Journal of Computer and Knowledge Engineering, Vol. 1, No.2, 2017.
DOI: 10.22067/cke.v1i2.63668

A Novel Routing Algorithm for Mobile ad-hoc Networks Based on Q-learning and its Generalization to FSR Routing Protocol

Mahmoud Alilou Abdolreza Hatamlou*

Abstract: This study proposes a novel routing algorithm using Q-learning. Q-learning is a machine learning (artificial intelligence) algorithm using the reinforcement learning policy which can be used to solve problems for which there are different ways to reach their goal. The proposed algorithm, the Modified Q-learning routing algorithm (MQRA), has eliminated the episodes of Q-learning required to gradually learn in different stages and this has made it a rapid routing algorithm. MQRA can be used in various types of networks. This study uses MQRA in mobile ad-hoc networks, its generalization to fisheye state routing (FSR) (a routing algorithm) and its performance results are compared with the standard FSR. Experimental results confirm the applicability and potential of the proposed algorithm.

Keywords: Routing Algorithm; Mobile ad-hoc Networks; FSR Protocol; Reinforcement Learning; Routing in MANETs.

1. Introduction

In mobile ad-hoc networks (MANETs), nodes or workstations, trying to send information, are constantly moving and their neighbors are always changing. Thus, finding the current position of each node and the path to send the information packages has become one of the most important problems of such networks. Various routing algorithms use different methods to find a route and reinforce a particular component based on the policy used in sending information packages through the network. Thus, it is natural that concentrating on reinforcing one parameter would lead to distraction from the weakness of other ones. For instance if rapid delivery of the packages is important in a network, the routing algorithm loses more time in the routing phase, while finding the shortest path or it may have to select either an unsure short path or a safe long path to send the information packages. Another significant problem of mobile networks is the energy consumption of the nodes to process, store and send packages through the network. A light and intelligent algorithm can mitigate power consumption of the network and affect its lifetime with a given amount of energy. Making an algorithm more intelligent usually requires more information about the network, more computation makes the algorithm more time complex and usually it is not possible to make an algorithm both light and intelligent. Therefore, the proposed algorithm is presented to find the optimized route with the least possible amount of computation and in a shorter time than other similar algorithms [1, 2].

Q-learning Algorithm

- Input:State space S, Act.Space A
- Discount 7(0<=7<1);Learning a(0=<a<1)
- Outputs:Q
- Repeat{
  - S=get_current_word_state()
  - a=pick_next_action(Q.s)
  - (r.s)=act_int_world(a)
  - Q(s.a)=Q(s.a)+α*(r+7*max_$a'(Q(s'.a'))$-Q(s.a))
- } Until (bored)

Fig 1. The standard Q-learning algorithm

2. Literature Review

MQRA significantly changes Q-learning as shown in Fig. 1 and shows great performance while routing. Standard Q-learning executes stages to obtain a path between the origin and the destination which is the shortest path between those nodes. The proposed algorithm eliminates all stages to find this path as shown in Fig. 2 and finds all paths between the origin and the destination in a shorter time than the standard Q-learning instead of just one path.

```
// MQRA Algorithm
 Repeat {
  for (i=0;i<dim_X;i++)
   for (j=0;j<dim_Y;j++)
    if(Route_table[i][j] != 1.0) {
     max_value = α * maximum_adjacent_node_value();
     Route_table[i][j] = max_value;
    } // End if
 } Until (converge)
```

Fig 2. MQRA for a mesh network.

In the equation shown in Fig. 2, α is a variable which can vary in relation to bandwidths of the links of each node. Thus, instead of computing short paths by the number of steps, optimized paths are obtained by the amount of transferred data. However, for the simplicity of the example here, α is considered constant and equal to 0.98.

We describe the algorithm by an example of a given network. Take an 8x8 mesh, thus we have an 8x8 routing table and we assume that node (5, 5) is the destination and node (0, 1) is the origin. Therefore, the value of (5, 5) in the routing Table is 1 and the rest of the Table has zero values.

Manuscript received April 10, 2017; accepted November 27, 2017.
M. Alilou, Department of Computer Science, Khoy Branch, Islamic Azad University, Salmas, Iran.
*A. Hatamlou, Dept. of Computer Science, Salmas Branch, Islamic Azad University, Khoy, Iran. hatamlou@iaukhoy.ac.ir.

