Journal of Computer Science

Original Research Paper

An Enhanced Dynamic Source Routing Algorithm for the Mobile Ad-Hoc Network using Reinforcement learning under the COVID-19 Conditions

Ehsan Moqimi, Amir Najafi and Mojtaba Ajami

Department of Computer Engineering, Zanjan Branch Islamic Azad University, Zanjan Branch, Iran

Article history
Received: 03-07-2020
Revised: 16-10-2020
Accepted: 31-10-2020

Corresponding Author:
Ehsan Moqimi
Department of Computer Engineering, Zanjan Branch Islamic Azad University, Zanjan Branch, Iran

Abstract: Today, the use of computer networks is evident in all walks of life, including mobile phones, vehicles, personal computers, the workplace and more. Creating and operating networks between each of the above-mentioned devices require good infrastructure to be able to communicate between these devices. To solve this problem, there were networks called ad-hoc networks, which were able to network with each other without the need for specific infrastructure and only through direct and wireless communication between the equipment. But how to make the right connections in these types of networks, as well as high-speed communication for devices, are also challenges for researchers. In this study, we try to implement the communication between these devices by the DSR routing algorithm and, using the Q learning technique of the reinforced learning algorithm, we enhance it to respond to QOS in Ad-Hoc mobile networks compared to previous solutions. In addition to what has been improved in this article. In the following articles, we will try to improve the discussion of communication security using this method and also apply this method to routing VANETs networks.

Keywords: Mobile Ad-Hoc Networks, DSR, Reinforcement Learning, Q learning, Routing Algorithm, COVID-19

Introduction

Today, wireless communication is one of the most important human communication needs. Therefore, the number of devices that have wireless receivers and transmitters is increasing day by day, which requires proper infrastructure to communicate with each other.

As a result, Mobile Ad-Hoc Networks (MANETs) have been developed to provide secure, wireless communication. These types of networks are a set of nodes and each node can communicate wirelessly, exchanging information with other nodes. Devices exchange information on these networks without the need for any infrastructure.

Given the prevalence of COVID-19 disease, which is one of the most challenging human problems to date, ad-hoc mobile networks can be of great help in controlling and preventing the spread of the disease. As we can see in the news and media, the countries involved in this virus have been forced to build morning hospitals due to the rapid spread of the disease and its progressive growth, more than the medical capacities of those countries.

Therefore, the speed of construction of these centers can be very important. Therefore, ad-hoc networks, one of the most important features of which is not the need for pre-prepared infrastructure, can be very effective in quickly equipping these centers. On the other hand, during quarantine and not leaving people's homes, stores can be set up quickly with the help of Unmanned Aerial Vehicle (UAVs) that are connected to each other using these networks to send basic necessities to people, especially the elderly and the sick.

Considering Hard Time Windows (HTW) constraint in delivering the product to customer using homogenous vehicles, A new approach to the use of Unmanned Aerial Vehicle (UAVs) communicated through MANET has been proposed, where the first objective function to be minimized represents the total costs of the network (including pollution costs depending on vehicles features, speed-related costs in heavy traffic and free speed, speed related costs in the light traffic condition, delivery drivers' wage and travelling costs of the route in the pick-up section). The second objective function is to

maximize the supply reliability which leads to customers' satisfaction maximization (Tirkolaee *et al*., 2020).

We need to focus on the optimal strategic and operational decisions to manage and maintain their logistic systems efficiently. As far as the optimal routes can cause cost reduction and improvement of service quality (Tirkolaee *et al*., 2019).

MANET stands for Mobile ad-hoc Network also called as wireless ad-hoc network or ad-hoc wireless network that usually has a routable networking environment on top of a link layer ad-hoc network. They consist of set of mobile nodes connected wirelessly in a self-configured, self-healing network without having a fixed infrastructure. MANET nodes are free to move randomly as the network topology changes frequently. Each node behaves as a router as they forward traffic to other specified node in the network (AnushkaKhattri, 2019).

For example: Figure 1 shows a plain ad-hoc network with 3 nodes. Node 1 and node 3 are not within range of each other, however the node 2 can be used to forward packets between node 1 and node 2. The node 2 will act as a router and these three nodes together form an ad-hoc network (Chouksey, 2016).

MANETs have unique characteristics that differentiate them from wired networks, namely their highly dynamic topology, wireless links and limited resources (Yadav and Hussain, 2017). One of the most important features making it complicated and expensive to improve the performance of this type of network is the mobility of nodes. The nodes that create the routes to link a source node to a destination node are free to move in random directions at different speeds, which mean that the network may experience unpredictable and rapid changes in topology. To deal with this, routes need to be found and constantly maintained, but this is not enough to guarantee the flow of data from the source to the destination because the nodes in ad-hoc networks generally have limited transmission range (Yadav and Hussain, 2017). Thus, even if the routing protocols are working efficiently in finding and maintaining routes, the network performance will be reduced if the nodes keep moving away from the transmission range.

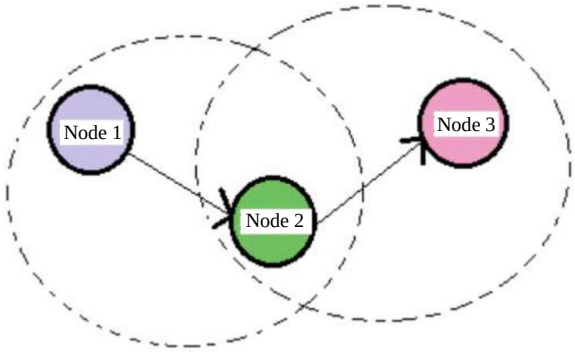

Fig. 1: Example of mobile ad-hoc network (Chouksey, 2016)

For this reason, protocols have been developed to better router and increase the efficiency of ad-hoc mobile networks. In general, ad-hoc network routing methods are divided into two types of protocols, one type of which is reactive routing protocols and the other type of which is called proactive routing protocols. Each of these categories has a subset of protocols. The DSR routing protocol, which is one of the reactive routing protocols, is one of the protocols that has attracted the attention of researchers due to its high efficiency and good efficiency in scenarios of ad-hoc mobile networks that have a lot of movement.

DSR are among the best known and most common reactive MANET routing protocols. Each node in DSR discovers a path to the destination dynamically. Every packet going through the network knows the path it has to follow to reach the destination, a technique called source routing (Yadav and Hussain, 2017). The information about the route is stored in the header of the packet sent. DSR comprises two mechanisms, one for route discovery and the other for route maintenance (Ali Alzahrani, 2019; Sarkar, 2016).

Therefore, to enhance the efficiency of DSR routing in MANET networks, researchers use some machine learning algorithms to try to respond to QOS components. Chatterjee and Das (2015) have tried to improve DSR routing algorithms with Ant Colony algorithm (AC).

