixes with CA cause increase the convergence speed and decrease the delay of routing.

The remainder of the paper is structured as follows. Section 2 describes the overview of CA and GA and some recent routing algorithms in MANET. Section 3 presents the proposed system model that contains parameters of QoS routing protocols and delay and energy parameter details. The proposed algorithm is presented in Sect. 4. The computational results are reported in Sect. 5. Finally, our concluding remarks and future work are represented in Sect. 6.

2 Related Work

In this section, we will overview some CA, GAand review routing algorithms that can support QoS in MANETs. In the following, we will briefly explain the CA and genetic algorithm.

2.1 Cellular Automata

CA is a discrete dynamic system in both time and space. This implies that the cells in the CA contain finite state automata, a finite value range, and based on a local update rule they change their states depending on the states of their neighbors. In fact all the cells simultaneously change their states, according to the same update rule. The process is repeated at discrete time steps.

This means, the state of a cell *i* at time t + 1 is affected by the states of its neighbors. Presuming a uniform rule, where each cell is affected by the same transition function, we can calculate the state of a cell *i* at time t + 1 using the following Eq. (1):

$$S_i^{(t+1)} = f\left(S_{i-1}^{(t)}, S_i^{(t)}, S_{i+1}^{(t)}\right) \tag{1}$$

where $S_i^{(t)}$ is the state of a cell i at time t and $f(\cdot)$ is the transition function that defines the rules governing the change of state of a cell. Although this formula is applicable for CA with a neighborhood radius of one ($r = 1$), it can be expanded to include more neighborhoods ($r > 1$) like Eq. (2):

$$S_i^{(t+1)} = f\left(S_{i-r}^{(t)}, \ldots, S_{i-1}^{(t)}, S_i^{(t)}, S_{i+1}^{(t)}, \ldots, S_{i+r}^{(t)}\right) \tag{2}$$

2.1.1 Formal Definition

In formal definition CA is a 4-tuple (L, S, N, f) where [20]:

L is a regular grid. The elements that compose this grid are called cells.
S is a finite set of states, $S = \{0, 1\ldots, \ s-1\}$.
N is a finite set (of size $|N| = n$) of neighborhood index, such that $\forall c \in N$ and $\forall k \in L : k + c \in L$

$f : S^n \rightarrow S$ is a transition function where n is the size of the neighborhood.

Cellular automata is composed of an n-dimensional lattice of identical and synchronous finite state machines. State S is updated (synchronously) in accordance with the transition function (or transition rule) f, implying that all the cells of the automata are updated synchronously, in parallel.

The neighborhood refers to the set of cells that can directly influence a particular cell. In homogeneous cellular models, all the cells have the same shape. Two common and popular neighborhood types include the Moore neighborhood and Von Neumann neighborhood. In a two-dimensional cellular grid with neighborhood radius = 1, the von Neumann neighborhood of a cell *i* includes four cells orthogonally touching it (Fig. 1a); in a cell *i* of the Moore neighborhood there is a set of eight cells which share a vertex or edge with that cell *i* (Fig. 1b).

A configuration $C_t : \longrightarrow S$ is a function that associates one state with one cell of the grid. It is the task of this function to change the configuration C_t to a new one C_{t+1} [Eq. (3)].

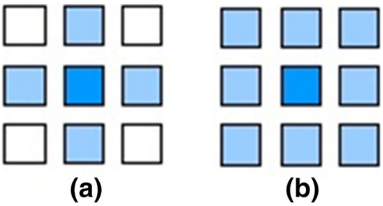


Fig. 1 **a** Von Neumann neighborhood, **b** Moore neighborhood in two-dimensional CA

$$C_{t+1}(k) = f(C_t(i)|i \in N(k)) \tag{3}$$

where N(k) is the set of all the neighbors of the cell k, as given in Eq. (4):

$$N(k) = i \in L|k - i \in N \tag{4}$$

2.2 Genetic Algorithm

The GA, a search heuristic that mimics the natural evolution process, is routinely used to produce useful solutions for problems of optimization and search. Genetic algorithms are categorized under the larger class of Evolutionary Algorithms (EA), which generate solutions to optimization problems utilizing the methods drawn from natural evolution, including those of inheritance, mutation, selection, and crossover. Developed by Holland [21] this technique is useful when the solution space to be searched is extensive. This results in sequential search computationally becoming expensive and time consuming. GA-based algorithms are powerful tools for approximating the NP-complete constrained optimization issues.

2.2.1 GA Cycle

A GA begins with a population, which includes a set of randomly generated possible solutions. Each individual solution, otherwise called a chromosome, may be represented either as a simple string or an array of genes which contains a part of the solution. Each gene is termed an allele. Each chromosome in a population should be of the same length. Each chromosome is assigned a fitness value using a fitness function based on the proximity of a chromosome to its optimal solution.

Two randomly selected chromosomes, termed “parents”, are capable of exchanging genetic information through recombination or crossover, and thus produce two new chromosomes called “child” or “offspring”. If both parent chromosomes share a particular pattern in their chromosome, then the same pattern will be passed on to the offspring. To achieve a good result, post crossover, a mutation is often applied on randomly selected chromosomes. Mutations facilitate the restoration of any lost genetic values when the population converges at a very quick rate. The chromosomes for the next generation are selected after crossover and mutation processing. To maintain the new generation at a level of fitness at least as fit as the previous generation, a few of the chromosomes in the current generation having the lowest performance can be replaced by the same number of the best performing chromosomes from the previous generation. This process is called elitism and the whole cycle is repeated until the stopping criterion of the algorithm is satisfied.

2.3 Review Related Work

Although there are many heuristic and artificial QoS routing algorithms proposed in MANETs, only a few of them will be discussed as follows:

2.3.1 Routing Algorithms Based on CA

It is clear that the CA is capable of resolving many complex systems (like MANET) and issues (routing in particular, in this study) in different applications, one of which is routing. A well-known approach used in routing based on CA is the Lee algorithm [22], which identifies the path having the lowest sum of weights, on the grid. If the weight of all the grid points (nodes) is set to one, the algorithm will find the shortest path between the source cell and destination cell. The authors, in their framework, consider that the number of states is related to the longest path or to the greatest accumulated weight that can occur. In [23] the authors modified the Lee algorithm by utilizing a finite number of states, and rather than storing the accumulated weights they stored the directions of the nodes in the path when a move to return to the starting point is essential. They mapped a classical algorithm onto the cellular processing model.