We start from slot 0 of the Table, move from left to right, top to bottom and set the value of each slot as α multiplied by the maximum value of its neighbors. If the neighbors with maximum values are all equal, one of them is selected by default.

Table 1. The initial routing table.

0	1	2	3	4	5	6	7	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3
0.00	0.00	0.00	0.00	0.00	0.98	0.96	0.94	4
0.00	0.00	0.00	0.00	0.98	1.0	0.98	0.96	5
0.00	0.00	0.00	0.00	0.96	0.98	0.96	0.94	6
0.00	0.00	0.00	0.00	0.94	0.96	0.94	0.92	7

In Table 1, first values of all slots equal to 0, except for the destination. After the first run of MQRA, this table is changed as shown in Fig. 3 and after about 8 iterations of this algorithm, the routing table is converged and all paths between the origin and the destination are obtained as shown in Fig. 4.

Table 2. The converged routing table.

	0	1	2	3	4	5	6	7
0	0.81	0.83	0.85	0.86	0.88	0.90	0.88	0.86
1	0.83	0.85	0.86	0.88	0.90	0.92	0.90	0.88
2	0.85	0.86	0.88	0.90	0.92	0.94	0.92	0.90
3	0.86	0.88	0.90	0.92	0.94	0.96	0.94	0.92
4	0.88	0.90	0.92	0.94	0.96	0.98	0.96	0.94
5	0.90	0.92	0.94	0.96	0.98	1.0	0.98	0.96
6	0.88	0.90	0.92	0.94	0.96	0.98	0.96	0.94
7	0.86	0.88	0.90	0.92	0.94	0.96	0.94	0.92

3. The FSR protocol

FSR is the reinforced protocol of GSR (both of which are based on the link state). Updating messages uses a significant amount of the bandwidth in GSR.

Fig. 3 presents an example of the fisheye boundary for node circles in red. This boundary is determined by the number of hops needed to reach a certain node. Not all updating messages of FSR contain all the information of the nodes. However, more information is provided about closer nodes than the farther ones. This decreases the size of updating message. The information of a node about its neighbors is updated frequently. However increasing the distance mitigates information validation. This process of dividing the network to different boundaries is performed for each node which means that there is no central node responsible for this division [4].

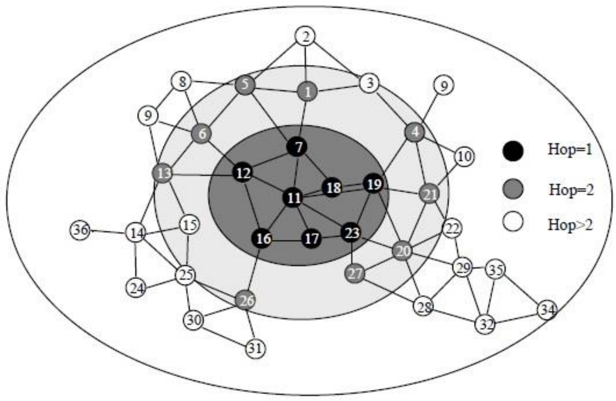


Fig 3. Boundary of the fisheye [3].

Despite the invalidity of information related to far neighbors, the routing procedure works correctly since approaching the destination increases the precision of information. This protocol is suitable for large-scale networks, since the protocol overhead is controlled.

The fisheye state routing protocol is table-driven (a proactive routing protocol). As it was mentioned, FSR is based on link state routing and it is able to provide the path information when needed. The link state package is exchanged periodically, not event-driven and the topology table is only sent to local neighbors instead of propagating in the entire network. The order of numbers is used to arrange the rows of the table, so that no row has the same number and thus routing is done with no cycles.

Updating messages of the nodes in smaller boundaries are more precise, since they send their routing tables more frequently; that is, nodes close to each other receive tables more frequently. However, the precision of farther nodes is mitigated, since it takes longer to exchange tables. Nonetheless, there is no need to find the path as done in demand based routing algorithms.

The fisheye boundary enables sending link state messages to nodes in different locations of the fisheye boundary in different time intervals. This leads to reducing the size of the link state packages.