The idea behind applying AC in DSR protocol is to discover and maintain the best routes among the nodes. Although AC requires limited computation and power from the individual nodes, it can still provide effective routing. Due to the self-organizing and adapting capabilities of the artificial ants AC can keep the routing tables efficiently updated (Chatterjee and Das, 2015).

After optimizing the DSR using AC by Shubhajeet Chatterjee. There were still no improvements in some QOS parameters. That's why in the design that we presented, we improved the parameters that we will refer to in the following chapters compared to the design presented by Shubhajeet Chatterjee using the reinforcement learning algorithm.

In reinforcement learning, the learner is a decision-making agent that takes actions in an environment and receives reward (or penalty) for all its actions in trying to solve a problem. After a set of trial-and-error runs, it should learn the best policy, which is the sequence of actions that maximize the total reward (Alpaydin, 2020).

Reinforcement learning is different from the learning methods we discussed before in a number of respects. It is call "learning with critic" as opposed to learning with a teacher which we have in supervised learning. A critic differs from a teacher in that it does not tell us what to do but only how well we have been doing in the past; the critic never informs in advance. The feedback from the critic is scarce and when it

comes, it comes late. This leads to credit assignment problem. After taking several actions and getting the reward, we would like to assess the individual actions we did in the past and find the moves that led us to win the reward so that we can record and recall them alter on.as we will see shortly, what a reinforcement learning program does is learn to generate an internal value for the intermediate states or actions in terms of how good they are in leading us to the goal and getting us to the real reward. Once such an internal reward mechanism is learned, the agent can just take the local action maximize it (Alpaydin, 2020).

Related Works

In this section we summarize the studies which are closely related to this research: One is Ant colony optimization based enhanced dynamic source routing algorithm for mobile Ad-hoc network, The second is an article entitled is Dynamic Routing Algorithm with Q-learning for Internet of things with Delayed Estimator, the third is another article entitled Reengineering MANET routing using Ant Colony Optimization: Modelling and Performance Study، the fourth is an article entitled energy efficient multipath routing using multi-objective grey wolf optimizer based dynamic source routing algorithm for MANET and the other is dynamic routing algorithm with q-learning for internet of things with delayed estimator.

a) In this scheme enhanced version of the well-known Dynamic Source Routing (DSR) scheme based on the Ant Colony Optimization (ACO) algorithm, which can produce a high data packet delivery ratio in low end to end delay with low routing overhead and low energy consumption (Boukerche *et al*., 2011). In that, when a node needs to send a packet to another node, like DSR, it first checks the cache for existing routes. When no routes are known, the sender node locally broadcasts the Route Request control packets to find out the routes. This is similar to the biological ants initially spreading out in all directions from their colony in search of food. Now, the ants, after finding the food source, come back to the colony and deposit pheromone on their way so that other ants get informed about the paths (Dorigo and Blum, 2005). Under the ant colony framework, the best route is selected by the pheromone level of the route (Chatterjee and Das, 2015). In this study, however, efforts have been made to improve criteria such as data packet delivery ratios, end-to-end delays, low-routing overflows and energy consumption. But it has not been able to improve enough
b) In this article, we will look at a plan for routing data transfer to the Internet Of Things (IOT). On the Internet of Things information exchange and communication require novel intelligent routing algorithms as traditional routing algorithms are unfit for current network environment (Lee *et al*., 2013). Q-routing implemented a dynamic adjustment which was based on the network environment by combining the Q-learning algorithm. However, Q-routing is a highly random network environment and leads to a decline in performance because of overestimation of values. To solve the problem, we propose an algorithm called Delayed Q-routing (DQ-routing), which uses two sets of value functions to carry out random delayed updates so as to reduce the overestimation of the value function and improve the rate of convergence (Shilova *et al*., 2016). The experiments indicate that Dqrouting algorithm gets well performance in several problems. We used the idea of this article to improve navigation using the reinforcement learning algorithm. Of course, the method of using the reinforcement learning algorithm in our article is completely innovative and only the idea of this article has been used (Wang *et al*., 2019)
c) This paper seek to compare preexisting and proposed routing algorithm for MANET based on the mechanism of the ant system, hence Ant Colony Optimization frame would be adopted. It is notable that MANET bandwidth, radio propagation, energy supply, etc. Different MAC protocols have proposed for ad-hoc networks (Anibrika *et al*., 2020)
d) In this study, Multi objective Grey Wolf Optimization (MGWO) based DSR protocol is proposed which utilizes energy, delay, lifetime and link quality as objective parameters. Initially, the routes are discovered based on the DSR strategy and the objective parameters are estimated using which the MGWO models an objective or fitness function. Fitness values are computed for each available path and are sorted in the best order as in wolves' hierarchy of MGWO. Finally, multi-path data transmission will be performed over the high fitness paths (alpha, beta and delta) while the next best omega paths will be utilized when the energy of dominant paths depletes below the level of omega paths (Ghaleb and Vasanthi, 2020)

Research Methodology

Due to the drastic natural changes that are necessary for ad-hoc networks, there is no the comprehensive and centralized solution that could provide the best path based on various parameters such as route length, route congestion and so on. Also, no node in ad-hoc networks alone can perform central management tasks, due to energy limitations and processing capacity limitations. As a result, routing in Ad-hoc Mobile Networks

(MANET) is one of the most challenging topics for researchers (Chatterjee and Das, 2015). The low rate of packet sending due to the successive rupture of communication lines in high-traffic scenarios, in previous models led to increased response delay depending on End to End, increased routing overhead due to poor route detection schemes and increased consumption it becomes energy. Therefore, in this chapter, we will present a proposed model for improving routing in ad-hoc networks based on reinforcement learning algorithm. This model is a type of reactive routing called Dynamic Source Routing (DSR) Improves reinforcement learning using the algorithm. This model has all the features of a comprehensive model.

As we know, in routing using the dynamic source routing algorithm, to find a new path and connect between nodes in ad-hoc mobile networks, we need to create request and replay routing packages, which in our proposed model called RRQA and RRPA Are named. The design of the structure and its components are explained in the next section. One of those packages, which is the path request packet, is sent to the nodes near that node through the source node that needs to find the path and so on until it reaches the destination node. The other node, which is the Reply node of the path, returns the found path to the source node after finding the path and reaching the destination by the destination node.

RRQA and RRPA Packet Sending Format in the Proposed Model

In our proposed model, we call these packets Agent Request Route (RRQA) and Route Response Agent (RRPA), the format of which has changed from the usual route request packages.