Finally, in [24] several writers who employed the modified Lee's Algorithm [23] presented a routing method which fulfilled the delay constraint as the QoS parameter. The source node broadcast the QRREQ message with the maximum delay to its neighbors. This algorithm works with fifteen states. Right at the beginning, all the cells, except the source, the destination and dead cells, are in the INIT state. Based on the algorithm, first the nodes in the neighborhood of the source cell become a wave. During the process, all the cells verify if there is any cell in the neighborhood that already has a wave node. If it finds a wave node, then the NODE_TRAVERSAL_TIME (NTT) compares with the maximum delay. If the NTT is less than the maximum delay in the message, the intermediate cell itself gets changed into a wave node and continues broadcasting the QRREQ. Else, if the NTT is greater than the maximum delay, it implies that this node does not satisfy the delayed constraint in the path and, therefore, it must discard the QRREQ, and does not become a wave node. This process continues until the wave reaches the destination cell. Now a path has been discovered. The destination may receive multiple QRREQ messages from the same sender. However, the destination replies to the first QRREQ message by sending a QRREP message utilizing the reverse path set up when the QRREQ messages are first forwarded. As this path is the shortest one between the source and destination, it represents the delay constraint. All the wave nodes in this path get changed to path nodes and until the topology of the network changes all the data transfer that occurs between this source and the destination takes place via this route. The algorithm gets terminated when the source cell recognizes that one of its neighbors has become a part of the path. The source node changes its state from INIT_S to FOUND and thereafter signals that the path is complete. From here on this method is referred to as the *base-QoS-method* in the research.

2.3.2 Routing Algorithms Based on GA

Genetic Algorithms have been shown to be an effective technique in optimizing and stochastic searching. A GA-based heuristic algorithm to resolve the bandwidth-delay-constrained least-cost multicast routing problem has been put forward in [25]. The authors have proposed the connectivity matrix of the edges encoding a scheme to represent the genotype. The penalizing strategy in a fitness function to handle the infeasible chromosomes is used [25]. Besides, they proposed a few new crossovers and mutations in the repairing strategy to deal with the illegal chromosomes and an avoidance strategy to prevent creating illegal chromosomes in them.

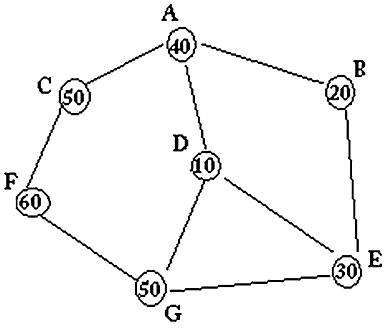
In [26] the authors suggested a new QoS routing approach for MANETs based on a GA which involves two parameters to satisfy the QoS path, namely, the delay and transmission success rate. This algorithm's main requirement is "robustness, rather than optimality". The GAMAN [26] protocol has the following features: (1) This is a source-based routing algorithm, (2) it utilizes a small population size, (3) the nodes in the subpopulation are concerned with the routes only within this subpopulation, (4) the information is transmitted only for the nodes in a population, thus avoiding the broadcast, (5) different routes are searched and sorted by ranking. The first one is used as the best option while the other ranked routes can be used as back-up routes, (6). By utilizing the tree-based GA method the loops can be avoided. GAMAN supports the soft QoS without hard guarantees. This implies that when the QoS required is not guaranteed, transient time periods may exist. Also, the QoS required should be provided when the established routes remain interrupted. The GAMAN algorithm proposed can be used for small and medium scale networks. The main aim is to quickly identify the efficient routes and the response to the dynamics of the network very fast [26]. Some recent research has been done with the genetic or hybrid approach in MANET routing such as [27–32]. Specifically, the authors in [27] propose a hybrid routing intelligent algorithm with an Ant Colony Optimization (ACO) algorithm and Particle Swarm Optimization (PSO) to refine the various metrics in MANET routing [27]. The convergence time and space are less, because in our proposed methods, we scale down the environment space and reduce the population for the genetic algorithm, so that the time and cost will be decreased. In [28], the authors suggest a (distributed hash table) DHT-based routing protocol that exploits a 3-D logical space that considers the physical intra-neighbor relationships of a node and exploits a 3-D structure to explain that relationship. This approach is not concerned about the parameters we wish to control. Moreover, according to [7] the authors, their approach called AntHocNet is based on concepts from the Ant Colony Optimization. It involves both reactive and proactive components. In the reactive path set-up phase, several paths are set up between the source and destination of a data session, and in the course of that communication session, the ants proactively test the existing paths and explore new ones. The approach thus proposed pays close attention to the neighbor zones and sufficiently includes local search [in particular, it considers these facts in the first cycle (CA)]; however, AntHocNet, is not based on the zone routing framework.

3 System Model

The following section explains the model proposed by describing the QoS constrained and its various features. Currently, the support of QoS has become the greatest need, because it affects high-speed communication systems and increases the use of distributed multimedia applications, like traffic control systems, video conferencing, or telemedicine systems which exhibit communication patterns.

Some assumptions have been included in our model. One assumption is that all the hosts communicate on the same shared wireless channel. Each node has a unique identifier with

Fig. 2 An example of QoS in MANET



at least one transmitter and one receiver. We also assume that the effective transmission distance between every node is equal. Two nodes are assumed to be neighbors with a link between them if they are in the transmission range of one another. The presence of a neighbor discovering protocol is another assumption. At intervals each node transmits a BEACON packet identifying itself; therefore, every node recognizes a set of neighbor nodes. We assume the presence of a MAC protocol, which resolves the media contention, encourages resource reservation and ensures that within the group of neighbors in the local broadcast range only the intended receiver retains the message while the other neighbors discard it.

The QoS type is correlated to the routing. The example discussed clarifies the way the QoS selections account for the routing. Figure 2 shows some mobile nodes in a MANET, labeled A, B… G. The number in each node indicates the residual battery energy for each one. If the number of hops is considered as a QoS metric for finding a path from node A to node G, path "A–D–G" will be selected as the best path from source node A to destination node G. But, if we consider residual battery energy as a QoS metric and wish to find a path from A to G with longer lifetimes, the best path will be "A–C–F–G". Therefore, the type of QoS has an effect in routing while the nodes participate in the transmission.

The main objective of the QoS routing protocol is to select a path from the source to the destination that meets the requirements of the desired QoS. The QoS constrained routing algorithms must be capable of finding a path with sufficient resources in order to fulfil the QoS requirements of a flow. The QoS constraints could include available bandwidths, cost, end-to-end delay, delay variation (jitter), battery energy, etc. [25, 26, 33]. From our model, it is evident that a QoS routing consists of, at least, two QoS metrics, namely delay and energy.