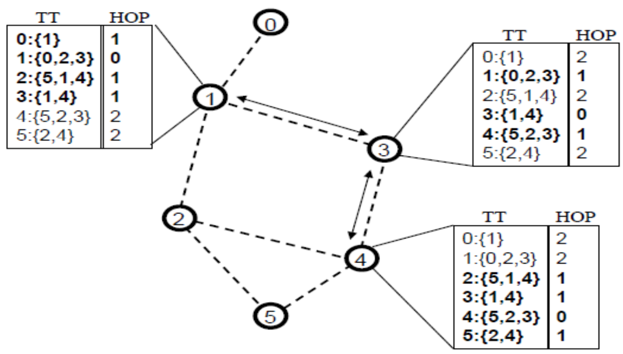


Fig 4. Reducing the message using fisheye [3, 5].

The highlighted row of table.12 is propagated more frequently to its neighbors since it has less hops. The TT column presents the neighbors.

Each node in FSR has the following information: The topology table, the link state list of neighbors, the routing table, Pros and Cons. The topology table is created using the information from link state messages. Each node has one slot in this table (the entire topology map). Each slot consists of two parts: the link state information and the destination order number. The routing table is created according to the information of this table. Information related to distance is obtained after creating the routing table and its information is used to determine the fisheye boundary for a node.

The topology table has the following information in each row: destination address, destination order number and the link state list. While receiving a link state message, the receiving node registers or updates the sender in its neighbors list. If it receives nothing from its neighbor after a certain time out, the corresponding row is deleted from the neighbors list. Each node stores the link state and the last time stamp of its neighbors. The routing table provides the next hop information to send the package to its destination. The rows of this table are varied if the topology table is changed. The rows of the routing table present the destination address and the next hop address.

FSR is suitable for large mobile networks since it is not sensitive to link malfunction through control messages. The malfunction links are not considered in exchanging the next connection messages and that means link changes do not necessarily change the routing tables. FSR is a simple method due to using the shortest updated paths. It is also a robust method due to exchanging a part of the updated message with only its neighbors which reduces traffic.

It is easy to find the destination, since the topology map and a simple addressing scheme is used. The drawback of this protocol is the complicated storage of the routing table, the computation overhead and also its inability to provide security as other protocols do [4-7].

3-1. Generalizing MQRA to FSR

In MQRA generalized to FSR which we call "MQ-FSRA", each node sends its score equal to 1% of its value together with its ID to all direct neighbors. Each receiving node stores the ID, the score and sends a percentage of the received score, as well as the ID of the first node to its neighbors.

Consider two nodes labeled A and Z in Fig. 7. A node which has a score coefficient equal 1.0 sends its label and a percentage of its score to its direct neighbors. All neighbors repeat the same so that A is identified in the entire network. All nodes, e.g. Z, do the same to be identified in the network. Fig. 5 is designed assuming A as centroid and each node like A belongs to an area with its corresponding centroid. The areas shown represent the frequency of sending packages. This means that for instance A sends its information, including label, a percentage of its score and other necessary information, more frequently in a limited area highlighted in the Figure. This frequency is reduced for farther areas and information packages are sent less frequently.

To clarify this point, pay attention to the eighth node in Fig. 7. This node, as it was considered beforehand, has saved some coefficients to send packs to nodes A and Z. the closeness coefficient of node 8 to node A is 0.94. This means that in case node 8 is supposed to send a pack to A, it must be sent through a neighbor that has a higher closeness coefficient to A than itself.