In Fig. 2, the code of the dedicated factory address (MAC) is more important than the Internet Protocol (IP) address, because each node requires network infrastructure to assign an IP address:

- Number of steps: A field in which the number of steps to the destination is displayed
- Initial Rewards: The amount of rewards available for the destination
- Q Value for source to destination: The initial value of Q that is updated for each route
- Type: Closed type. Specifies RRQA = 1, RRPA = 0. Intermediate node address stack: MAC address stores intermediate nodes as a stack

Measurement of Congestion Level

Each node calculates the amount of congestion that depends on the occupation of the node buffer, the size of the load channel and the rate of loss of packets. This is a numerical value between zero and one. (1- high congestion, 0- low congestion) because this value is a linear size that shows a completely similar effect based on high congestion (0.7 or 0.8) and also low congestion (0.5 or 0.4). Therefore, to provide an appropriate effect based on node congestion, the value of the new node congestion is calculated nonlinearly from the following equation (Chatterjee and Das, 2015):

$$\begin{aligned} &Node\ nonlinear\ congestion \\ &= \left(1-\left(1-Node\ linear\ congestion\right)\right)^2 \end{aligned} \tag{1}$$

Example:

$$Linear\ congestion = 0/2\ Nonlinear\ congestion$$
$$= \left(1-\left(1-0/2\right)^2 = 0/86\right.$$

The above formula will be more effective than the linear type in conditions with high congestion. However, it should be noted that delays and energy consumption will increase under conditions with a high degree of congestion (Chatterjee and Das, 2015).

The density of each node can play an important role in making a better decision about choosing the optimal path, which is calculated and updated by the node itself. A method has been proposed in which the skepticism predicts the path to some extent before it occurs. In it, each node must identify its available neighbors at specified intervals. This will be fully explained in the next section.

Source address (48 bits)			
Destination address (48 bits)			
number of steps (7 bits)	initial source-to-destination reward	Q value from source to destination	**Packet type**
middle node address stack			

Fig. 2: The format of packet detection packets in the proposed method

Measuring the Communication level of a Route

Due to the random movement of nodes in ad-hoc Mobile Networks (MANETs), disconnection of lines in one direction is a very common occurrence. Now we want, according to the following plan, to disconnect the lines before it happens. To predict the intersection of lines, we have a value of Received Signal Strength Metric (RSSM) for each line (Chatterjee and Das, 2015). Now, during the connection setup (before any data packet needs to be sent), the antennas of each node publish a NASM packet that is in 3-2 format.

When a closed node receives NASM from other nodes, it stores the source address, antenna gain, maximum transmission power and maximum line speed limit for calculating RSSM. It also stores NNCM, which is the level of line congestion. Antenna gain, maximum transmission power and 10-bit speed limit are considered, because we were not able to determine the maximum appropriate size of this field beforehand, so we have considered an average bit value. Or we will discuss the correct one (0 or 1). So like the CM field, we have reserved 20 bits for the NNCM field.

We know that the signal strength received from the neighboring node *i* can be expressed from a distance X. (RRSIX) λ is the wavelength used in occasional Mobile Networks (MANETS):

$$RRS_{ix} = \frac{G_r \times G_t \times S_t}{\left(4\pi \times \frac{x}{\lambda}\right)^2} \tag{2}$$

G_r antenna receiving rate, G_t antenna transfer rate, S_t Maximum transmission power from antenna transmission, we also have a value of T_i, which is the threshold of the signal strength received from the neighbors, which is as follows:

$$T_i = \frac{G_r \times G_t \times S_t}{\left(4\pi \times \frac{0.905R}{\lambda}\right)^2} \tag{3}$$

The receiving node knows what its antenna rate is (G_r), also knows its coverage Radius (R) and knows what the wavelength of the network is when the node propagates the antenna reception rate (G_t) and also completes the maximum transmission power (S_t) of the neighboring antenna. So the value of the neighbor's threshold can be easily calculated with a simple delay calculation (Chatterjee and Das, 2015).

RSSM calculation for each of the links (Received Signal Strength Matric) the nodes depend on the value (t_j) (communication with each of the neighboring nodes). The value of RSSM for line *I* (communication) with the neighbor node *i* to show:

$$RSSM_i = \begin{cases} 0 & if\ RSS_{ix} < T_i \\ 1 - \frac{T_i}{RSS_{ix}} & if\ RSS_{ix} \geq T_i \end{cases} \tag{4}$$

Lines with zero RSSM value are immediately removed from memory and lines with non-volatile RSSM are stored in memory. As you can see in the Fig. 3, a node has two caches, one of which is for storing lines with RSSM, NNCM, the value of *Q* and the value of *R*, or the reward for each of the lines and the second is the cache for paths. That is the path to the different nodes of the network. Due to network changes per unit time, the RSSM and NNCM values of the lines need to be periodically updated:

$$\alpha = \frac{0.0946}{V_{host} + V_{neighbor}} \tag{5}$$

In which *R* is the radius covered by node *V*, the maximum speed limit of nodes in the base node and the neighboring node is, which *R* and *V* considers the base node itself as each node and using NASM packages, the neighbor *V* is also obtained and the value of α It calculates easily. After α seconds, each node releases a NASM packet to update the RSSM and NNCM of its lines. After receiving the NASM response, it checks its memory node and replaces the previous RSSM and NNCM value with the new value if it finds the desired line. Slow and if there is no line in the memory, it stores it in memory and if the RSSM value of the lines is zero, it clears it from memory (Chatterjee and Das, 2015).

The only way to avoid the phenomenon of line failure is to anticipate it before it occurs. To predict line failure, we create some threshold numbers (T_i) that are the strength of the signals received from the lines if a node receives the NASM response packet from its neighboring lines with the RSSIX signal strength and this value is greater than or equal to the threshold number the line is secure. For example, in dynamic ad-hoc MANET mobile network environments, lines are maintained Several times (Sayt).

They set the RSSM value of the line to $1 - \frac{T_i}{RSS_{ix}}$. If the signal strength is less than the threshold value, then that line breaks. Now, this time interval (α) must be chosen so that $t \geq \alpha$, for example $t_{minimum} = \alpha$, otherwise, for example, if ($t < \alpha$) it is possible to break the reliability of the lines in time interval α and the desired node is broken. The lines are neglected at this distance and try to use this link to send data.

As mentioned, the T_i value of the threshold is obtained from the signal strength of a part of the neighboring nodes, which is from the signal strength

received from the node antenna, which is at a distance of $R * X$ (X is between zero and one) from this node. So if the neighboring node is exactly at the distance $X * R$ from the base node and both of them start to move away from each other in exactly the opposite way with the maximum V speed of the host and the neighboring V, respectively. We can say that when you can keep your communication line active for more than $\frac{(1-x)R}{V_{host}+V_{neighbor}}$ time, the minimum value of T ($T_{minimum}$), is $\frac{(1-x)R}{V_{host}+V_{neighbor}}$. Therefore, we have considered the value of α, within this time interval α node can trust all the neighbors that have RSSM above zero. Releases NASM package.