We assume that a mobile ad-hoc network (MANET) includes N mobile wireless devices (nodes) distributed in a given geographical region. The geographical region where the nodes are located is viewed as a logical two-dimensional (2D) grid of cells. Two grid cells are referred to as neighbors if they share a common side. Therefore, each grid cell has four neighboring cells (Left, Right, Up, Down) except for those cells located on the boundaries of the grid which have two (corners) or two neighboring cells each. A path in the 2-D grid is in fact a sequence of the neighboring grid cells. Therefore, a node found in cell (x, y), stores in an array variable List of Gateways the IDs of the eight gateway nodes located in the neighboring grid cells as follows: Right: (x + i, y), Left: (x − i, y), Up: (x, y + j), Down: (x, y − j). Two MANET nodes are termed neighboring nodes if they are within each other's transmission range. Each grid cell is identified by a pair of integer grid coordinates (x, y) as depicted in Fig. 1. Each MANET node has a unique node id (IP or MAC address). We assume that the nodes are able to receive their geographical positions through a low-power GPS receiver.

3.1 Delay

Delay is a significant performance characteristic of telecommunications networks. It indicates how long it takes for a packet to travel across the network from the source to the destination. In a network, packet delay comprises several aspects: (1) *Processing delay*: time taken to process the packet header, (2) *Queuing delay*: time taken by the packet in routing queues, (3) *Transmission delay*: time taken to push the "bits" of the packets into the link, and (4) *Propagation delay*: time taken for a signal to reach its destination. Therefore, the end-to-end delay of a path is the sum of the node delay at each node in addition to the link delay at each link on the path.

In this paper, the RREQ includes a maximum delay extension. This field will be set to delay requirement denoted as D_{max}. The goal is to discover a path from the source node ***s*** to the destination node *d* that the delay time of this work does not exceed the value of D_{max}. To control the RREQ propagation with the delay metric we considered the following assumptions:

Set node traversal time to 40 ms.

If the node traversal time is longer than or equal to the maximum (remaining) delay in the delay field of RREQ, the intermediate node must discard the RREQ.

If the node traversal time is less than the maximum (remaining) delay in message, the intermediate node must send a RREP to the originator or continue to broadcast the RREQ.

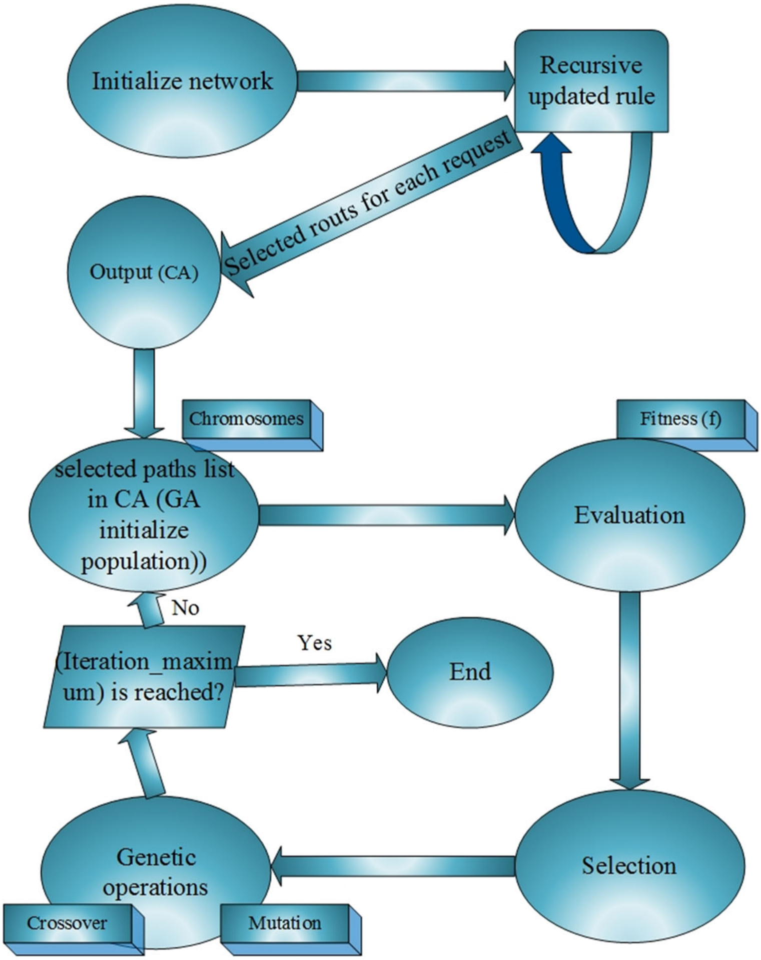


Fig. 3 Steps in proposed method

3.2 Energy

In protocols like the on-demand protocols, the shortest path is found via the route discovery process and is used until it breaks. It reduces the energy of the nodes on this route. When a node loses its energy and cannot forward any messages, it will fall out of the network. This action has a negative effect on the lifetime of the ad-hoc networks. Therefore, one of the objectives of the routing protocol is to search among the nodes having high battery energy. This will extend the network lifetime. Some percentage of the initial energy is considered the energy constraint. In our algorithm, this energy is called *minEnergy*. It is assumed that the initial energy of the node is the maximum energy supplied by the battery when it is fully charged [17].

4 Proposed Method

We propose a methodology that will enable the discovery of routes in MANETs which satisfy both the delay constraints and some simple energy constraints (every node on a path has a minimum energy level). In this paper finding a route in MANET, must include one which guarantees two QoSs parameters; delay and energy. In this work we define a routing method that comprises the CA and GA techniques (Fig. 3).

The mapping of CA with the MANET topology is explained to understand precisely the details of the implementations. To locate a path with delay as the QoS constraint, we assume a two dimensional network plane which includes many mobile nodes spread randomly. No hierarchy is found between the nodes, and the network plane is completely homogeneous (with all the nodes having the same properties). Data movement between two cells is termed one hop. Every node represents one cell of the cellular automaton. Each cell has a state corresponding to the status of the node that occupies it. The neighborhood of each cell for identifying the route is the range of communication of each node otherwise called the von-Neumann neighborhood governed by a circle of radius 1 around the cell. The network plane also includes dead cells which block communication. They represent physical obstacles (like high-rise buildings) or simply a missing communication link. Two nodes with a dead cell between them cannot communicate directly with each other. As the network nodes can move randomly to any of the neighboring cells the Moore neighborhood is used for mobility.

In our method, the source node broadcasts the QRREQ message with the delay requirement of the connection request [maximum delay (D_{max})], to its communicating neighbors. They include all the nodes at the top, down, left and right side.

The wave nodes then re-broadcast the message to their neighbors, and also set up a reverse path to the sender. Some nodes further, if provided with a delay constraint, include a wave node to their neighbors, become a wave, re-broadcast this message, and set up a reverse path to the nodes from which they had received the message.