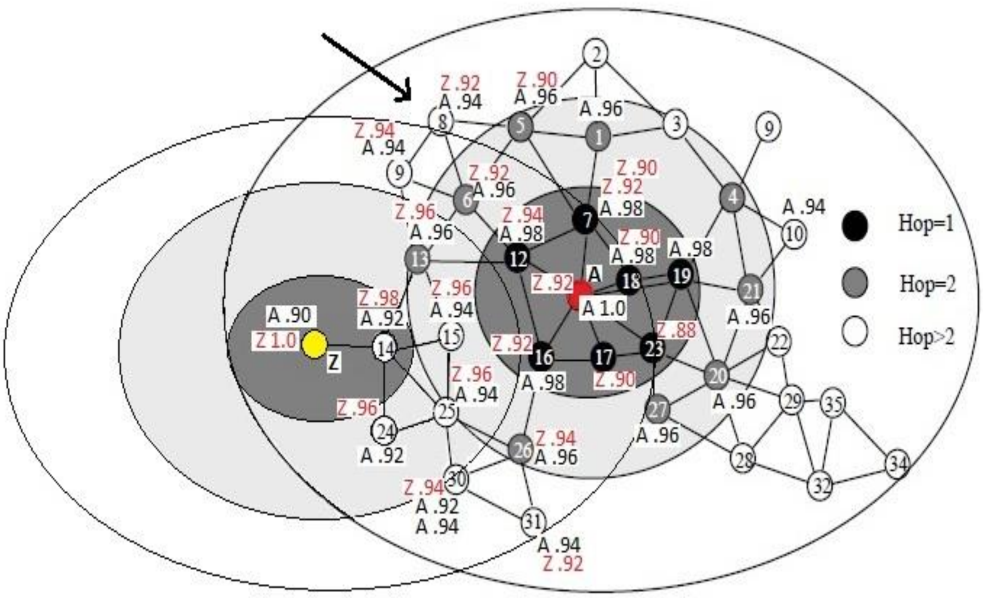
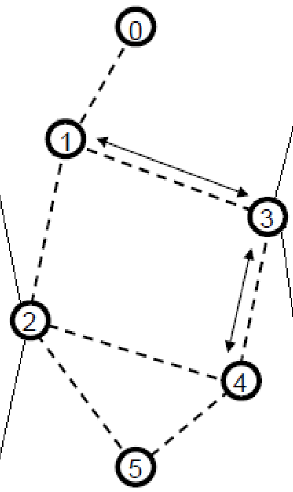


Fig 5. Routing in MQ-FSRA

Dest	Next hop	Near factor
0	1	0.9604
1	1	0.9800
2	--	1.0
3	1	0.9604
4	4	0.9800
5	5	0.9800

Dest	Next hop	Near factor
0	1	0.9604
0	4	0.9223
0	5	0.9039
1	1	0.9800
1	4	0.9411
1	5	0.9223
2	--	1.0
3	1	0.9604
3	4	09604
3	5	0.9411
4	4	0.9800
4	5	0.9604
4	1	0.9411
5	5	0.9800
5	4	0.9604
5	1	0.9223



Dest-ination	Next hop	Near factor
0	1	0.9604
0	4	0.9223
0	4	0.9039
1	1	0.9800
1	4	0.9411
1	4	0.9223
2	4	0.9604
2	1	0.9604
2	4	0.9411
3	---	1.0
4	4	0.9800
4	1	0.9411
4	1	0.9223
5	4	0.9604
5	1	0.9411
5	4	0.9411
5	1	0.9223

Fig 6. The navigation table structure in MQ-FSR.

In the example above, node 8 must choose node 6 or 5 in the next step to convey a pack to A, and if it wants to send a pack to node Z, the next step will be sending a pack to node 9. In this way, dispatch of any pack from any point in the net to the desired destination is possible through the shortest way.

With close attention to Fig. 5 we will notice that MQ-FSR easily supports Multi Pathing without calculating and saving any extra data in comparison with single pathing.

In order to dispatch any pack to destination, any node might simultaneously choose various neighboring nodes with higher closeness coefficients and choose one of the neighbors under the same circumstances. Choosing a neighbor can take place haphazardly or intelligently. The next step, for instance, can be based on the battery level in case some neighbors are under the same conditions. Namely, to choose the node among the neighbors that has a higher battery supply or to choose the neighboring node that has fewer tasks in buffer queue awaiting to be processed.

Pay attention to node 2 in this hypothetical net. As it can be seen in contrast with FSR list, the neighbors of other nodes are not saved in the table of this node. Moreover, instead of the number of paces, the closeness coefficient to the destination is mentioned. In this table, in case there are multiple routes to a destination. The longer ones with lower closeness coefficients can be eliminated and an optimum table can be produced by reduction of navigation table rows.

Therefore, less amount of information is propagated through the network. However, routing is performed correctly as mentioned before. This is so because as packages approach their destinations, the information related to the destination becomes more precise and the package is guided to its destination.

3-2. The Advantage of MQ-FSRA to FSR

As we can see in Fig. 6, FSR always propagates the network topology to direct neighbors of each node with different frequencies. This makes two problems arise which are not inherent to MQ-FSRA. First, each node must store its direct neighbors and a list of neighbors of other nodes. This makes each node identify its neighbors through sending and receiving packages and sending collected information to other nodes which consumes a significant amount of the energy of the network. Second, there is the problem of node dependencies; that is, nodes must try to send their packages by their neighbors whose information is propagated through the entire the network. This means an implicit dependency between nodes which makes the network update itself frequently. This is more intensified when velocities of nodes are great or nodes often fail. Obviously, none of these two problems exist in MQ-FSRA, since no list of neighbors is sent, there is less amount of information, there is no need to collect the information related to neighbors and there is no dependencies between neighbors. Wherever nodes are located in the network, they only have to send their information to one node which has a higher score.