The space covered by the R-range antennas can be considered a circular space with a radius of R. We know that the average distance between two random mobile nodes in a circular space with a radius of R is 0.9054 R. The average distance from randomly receiving nodes from a transmitting node in the antenna range to radius R is 0.9054 R, here the threshold value is used to predict the phenomenon of line failure, before it occurs, at a greater distance. So the value of the signal strength threshold is generated by the signal strength received from the antenna of a node located at a distance of 0.9054 R from the base node (Fig. 4) and the time interval α corresponds to the time $\frac{(1-x)R}{V_{host}+V_{neighbor}}$ (Chatterjee and Das, 2015).

source 47 bits	NNCM 20 bit	Antenna gain ۱۰bit	Maximum Transmitting Power ۱۰bit	Speed limit ۱۰bit

Fig. 3: NASM package format

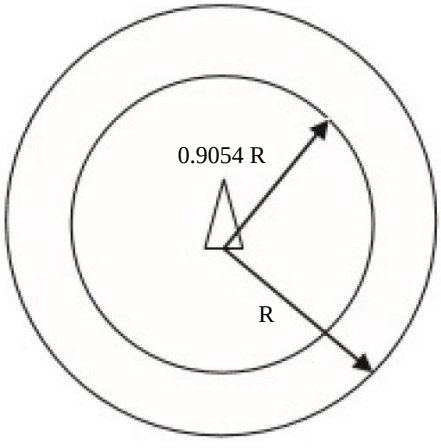

Fig. 4: R range and antenna threshold of each node

In previous designs, packages were published periodically (in *t*-hello seconds) by nodes and the destination node stored the sender's node in memory and deleted it from memory when a line was missing after a period of time (2 * *t*-hello). This technique had two important drawbacks, first of all, the time interval (*t*-hello) was chosen arbitrarily. When a node receives a message from a neighboring node and then due to a random movement of the node, the neighboring node leaves the area covered and the node is unaware of this event and continues the broken path in its memory until the end of time *x*. If the broken line is used to transfer information during this time interval, it will fail. On the other hand, storing this broken line is unnecessary to transfer memory. In our design, we select the repetition time in a way that is based on the random movement of the nodes. In this scheme, there is no chance of failure for high reliability lines at α time (those whose RSSM is greater than zero) At the end of α time, the RSSMs will be updated again and in addition to the effective use of CACHE memory, the lines whose RSSM is zero will be erased from memory (Chatterjee and Das, 2015).

Also, in the middle lines, no measurements are made for reliability. Due to the random movement of the nodes, no assurance is given for the reliability characteristics of the intermediate nodes. In our design, because the RRQA packets, when using the information obtained from the Neighbor Assessment (NASM) packet, it pays to discover the path when needed, which at any time due to the return of the operating packet to Frequent access to all available nodes, during routing, can also evaluate routes whose RSSM lines are zero and calculate them in the *Q* matrix. Naturally, it will have a lower final *Q* value than other paths and as a result, due to the process of learning the agent, it will be deleted in the process of discovering the path and will not be stored in CACHE.

The future of nodes is randomly predicted based on the current position and velocity using the Global Geographic Positioning System (GPS) of the node path and the reliability prediction of the lines based on it. But the MANET case scenario may not be able to support the Global Geographic Location System (GPS) because one of the notable features of the MANET mobile network is the ability to communicate without expanding existing infrastructure. Accidental processing may lead to errors. To a large extent, in high-mobility scenarios. In our design, we will not encounter a problem that requires random prediction with the Global Geographic Positioning System (GPS) to gain gradual confidence in the lines (Chatterjee and Das, 2015).

To use the reinforcement learning algorithm, we must assign rewards using the calculated parameters for each operation performed (here the path discovery). In fact, according to the theory of reinforcement learning, each node needs a reward to evaluate the action in order to

find the most optimal path among the discovered paths. The next section discusses how to do this in this article.

Calculating the Amount of Reward for Each Line of Communication with the Neighboring Node

In our proposed scheme, to calculate the amount of reward in each node separately, after finding the answer from the neighbors' assessment packages in the time interval α, we calculate using RSSM and NNCM values and save them in the line reward table. Each communication line has its own reward in each node.

As mentioned in the previous sections, the RSSM and NNCM values are between zero and one, but with the difference that the RSSM value is closer to one indicates more confidence than failure in lines and approaching zero indicates zero. Less reliability than line failure. But the closer the NNCM is to one, the higher the density of the next node and the closer it is to zero, the lower the density in the next node. So because we need to maximize rewards in reinforcing learning, we need to turn the NNCM component into a positive incremental process instead of a positive one using the following 6 formula:

$$NNCM' = 1 - NNCM \tag{6}$$

And by multiplying these two like formula 7, we calculate the amount of *RW* reward for each of the lines:

$$RW = \left(NNCM' * RSSM \right) * 100 \tag{7}$$

The value of RW is also a number between zero and one, which when approaching one indicates a low density of the next node and high confidence in failure of lines and closer to zero indicates high density and uncertainty of failure of high lines. Now, to maximize the value of the RW bonus and make it more noticeable, we multiply it by 100.

Calculate the Value of Q

Depending on how the reinforcement learning algorithm works, agents need algorithms to learn. Also, in matters where the current situation depends on the previous situation, there is the property of Markov. One of the algorithms used for these issues, which we have used in this article, is the *Q* learning algorithm, which is based on reinforcement learning. The agent also increases its *Q* value after several stages of learning. It will show us the best possible path.

After all these explanations, we will go into the details of the process of calculating the value of *Q*. At first, the value of *Q* is considered zero in all nodes and for all communication lines. After sending the RRQA package to each node through Calculated and stored in the *Q* table in Cache (CACHE).

$$Q\left(state, action \right) = RW\left(state, action \right) + \delta * Max\left[Q\left(nexstate, allaction \right) \right] \tag{8}$$

In formula 8, *Q* (state, action) means to calculate the value of *Q* for the current state or node in which we are now and the node we are working to reach and in the RRQA packet. For example, in *Q* (2,1) we mean that we are now in node 2 and we intend to go to node one and we have calculated the value of *Q* for the path of node 1 in node 2. We mean the value of RW (state, action) of the same reward calculated for the node 1 in node 2. The value of δ is a value between zero and one, which is the coefficient for adjusting the method of checking bonuses. When it is closer to zero, it means that the factor tends to examine immediate rewards and when it is closer to one, it means that it tends to examine future rewards. Based on our experience, we have selected a value of 0.8 in the simulations and the meaning of Max[Q(nexstate, allaction) in formula 8 is that the maximum value of *Q* in the next step is ahead of the possible states that in RRPA packages this value is sent to the source node after calculation in the destination node. So we have all the components for calculating the value of *Q*. After calculating the value of *Q*, the new value replaces the previous value and this process ends after the routing is completed and again all the values of *Q* return to zero.