This process goes on until the message is received by the destination node or the delay experienced by the packet exceeds the limit D_{max}. If there is more than one path from the sender to the destination, the destination may receive multiple QRREQ messages for the same sender. Therefore, the destination will reply to some of the QRREQ messages by sending a QRREP message using the reverse path set up when the QRREQ messages are forwarded. All the wave nodes found along these routes, between the source and the destination, become the path nodes. All communications between the source and the destination from this point onwards occurs via this path until the topology of the network changes. All the other wave nodes that do not find a path node in their neighbor pointing towards them are given a clear state message to move them from the wave state to a clear state.

The local states include the following:

The INIT (initial state of each node barring the source and destination), Dead (Dead Cell), INIT_S (Initial State of the Source node), INIT_D (initial state of the destination), WU (Wave Up), WD (Down), WR (Right), WL (Left), PU (Path Up), PD (Path Down), PR (Path Right), PL (Path Left), CLEAR (the final state of the node which received a wave message but is not likely to become a path node) and FOUND (route found; final state of the Source node). Keeping these set of states in view, we consider the von-Neumann neighborhood in which each cell can communicate with all the cells on the top, down, left and right. The local transition rules for each cell based on these neighborhoods are drawn up in accordance with the routing mechanism.

To implement mobility, we assume that the nodes can move only to the neighboring cells. A Moore neighborhood is selected because it allows movement in more number of directions than does the von Neumann neighborhood. As we used a two-dimensional grid, each node can have no more than eight nodes as neighbors. Therefore, in this experiment, a node can, with deterministic speed, move around in random fashion to any one of each of the surrounding cells, by selecting any direction, i.e. North, North-East, North-West, East, West, South-West, South-East and South. Cellular automata being discrete event time systems, the nodes, in each time step, move to one of the neighboring cells and the routing algorithm is performed. Execution of the mobility rules is considered higher than the routing rules, which implies that the first topology is changed and then the process of finding the route between the source and destination is done. Therefore, to find the path having maximum energy, the GA algorithm is applied on these paths.

One of the objectives of this study is to exploit the inherent parallelism of the CA and GA apart from the optimal solutions. To sum up, our goal is to quickly compute good routes in keeping with the CA rules, and select the best one with respect to the energy minimization and the dynamics of the network very fast. Therefore, we closely watch for the optimality of routes and robustness. We will satisfy the delay of the system using the CA approach and energy consumption by the GA method. In our approach, we utilize two cycles—first, in our initial population, we try to locate paths that respect the quality of the service of delay for that request, and then, from among the selected paths, we attempt to select the best one based on the genetic optimization approach with respect to the fitness function that accounts for the energy control parameter for every single path. We assume the occurrence of frequent changes in network topology, although they are not frequent enough to make any sort of route computation useless. Specifically, we assume that the topology changes are typically followed by at least a short period of stability. It must be noted that our concern is limited to the relative mobility of the hosts and not to the absolute mobility of the hosts.

In the CA cycle, as shown in Fig. 3, when the request arrives, the situation of the current MANET nodes will be to initialize. Then based on the first QoS parameter (minimize the delay), they will find some paths for the next cycle. The execution time for this cycle will be related to the type of request but not more than O (N.N) (i.e., N is the number of nodes in the graph), because, the CA works according to the rules and finding the shortest paths is similar to Dijkstra's algorithm. In particular, it initializes all the cells with specified initial conditions. For each cell, it determines what it should become in the next time step, in line with the states of its neighbors. Then, it updates all the cells simultaneously until it finds the suitable paths.

The GA cycle as shown in Fig. 3 begins with an initial population of potential solutions which is created as a starting point for the search (the selected routes arise from the CA rules). Next, the performance (fitness, f) of each individual is evaluated in light of the constraints imposed by the problem. Based on the fitness of each individual, a selection mechanism selects "parents" for the crossover and mutation operators. The crossover operator picks up two chromosomes and swaps a bit of their genetic information to produce new chromosomes. The mutation operator then introduces new genetic structures into the population by the random modification of some of the genes, and helps the search algorithm to escape from the local optimum.

The offspring resulting from the genetic manipulation process become the next population to be evaluated. GA can replace either an entire population or its less fit members. The creation-evaluation-selection-manipulation cycle is repeated until a satisfactory solution to the problem is found or some other termination criteria is met. The runtime of this cycle is similar to the first cycle.

4.1 Discovering Initial Routes

The main plan in this study is based on the CA. We apply two-dimensional CA to identify some of the shortest paths that fulfil the delay constraint. To achieve this goal, we use the base-QoS-method (proposed algorithm in [24]) with some variances. Therefore, in the initializing step from the list of gateways for all the nodes, each node sends out a "HELLO" message to its neighborhood. The presence of a neighbor node is ascertained when a "HELLO" message is received. When a node receives a "HELLO" message it selects the latest message heard in a specific direction as a gateway for that particular direction. Therefore, now, the nodes recognize their neighbors and try to communicate with them in the future. We assume that, no failure is possible for each node and all the nodes are available in system.

Each cellular unit in the CA is associated with each node in the network. As we use the von Neumann neighborhood each cell can have a maximum of four neighbors. Further, each cell can possess one state, which relates to the status of the node that occupies it. In this experiment, the local neighborhood for the cells is of radius $r = 1$. Therefore, we use Eq. (1) as a CA transaction function.

In the base-QoS-method, there is only one shortest path that can be the best for the delay parameter while there are other parameters that have significant parts to play in the mobile ad-hoc networks, such as Energy, Bandwidth etc. In some applications, for example, in Military Applications, we utilize mobile wireless devices that process for long time periods without power charging. Thus, by making some changes in the base-QoS-method we will generate several short paths that fulfil the delay constraint.

To achieve this result, some rules need to be changed. We should not stop discovering paths after identifying the shortest one. Then suitable paths will be discovered by using the GA technique.