How to Navigate

Like DSR, when the source node starts sending a packet to the destination node, it first searches for its cache to find the appropriate path. If there is a route, it will send. However, if it does not find a path, it will start finding a new path by sending the RRQA packet randomly (Tokic, 2010) among the nodes that have announced their readiness and simultaneously calculating the Q value of the path taken, as well as this process in the next node until it reaches It continues to the destination, then stores the detected path in the RRPA package and sends it to the source. This process detects a certain number of paths for all required paths and from among them, the path that has the highest reward as the path. Optimally selects and saves α time in memory and uses it to send data. The maintenance process will be the same as the DSR algorithm.

Interpretation and Conclusion

Evaluation of the Proposed Model

Based on the implementation method discussed in the previous sections, we started simulating to evaluate our proposed model in the MATLAB analytical software environment with the defaults that will be mentioned. After conducting studies, we concluded that our proposed

model has a lower routing overhead than the optimization model using ant colonies in some routes and therefore has a higher performance in routing some routes than the optimal model. It is made using the colony of ants. Also, in our model, after simulation, the delays are low due to finding short and effective routes in all routes. This also indicates the high performance of this model compared to the optimized model using ant colonies.

Since energy consumption is very important in ad-hoc networks, in our simulation, we also evaluated the energy consumption of routing in each of the models (Fig. 5). Based on this, our proposed model for each route with energy consumption was much lower than the optimization model using ant colonies (Table 1). Based on this, in evaluating the performance of our proposed model according to the obtained results, we can conclude that our model has a better performance in improving the QOS components than the optimization model using ant colony.

Comparison of Model Diagrams

Problem Defaults

Results from Simulation

In all diagrams of steps 1 to 30, there are actually 30 different routes from each node to the other node. For example, the number 1 means the path from node one to node 2. The complete list of paths and their numbers are listed in the Table 2-13, which is the path of our nodes after simulation. N1-N5 in these tables are source node until destination node numb.

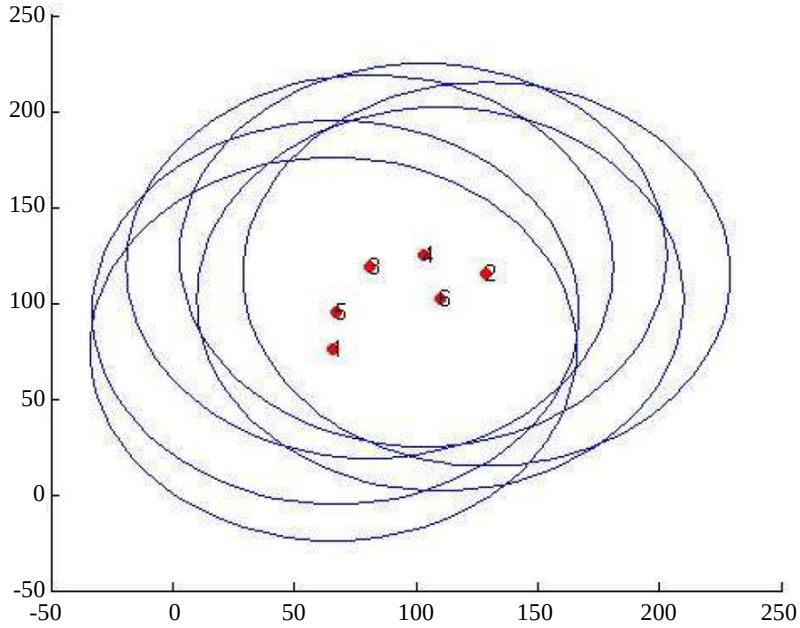

Fig. 5: How to establish nodes randomly in a simulation environment

Table 1: Simulation parameters

Row	Component name	Default value	Unit
1	Number of nodes	6	Number
2	Dimensions of the simulation	300*300	unit
3	Sizes of each package	64	Kbyte
4	Energies required for each processing	0,3	Jules
5	Number of steps to reinforcement learning simulation	20	times
6	Speeds of each node transfer	11264	Kbit/s
7	Antenna sent frequency	2,4	GHz

Due to the above tables, after routing using reinforcement learning method and the paths that were discovered with this method, QOS factors such as end to end delay, routing overhead and energy consumption were evaluated. The results of each of these assessments are presented in the form of a comparative chart between the reinforcement learning method and the previous method.

Table 2: Table of routes in node 1 by the ant colony method

Route	N1	N2	N3	N4	N5
1	1	5	3	4	2
2	1	5	3		
3	1	5	3	4	
4	1	5			
5	1	5	3	4	6

Table 3: Table of routes in node 1 with reinforcement learning method

Route	N1	N2	N3	N4
1	1	3	6	2
2	1	5	3	
3	1	3	4	
4	1	5		
5	1	3	6	

Table 4: Table of routes in node 2 with the ant colony method

Route	N1	N2	N3	N4	N5
1	2	4	3	5	1
2	2	6	4	3	
3	2	6	4		
4	2	4	3	5	
5	2	6			

Table 5: Table of routes in node 2 with reinforcement learning method

Route	N1	N2	N3	N4
1	2	6	5	1
2	2	4	3	
3	2	4		
4	2	6	3	5
5	2	6		

Table 6: Table of routes in node 3 with the ant colony method

Route	N1	N2	N3	N4
1		3	5	1
2	3	4	6	2
3	3	6	4	
4	3	5		
5	3	4	6	

Table 7: Table of routes in node 3 with reinforcement learning method

Route	N1	N2	N3
1	3	5	1
2	3	6	2
3	3	4	
4	3	5	
5	3	6	

Charts a, b and c of Fig. 6, the route tracking, end to end delay and energy consumption between Table 2, which are the routes discovered by the ant colony method and Table 3, which are the routes obtained from Comparative learning method has been compared.