4.2 Discovering an Efficient Route

Here, we use GA to discover the way in which the CA finds the best path through those mentioned in the section above. A general structure of the proposed algorithm is as follows in Algorithm 1:

Algorithm 1: Proposed structure of the genetic algorithm joint with CA

```
Input: Population of Paths, which are found by CA rules, P.
Output: Selected Paths respect both QoSs
Begin
    Initialize the population, P and a mating pool $P_{next}$
    Evaluate P.
    While stopping conditions not true do
    Sort P in descending order of fitness.
    Apply Selection from the worst n ($1 \leq$ n$\leq$ population size) paths in P to $P_{next}$.
    The best paths in P (of size population size – n) are left untouched and survive to the next generation intact.
    Crossover $P_{next}$.
    Mutate $P_{next}$.
    Copy the n paths of $P_{next}$ to P, replacing the worst n paths in P.
    Evaluate P.
End.
```

Initially, a population of paths is created. In our case, the population consisted of 20 chromosomes (each representing a path). Each path contains some nodes, each of which has an energy value. This initial population is then evaluated and the energy value of each node in the path is examined. If, in a path, a node with an energy value less than the *minEnergy* is identified, the fitness value of that path will be set at 0. This implies that this path does not offer an energy constraint and therefore should not move on to the next population. Otherwise, if all the nodes have sufficient energy then we can count the number of nodes whose values are greater than that of the *maxEnergy* value and divide this number by the total number of nodes in the path (The value of *maxEnergy* is set to half of the maximum energy provided by the battery when it is fully charged). The fitness value of this path refers to the number which results from this computation. Therefore, the paths that give higher values are better than those which generate lower values. The fitness value is calculated as shown in Eq. (5).

$$f = \begin{cases} 0, & \text{if } \exists E_i : E_i < \text{minEnergy} \\ \frac{m}{n}, & \text{if } \forall E_i : E_i \geq \text{minEnergy} \end{cases} \tag{5}$$

where f is the fitness function, E_i the energy value of each node, *minEnergy* the value of the energy constraint, and m is the number of nodes in the path in which their energies are higher than *maxEnergy*, and N is the number of all the nodes in the path.

Algorithm 1 states that after the initial evaluation, the GA goes into the "evaluation loop". The GA continues to evolve until the stopping criteria are fulfilled. This condition can be the maximum computation time, path similarity or the number of generations. In each generation, the chromosomes/paths in the population are sorted out in the descending order, based on their fitness (i.e., Sort instruction in the Algorithm 1). In this experiment, the best single path is retained intact and passed on to the next generation. For the rest of the 19 chromosomes/paths, a selection, crossover and mutation operators are applied. Tournament selection is applied to the 19 chromosomes, using the Roulette Wheel to create a mating pool of 19. To the chromosomes in the mating pool, a one-point crossover with a probability of 0.8 and a gene mutation operator of a probability of 0.001 are applied. These 19 chromosomes are then copied to the current population P, which, together with the best one single chromosome, forms the next chromosome generation. After 100 generations, we will obtain some paths that satisfy the energy constraint. In fact, these paths satisfy both the energy and delay constraints. The path with the greatest fitness value is used as the desired path while the other ranked paths can be used as back-up ones.

5 Performance Evaluation

In the following section, the experimental set-up and metrics for our simulation study are discussed.

5.1 Experimental Setup

MATLAB has been used as the simulator to analysis and compare the models. It is a software package for high performance numerical computation and visualization. It provides an interactive environment with hundreds of built in functions for technical computation, graphics and animation. MATLAB provides very good tools in many types of scientific computations, such as data analysis, signal processing, optimization and numerical solution of ordinary differential equations. The simulation has been done for a rectangular network with 625 mobile nodes in an area of 3000 in 3000 m^2 and at different pause times and speeds. We considered two scenarios on AODV protocol, base-QoS-method protocol and our method for comparing. Briefly, our approach compares to AODV, provides routing structure but AODV does not build it. Moreover, our approach find multiple routes, AODV does not. The similarities of our approach compare to AODV, both of them consider beacons distributing (i.e., HELLO messages), both of them find the shortest path based on the weight of the graph's edges. Data traffic is generated by 20 constant bit rate (CBR) sources sending one 64-byte packet per second. Nodes move according to the random waypoint model, with a minimum speed of 0 m/s, a maximum speed of 25 m/s.

In the first scenario, we evaluated the effect of changing the topology (possibly, mobility rate). To analyze the effect of mobility, the pause time changes from zero to 600 s based on the standard mobility models (e.g., random waypoint model [4]). Pause time is a time in which all nodes in network are motionless but transmission continues. Pause time 0 s and pause time 600 s represent respectively high mobility and no mobility.

In the second scenario, we examined effect of node speeds (possibly, mobility) on the performance of the routing. Node speeds are 5, 10, 15, 20, and 25 m/s (similar to [4]). Furthermore, the pause time is 60 s in this scenario. We used the same setup as in the scalability study of AODV performed in [34].

For both scenarios, the QoS constraint is set to 0.3 and 0.1 s for delay and 30 % of initial energy for energy constraint. The initial energy for each node is set to 20 J. Therefore, we provided the threshold for each mobile node as residual energy that prohibit us from applying the node in routing, denotes *minEnergy* will set to about 6 J. The new algorithm is introduced with and without QoS parameters. The AODV and base-QoS-method are also simulated to be compared with the new protocol.

5.2 Experimental Metrics

The performance metrics, used to evaluate the performance of the routing protocols, are defined as follows.

5.2.1 Packet Delivery Ratio

This refers to the ratio between the data packets received by the destination and the total number of packets originally sent out by the source. This ratio presents the effectiveness of the protocol in delivering data packets to receiver [in Eq. (6)].

$$\mathrm{F} = \frac{1}{\mathrm{C}} \sum_{\mathrm{f}=1}^{\mathrm{C}} \frac{\mathrm{R_f}}{\mathrm{T_f}} \tag{6}$$

where F is the fraction of the packets delivered successfully, C is total number of connection flows, f is the unique flow id, $\mathrm{R_f}$ is the number of unique packets received from flow f and $\mathrm{T_f}$ is the number of packets transmitted to flow f.

5.2.2 Average End to End Delay

This includes all the likely delays due to buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, propagation and transfer times [in Eq. (7)].

$$\mathrm{D} = \frac{1}{\mathrm{S}} \sum_{i=1}^{\mathrm{S}} \mathrm{r}_i - \mathrm{S}_i \tag{7}$$

where D is the average end-to-end delay, S is the number of packets successfully received, *i* is the unique packet identifier, $\mathrm{r_i}$ is the time at which a packet with its unique identifier *i* is received, and $\mathrm{S_i}$ is the time at which a packet with the unique identifier *i* requests a route to be sent.

5.3 Experimental Results

Figure 4 shows the simulation results in which the average delay of the three protocols decreases as time increases. By using constraints, ours and the base-QoS-method reveal better average end-to-end delay values than the AODV because they force the network to satisfy certain delay constraints; else, the packets are discarded. In higher delay bound scenarios ($\mathrm{D_{max}} = 0.3$) our method shows results closer to the AODV protocol. The low end-to-end delay of the packets ensures the on-time transmissions required by real time traffic transmissions.