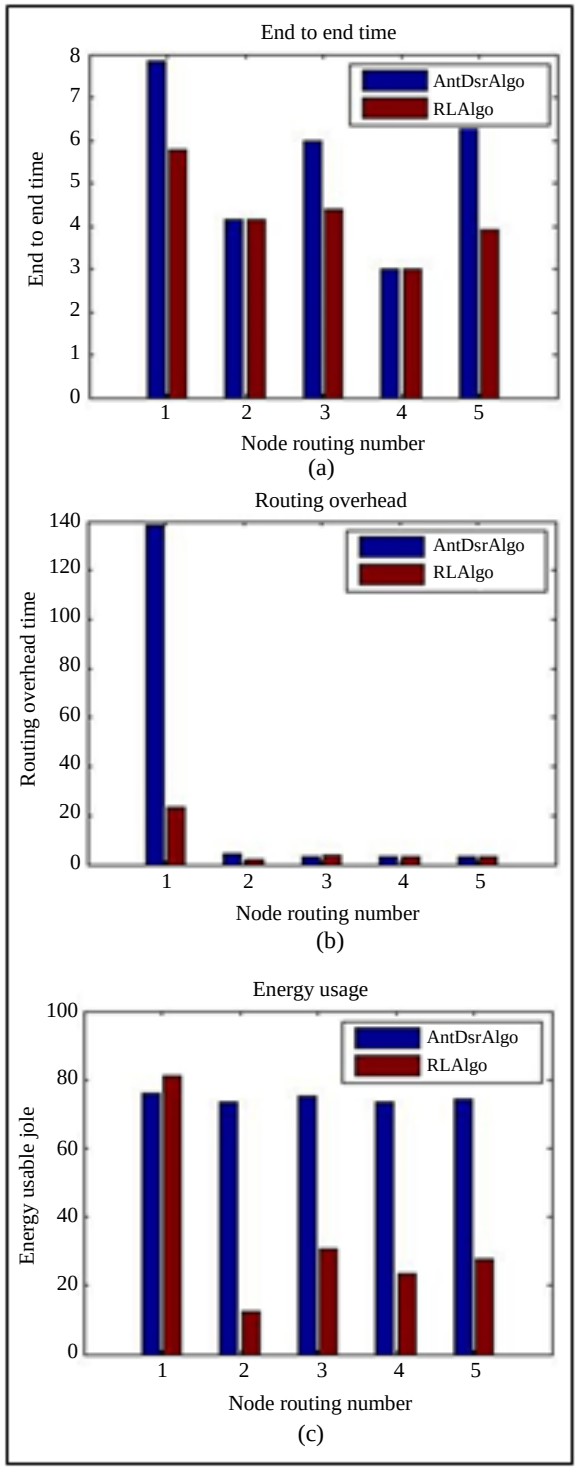

Fig. 6: Comparative evaluation charts between Table 2 and 3

Charts a, b and c of Figure 7, the route tracking, end to end delay and energy consumption between Table 4, which are the routes discovered by the ant colony method and Table 5, which are the routes obtained from Comparative learning method has been compared.

Charts d, e and f of Figure 7, the route tracking, end to end delay and energy consumption between Table 6, which are the routes discovered by the ant colony method and Table 7, which are the routes obtained from Comparative learning method has been compared.

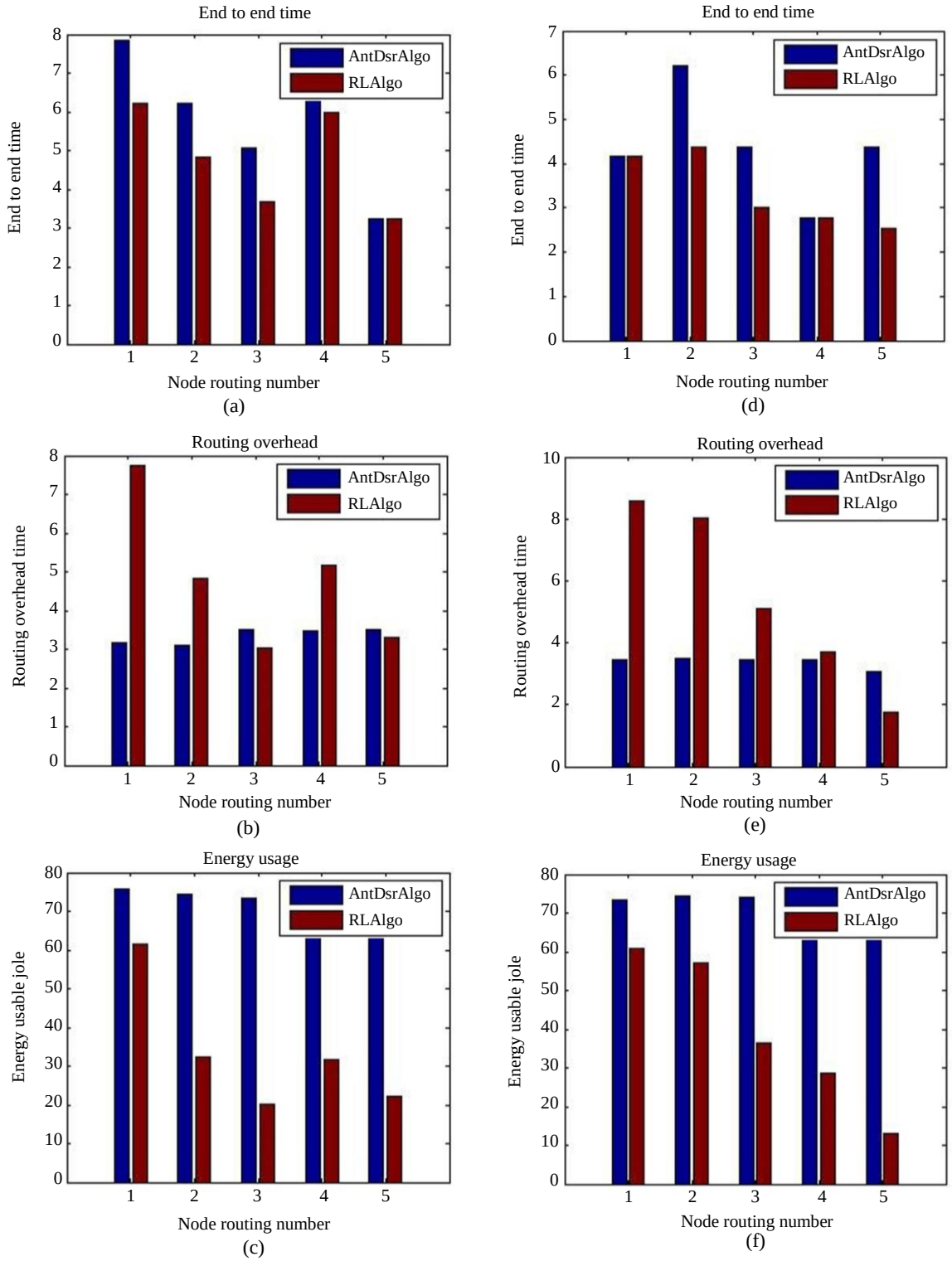

Fig. 7: Comparative evaluation charts between Tables 4 and 5, also Comparative evaluation charts between Tables 6 and 7

Charts a, b and c of Fig. 8, the route tracking, end to end delay and energy consumption between Table 8, which are the routes discovered by the ant colony method and Table 9, which are the routes obtained from Comparative learning method has been compared.