Figure 5 shows the values of the end-to-end delay for AODV and both the QoS methods (ours and the base-QoS-method) at different node mobility. The end-to-end delay is observed to increase as the node speed increases because higher mobility results in more links being broken. The delay is still better than that required for the QoS methods.

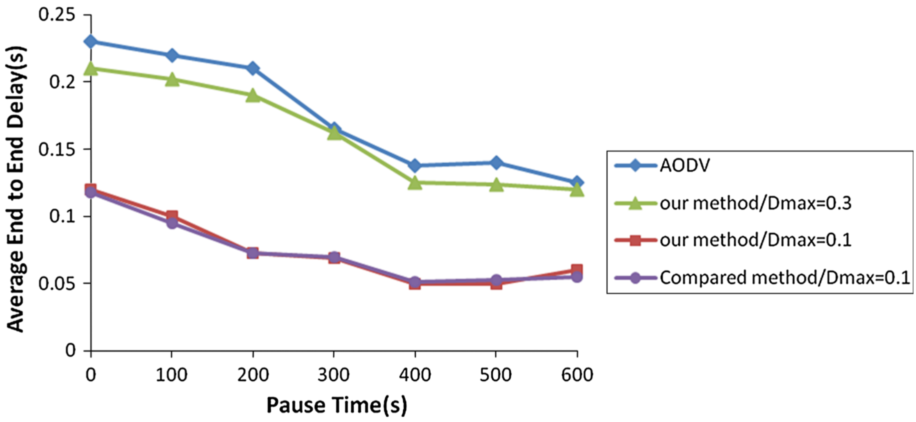


Fig. 4 Effect of pause time (s) rate on average end-to-end delay (s) (1st scenario)

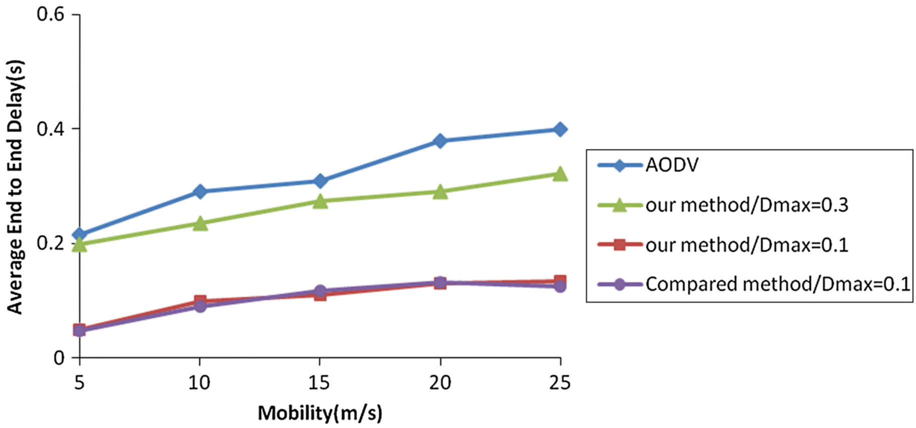


Fig. 5 Effect of mobility (m/s) rate on average end-to-end delay (s)

Simulation results also show that both QoS methods reveal comparable results in end-to-end delay metric. (i.e., the first scenario).

Figures 6 and 7 show the packet delivery fraction for the AODV as well as ours and the base-QoS method for different mobility rates and pause times. As mentioned prior, during the higher pause time at the lower speed, we observe less mobility. These are the reasons which cause fewer numbers of routes to break. Additionally, in Fig. 6 by increasing the pause time and in Fig. 7 by decreasing the speed, the results of the mobility in each method are decreased. (i.e., the second scenario).

Ours and the base-QoS-method reveal low performance with a low packet delivery fraction because the routes available for them do not satisfy the QoS requirements and packets get dropped. Simulation results also indicate that the performance of these methods drops in providing QoS assurance and they perform less when they have less D_{max}. This is because in the case of higher D_{max} there is ample time to find a suitable path and more data

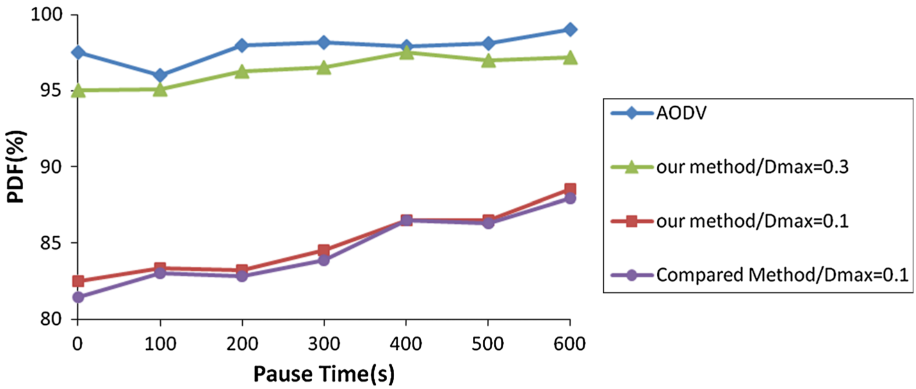


Fig. 6 Effect of pause time (s) on packet delivery ratio (PDF)

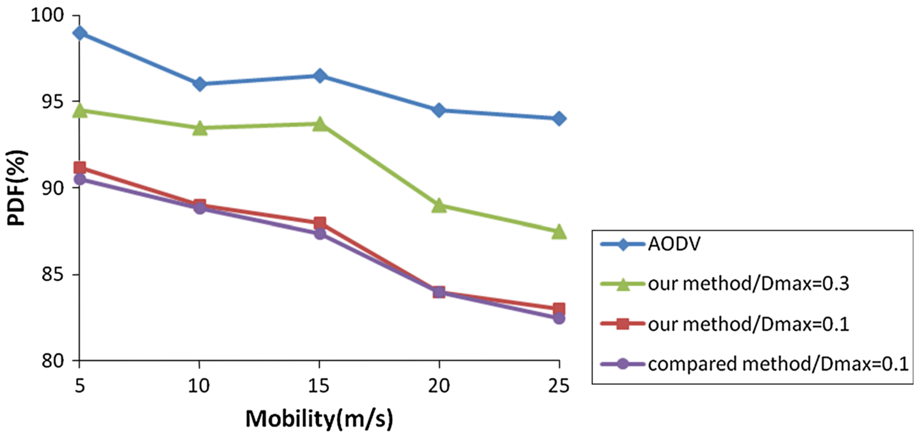


Fig. 7 Effect of mobility (m/s) on packet delivery ratio (PDF)

packets are received by the destination. As our method finds more paths to transfer packet data, it shows a higher ratio rather than the base-QoS-method (Figs. 6, 7).