Charts d, e and f of Fig. 8, the route tracking, end to end delay and energy consumption between Table 10, which are the routes discovered by the ant colony method and Table 11, which are the routes obtained from Comparative learning method has been compared.

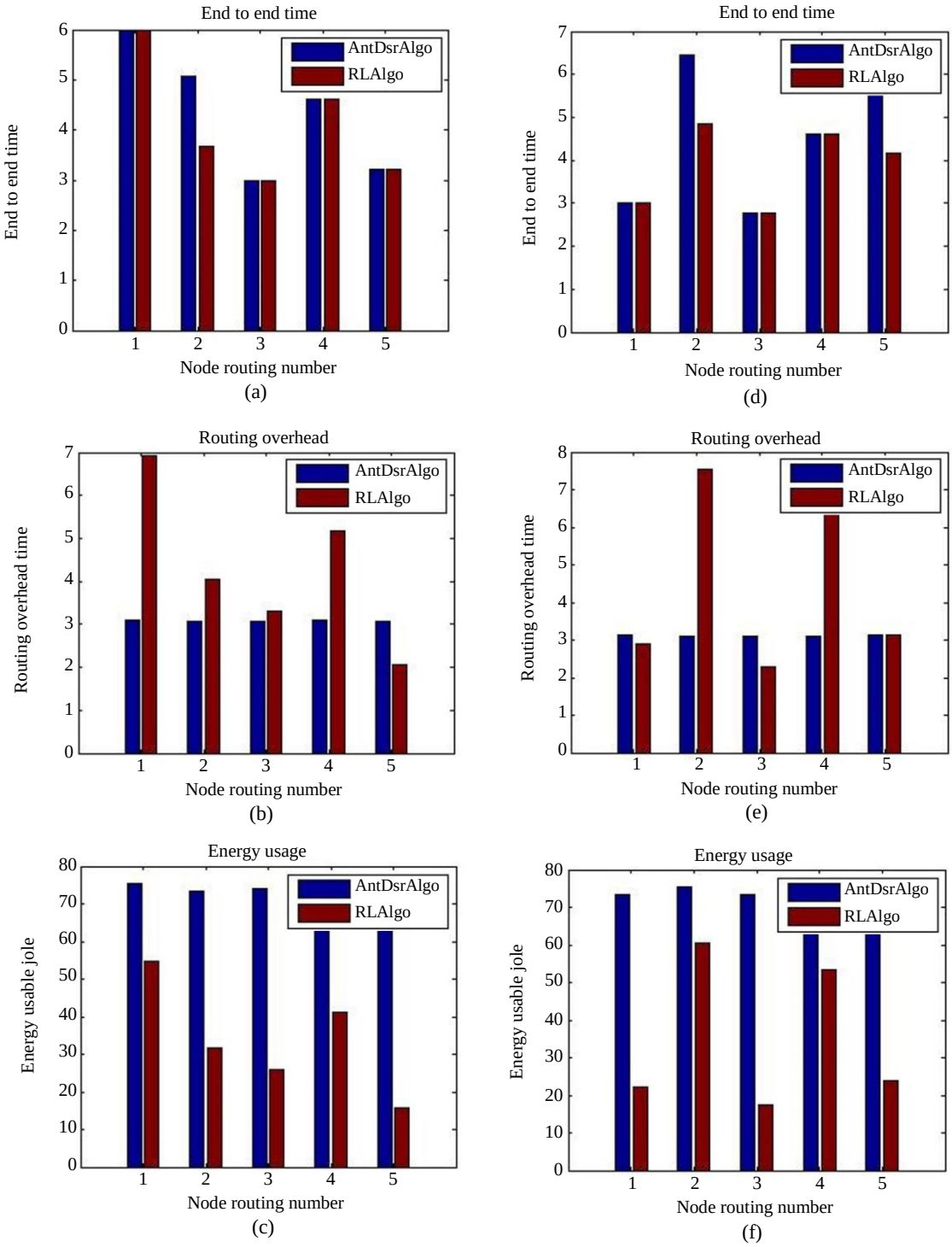

Fig. 8: Comparative evaluation charts between Tables 8 and 9, also comparative evaluation charts between Tables 10 and 11

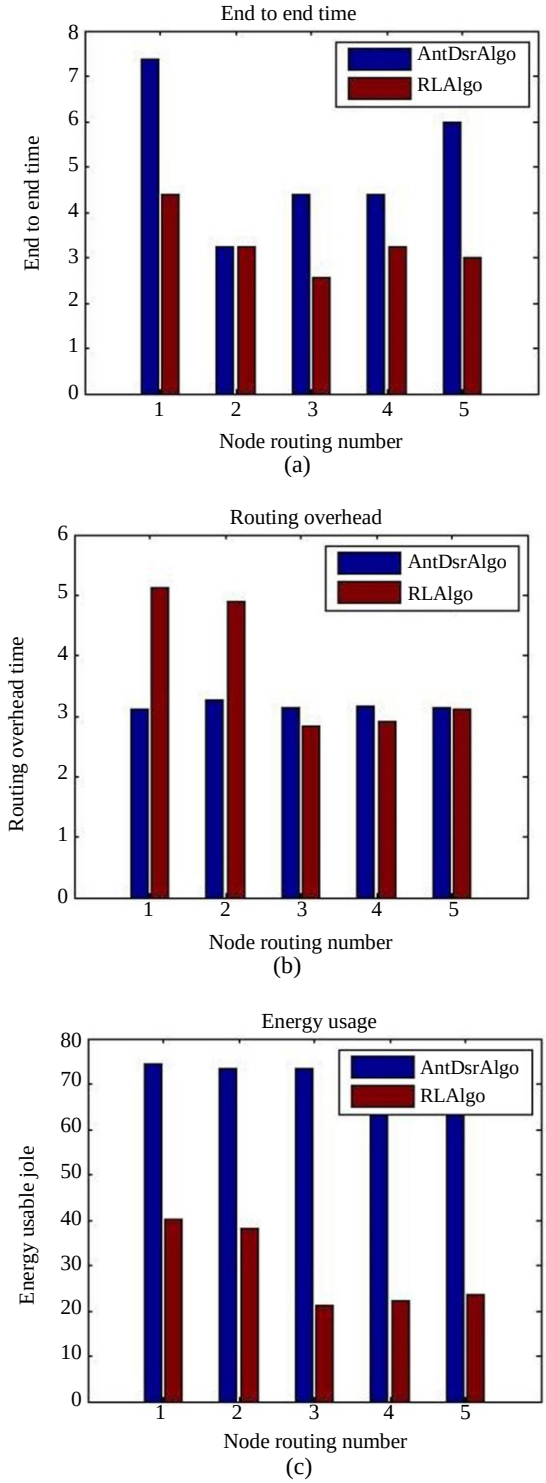

Fig. 9: Comparative evaluation charts between Tables 12 and 13

Charts a, b and c of Fig. 9, the route tracking, end to end delay and energy consumption between Table 12, which are the routes discovered by the ant colony method and Table 13, which are the routes obtained from Comparative learning method has been compared.

Table 8: Table of routes in node 4 by the ant colony method

Route	N1	N2	N3	N4
1	4	3	5	1
2	4	6	2	
3	4	3		
4	4	3	5	
5	4	6		

Table 9: Table of routes in node 4 with reinforcement learning method

Route	N1	N2	N3	N4
1	4	3	5	1
2	4	2		
3	4	3		
4	4	3	5	
5				

Table 10: Table of routes in node 5 by ant colony method

Route	N1	N2	N3	N4
1	5	1		
2	5	3	4	2
3	5	3		
4	5	3	4	
5	5	3	4	6

Table 11: Table of routes in node 5 with reinforcement learning method

Route	N1	N2	N3
1	5	1	
2	5	6	2
3	5	3	
4	5	3	4
5	5	3	6

Table 12: Table of routes in node 6 by ant colony method

Route	N1	N2	N3	N4	N5
1	6	4	3	5	1
2	6	2			
3	6	4	3		
4	6	3	4		
5	6	4	3	5	

Table 13: Table of routes in node 6 with reinforcement learning method

Route	N1	N2	N3
1	6	5	1
2	6	2	
3	6	3	
4	6	4	
5	6	5	

As you can see, using the reinforcement learning method, in addition to finding shorter paths, in 100% of the routes performed with end-to-end delays, it was less than the hair colony method, in 99.7% of cases with energy consumption. It is less than that method and in 50% of cases it has a low routing overhead compared to the reinforcement learning method.

Conclusion, Limitations and Implications

According to what was presented in this article, ad-hoc mobile networks are being developed day by day due to the need for infrastructure for development and launch and various applications of them can be seen. The results of the project presented in this study, which was able to prove performance improvement in Dynamic Source Routing protocol (DSR), compared to the ant colony-based model in the basic parameters of energy consumption, point-to point transmission speed and routing overhead, proved It can be a good way to implement this plan in a variety of ad-hoc networks. In order to prove the effectiveness of this plan in the future, it is planned to implement it in cases such as military communication networks as well as rescue teams to separate the networks from the usual mobile networks and do not need pre-designed infrastructure. Due to the fact that this model has advantages in energy consumption, point-to-point transmission speed and routing overhead compared to the optimized model based on ant colonies, this model can also be used in Vehicles Ad hoc Networks (VANET). Simulated and evaluated and considering that the issue of security in wireless networks is one of the main challenges we will do research on security in these networks and the proposed model in the future.

Acknowledgment

want to thank different people for helping with this project. Mr. Emami and Mr. Mohammadi, employees of my office, to help them collect data and create a simulation environment.

Author's Contributions

All authors equally contributed in this work.

Ethics

This statement is signed by all the authors to indicate agreement that the above information is true and correct.

References

Ali Alzahrani, H. J. (2019). Analysing the Effect of Mobility on the Performance of MANET Routing Protocols. UK Performance Engineering Workshop (p. 35th). UK: Performance Engineering.

Alpaydin, E. (2020). Introduction to Machine Learning. Massachusetts: Cambridge. MIT press.

AnushkaKhattri. (2019). Introduction of mobile ad-hoc network MANET. Geeks for Geeks: https://www.geeksfor_geeks.org/introduction-ofmobile-ad-hoc-network-manet/?ref=lbp

Anibrika, B. S. K., Asante, M., Hayfron-Acquah, B., & Gavua, E. K. (2020). Reengineering MANET Routing using Ant Colony Optimization: Modelling and Performance Study. International Journal of Computer Applications, 975, 8887.

Boukerche, A., Turgut, B., Aydin, N., Ahmad, M. Z., Bölöni, L., & Turgut, D. (2011). Routing protocols in ad hoc networks: A survey. Computer networks, 55(13), 3032-3080.

Chatterjee, S., & Das, S. (2015). Ant colony optimization based enhanced dynamic source routing algorithm for mobile Ad-hoc network. Information Sciences, 295, 67-90.

Chouksey, P. (2016). Introduction to MANET‖. International Journal of Scientific Research in Network Security and Communication, 4(2), 15-19.

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. Theoretical computer science, 344(2-3), 243-278.

Ghaleb, S. A. M., & Vasanthi, V. (2020). Energy Efficient Multipath Routing Using Multi-Objective Grey Wolf Optimizer based Dynamic Source Routing Algorithm for MANET. International Journal of Advanced Science and Technology, 29(3), 6096-6117.

Lee, G. M., Crespi, N., Choi, J. K., & Boussard, M. (2013). Internet of things. In Evolution of telecommunication services (pp. 257-282). Springer, Berlin, Heidelberg.

Sarkar, S. B. T. (2016). Ad Hoc Mobile Wireless Networks. CRC Press: Taylor & Francis Group.

Shilova, Y., Kavalerov, M., & Bezukladnikov, I. (2016, February). Full Echo Q-routing with adaptive learning rates: a reinforcement learning approach to network routing. In 2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW) (pp. 341-344). IEEE.

Tirkolaee, E. B., Goli, A., Bakhshi, M., & Sangaiah, A. K. (2019). An Efficient Biography-Based Optimization Algorithm to Solve the Location Routing Problem With Intermediate Depots for Multiple Perishable Products. In Deep Learning and Parallel Computing Environment for Bioengineering Systems (pp. 189-205). Academic Press.

Tirkolaee, E. B., Goli, A., Faridnia, A., Soltani, M., & Weber, G. W. (2020). Multi-objective optimization for the reliable pollution-routing problem with cross-dock selection using Pareto-based algorithms. Journal of Cleaner Production, 276, 122927.

Tokic, M. (2010, September). Adaptive ε-greedy exploration in reinforcement learning based on value differences. In Annual Conference on Artificial Intelligence (pp. 203-210). Springer, Berlin, Heidelberg.

Wang, F., Feng, R., & Chen, H. (2019). Dynamic Routing Algorithm with Q-learning for Internet of things with Delayed Estimator. E&ES, 234(1), 012048.

Yadav, P., & Hussain, M. (2017, April). A secure AODV routing protocol with node authentication. In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) (Vol. 1, pp. 489-493). IEEE.