Simulation results also reveal that when the mobile nodes moved at higher mobility speeds, our method with a high delay constraint has an average of 88 % packet delivery fraction, a value almost similar to that of the base-QoS-method (with high delay constraint) (Fig. 7).

5.4 Node Lifetime

Figure 8 indicates that the nodes of the AODV and base-QoS-methods die earlier than those in our method. This is because in our method the intermediate node compares its available energy to the *minEnergy* (the value of energy constraint). If the energy required is unavailable, the packet is discarded (i.e., in both the scenarios mentioned).

Figure 8 also proves that with our method the best nodes can be found to make the path by using GA and CA. Short paths with high-energy nodes are destroyed later.

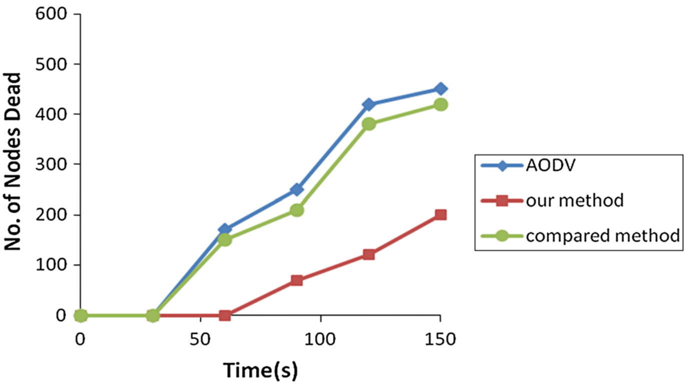


Fig. 8 Number of nodes dead versus time(s)

6 Conclusions

MANET applications are widely applied in almost in every field including military and commercial applications. Therefore, it becomes crucial to identify a path from the source to the destination to deliver a data packet that fulfils the requirements of QoS. In this paper, we proposed a QoS routing algorithm in MANET that satisfies the delay and energy constraints (i.e., able to support the two QoS parameters of delay and energy). Employing both CA and GA techniques we attempted to improve the network lifetime and end-to-end delay and compared our algorithm with the AODV protocol. AODV is a well-recognized protocol among the existing routing algorithms. Simulation results revealed better performance with our algorithm with respect to the average end-to-end delay and lifetime of the nodes than the AODV protocol; however, it showed less efficiency in the packet delivery ratio. This was due to the time it tries to satisfy the QoS constraints which resulted in discard packets. The simulation results also showed that our algorithm and base-QoS-method were comparable with the average end-to-end delay results; however, our algorithm showed better performance in the packet delivery ratio and node lifetime.

One of the objectives of this work was to exploit the inherent parallelism of the CA and GA rather than the optimal solutions. It is more important to achieve a stable path than a quick one. Therefore, the algorithm proposed was useful in MANETs, involving low mobility topology such as warfare networking in which devices are not required to move quickly. In this network, route stability is of greater value. Further research is warranted, in particular, a thorough examination of some methods of enhancing the packet delivery fraction or utilizing other QoS constraints like bandwidth to evaluate our algorithm.

References

1. Royer, E. M., & Chai-Keong, T. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications, 6*(2), 46–55.
2. Perkins, C., & Bhagwat, P. (1994). *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*. In Proceedings of of ACM SIGCOMM conference (pp. 234–244).
3. Perkins, C. E., & Royer, E. M. (1999). *Ad hoc on-demand distance vector routing*. In Proceedings of WMCSA '99. Second IEEE workshop on (pp. 90–100).

4. Johnson, D. B., & Maltz, D. A. (1996). *Mobile computing, chapter dynamic source routing in ad hoc wireless networks* (pp. 153–181). Norwell, MA: Kluwer.
5. Park, V. D., & Corson, M. S. (1997). A highly adaptive distributed routing algorithm for mobile wireless networks. In Proceedings of INFOCOM _97 (Vol. 3, pp. 1405–1413).
6. Moussa, M. I. & Badr, E. M. (2013). A new parallel algorithm for computing Minimum Spanning Tree. *International Journal of Soft Computing, Mathematics and Control*, *2*(2).
7. Caro, G. D., Ducatelle, F., & Gambardella, L. M. (2004). AntHocNet: An ant-based hybrid routing algorithm for mobile ad hoc networks. In *Parallel problem solving from Nature—PPSN VIII* (Vol. 3242, pp. 461–470). Springer
8. Hajj, H., et al. (2014). An algorithm-centric energy-aware design methodology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 22*(11), 2431–2435.
9. Mostafaei, H. & Shojafar, M. (2015). A new meta-heuristic algorithm for maximizing lifetime of wireless sensor networks. *Wireless Personal Communications*, *82*(2), 723–742.
10. Roy, B., et al. (2012). Ant colony based routing for mobile ad-hoc networks towards improved quality of services. *Journal of Emerging Trends in Computing and Information Sciences, 3*(1), 10–14.
11. Shamshirband, Sh, et al. (2014). D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks. *Measurement, 55*, 212–226.
12. Shamshirband, Sh, et al. (2014). Co-FAIS: Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks. *Journal of Network and Computer Applications, 42*, 102–117.
13. Lyashenko, V., Victoria, C., Drugarin, A., & Ahmad, M. A. (2015). Algorithmic research and application using the rayleigh method. *International Journal of Science and Research (IJSR), 4*(4), 1669–16771.
14. Aimad, A., et al. (2014). Robust sensorless sliding mode flux observer for DTC-SVM-based drive with inverter nonlinearity compensation. *Journal of Power Electronics, 14*(1), 125–134.
15. Ebrahimi, M., et al. (2012). Adaptive reinforcement learning method for networks-on-chip. In 2012 International conference on Embedded Computer Systems (SAMOS) (pp. 236–243). IEEE.
16. Mungara, J., Setti, S. P., & Vasanth, G. (2009). New model for quality of service in mobile ad hoc network. *IJCSNS*, *9*(12), 174–180.
17. Asokan, R., & Natarajan, A. M. (2008). An approach for reducing the end-to-end delay and increasing network lifetime in mobile adhoc networks. *World Academy of Science, Engineering and Technology*, *48*.
18. Sayyad, A., Shojafar, M., Delkhah, Z., & Ahamadi, A. (2011). Region directed diffusion in sensor network using learning automata: RDDLA. *Journal of Advances in Computer Research, 1*(3), 71–84.
19. Moraru, R. I., Băbuţ, G. B., & Cioca, L. I. (2014). Rationale and criteria development for risk assessment tool selection in work environments. *Environmental Engineering and Management Journal, 13*(6), 1371–1376.
20. Haghighat, A. T., Faez, K., Mowlaei, A. Ghahremani, Y., & Dehghan, M. (2002). Efficient multicast routing with multiple QOS constraints based on genetic algorithms. In Proceedings of SoftCOM2002, Croatia (pp. 626–630), Nov 8–11, 2002.
21. Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
22. Lee, C. Y. (1961). An algorithm for path connections and its applications. In *IRE Transaction on Electronic Computers* (pp. 345–365), Sept 1961.
23. Hochberger, C., & Hoffmann, R. (1996). *Solving routing problems with cellular automata*. In Proceedings of the second conference on cellular automata for research and industry, Milan, Italy.
24. Ghalavand, A., Khademzadeh, A., Dana, A., & Ghalavand, G. (2011). A routing algorithm based on cellular automata for mobile ad hoc networks. *IJCSI*, *8*(5).
25. Barolli, L., Koyama, A., Suganuma, T., & Shiratori, N. (2003). GAMAN: A GA based QoS routing method for mobile ad hoc networks. *Journal of Interconnection Networks (JOIN), 4*(3), 251–270.
26. Wu, K., & Harms, J. (2001). QoS support in mobile ad hoc net-works. *Crossing Boundaries - An Interdisciplinary Journal, 1*(1), 92–106.
27. Nancharaiaha, B., & Mohan, B. C. (2014). The performance of a hybrid routing intelligent algorithm in a mobile ad hoc network. *Elsevier, Computers & Electrical Engineering*, *40*(4), 1255–1264.
28. Abid, S. A., Othman, M., Shah, N., Ali, M., & Khan, A. R. (2014). 3D-RP: A DHT-based routing protocol for MANETs. *The Computer Journal*, *58*(2), 258–279.
29. Suna, B., Pi, S., Gui, C., Zeng, Y., Yan, B., Wang, W., & Qin, Q. (2008). Multiple constraints QoS multicast routing optimization algorithm in MANET based on GA. *Elsevier, Progress in Natural Science*, *18*(3), 331–336.
30. Huanga, J., & Liu, Y. (2010) MOEAQ: A QoS-aware multicast routing algorithm for MANET. *Elsevier, Expert Systems with Applications, 37*(2), 1391–1399.

31. Yena, Y.-Sh., Chao, H.-Ch., Changd, R.-Sh., & Vasilakos, A. (2011). Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs. *Elsevier, Mathematical and Computer Modelling*, *53*(11–12), 2238–2250.
32. Ramasubramanian, V., Haas, Z. J., & Sirer, E. G. (2003). *SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks.* In proceedings of the 4th ACM international symposium on mobile ad hoc networking and computing (pp. 303–314).
33. Cunha, R. O., Silva, A. P., Loreiro, A. A. F., & Ruiz, L. B. (2005). *Simulating large wireless sensor networks using cellular automata.* In Proceedings of 38th annual simulation symposium (pp. 323–330).
34. Lee, S. J., Royer, E. M., & Perkins, C. E. (2003). Scalability study of the ad hoc on-demand distance vector routing protocol. *ACM/Wiley International Journal of Network Management, 13*(2), 97–114.

Mitra Ahmadi received her Bsc in Computer Engineering-Software major in Iran University Science and Technology, Tehran, Iran in 2006 and her Msc in computer software engineering in Department of Computer. Eng, Islamic Azad University, South Tehran Branch, Tehran, Iran in 2012. Respectively. Her interests contains Artificial intelligance, Sensor network, channel assignment and routing in network. She published several internal papers in her fields. Currently, she is a programmer and data analyzer in private company in Tehran since Now.

Mohammad Shojafar is currently a 3rd year Ph.D. student in Information and Communication Engineering at DIET Dept. of the "La Sapienza" University of Rome. He Received his Msc in Software Engineering in Qazvin Islamic Azad University, Qazvin, Iran in 2010. Also, he Received his B.sc in Computer Engineering-Software major in Iran University Science and Technology, Tehran, Iran in 2006. His current research focuses on wireless communications, distributed computing and mathematical and AI optimization. He is an author/co-author of 40+ peer-reviewed publications (h-index = 9) in well-known conferences (e.g., IEEE ICC, GLOBECOM, PIMRC, HIS, ISDA) and journals in IEEE, Elsevier, IOS press and Springer publishers. Since 2013, he is the membership of IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing and a Distinguished Lecturer of IEEE Computer Society representing Europe. In addition, Mohammad was a Programmer and Analyzer in Exploration Directorate Section at N.I.O.C in Iran from 2012 to 2013. Email: shojafar@diet.uniroma1.it and follow more information at: www.mshojafar.com.

Ahmad Khademzadeh was born in Mashhad, Iran, in 1943. He received the B.Sc. degree in applied physics from Ferdowsi University, Mashhad, Iran, in 1969 and the M.Sc., Ph.D. degrees respectively in Digital Communication and Information Theory and Error Control Coding from the University of Kent, Canterbury, U. K. He is currently the Head of Education and National Scientific and Informational Scientific Cooperation Department at Iran Telecom Research Center (ITRC).He was the head of Test Engineering Group and the director of Computer and Communication Department at ITRC. He is also a lecturer at Tehran Universities and he is a committee member of Iranian Computer society and also a committee member of the Iranian Electrical Engineering Conference Permanent Committee. Dr. Khadem Zadeh has been received four distinguished national and international awards including Kharazmi International Award, and has been selected as the National outstanding researcher of the Iran Ministry of Information and Communication Technology. Mr.Khadem Zadeh could be reached at zadeh@itrc.ac.ir.

Kambiz Badie in currently Knowledge Management and e-Organizations Group, IT Research Faculty, Research Institute for ICT, Tehran, Iran. He received all his degrees from Tokyo Institute of Technology, Japan, majoring in pattern recognition. His interest for pattern recognition lies mostly in his motivation for grasping the mechanisms behind cognitive processes such as perception, intuition, imagination and interpretation. Dr. Badie is one of the active researchers in the areas of interdisciplinary and transdisciplinary studies in Iran, and has a high motivation for applying intelligent/cognitive modeling methodology to the human issues.

Reza Tavoli is Ph.D Student in computer Science in Faculty of Electrical Engineering, Computer and Information Technology at Qazvin Islamic Azad University from Feb 2011. Reza Received his B.sc in Computer Engineering-Software major in Iran University Science and Technology, Tehran, Iran in 2006 and his master of science in software in science and research Islamic Azad University, Tehran in 2009. His interests include applied Artificial intelligance in image retrevial and data mining. He published more than 10 conference and journal papers in his interest. Email: r.tavoli@gmail.com.