4. Simulation Results

In the experiments that were conducted, nodes are mobile and the amount of their mobility is 0.5ms. The amount of energy is limited and the same for all nodes. The network space is 100x1000 and the number of nodes varies depending on the experiment; that is new nodes can join or leave the network during its lifetime. All facilities of the nodes are the same, wireless transmission is used for sending and receiving with a maximum bandwidth of 50Mbit/s and the radius of 30m.

In this section, we evaluate MQ-FSRA with important policies of routing protocol evaluations, i.e. average routing overhead and average package loss, and also compare the results with FSR. The following simulation results indicate that MQ-FSRA provides better results using these policies.

Routing overhead is the ratio of total number of sent control packages to total data packages received successfully at the destination. Figs. 7 and 8 present average routing overhead and average packet loss of FSR and MQ-FSRA. These diagrams consider the overhead amounts separately according to the number of current nodes of the network and the velocity of the nodes. Also average package lost is given based on node failures. As we can see in Figs. 7 and 8, MQ-FSRA has better performance compared with FSR, through reduction of navigation table rows.

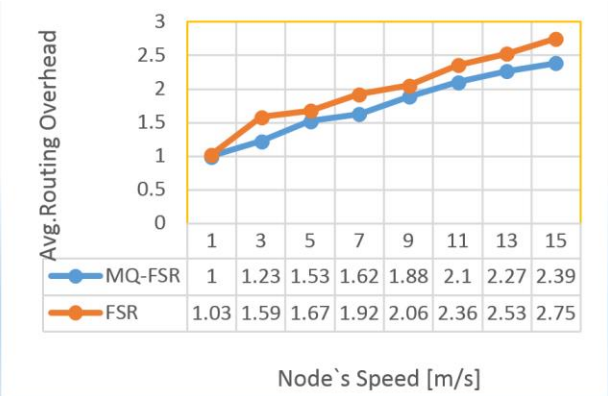


Fig 7. Average routing overhead according to cost functions with different number of nodes.

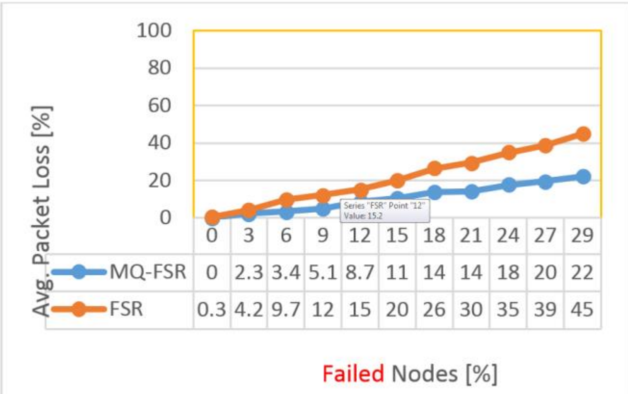


Fig 8. Average package loss according to the proportion of node failures.

5. Conclusions

As it was discussed in this study, the novel proposed algorithm called "MQRA" is a light and rapid algorithm, which can adapt to the environment and it can also be generalized to various protocols. Generalizing the proposed algorithm to FSR significantly reduces FSR computations and eliminates node dependencies. This leads to long-term updates of MQ-FSRA and imposes less overhead in the path finding phase and routing reconfiguration. Node independency reduces package loss due to node failures and path disconnection while sending the package. MQ-FSRA needs less amount of stored information to find a route compared to FSR since in contrast to FSR, MQ-FSRA does not need to store the information about direct and indirect neighbors. Therefore, it requires less memory and consumes less energy. Adding facilities like GPS to MQ-FSRA enables provision of new services which we intend to discuss in another paper. However, FSR does not predict utilizing such capabilities.

References

[1] R. S. Sutton and A. G. Barto, ―Reinforcement Learning: An Introduction,The MIT Press, 1998.

[2] A. F¨orster, ―Machine learning techniques applied to wireless ad-hoc networks: Guide and survey, ‖ *in Proceedings of the 3rd International Conference on Intelligent Sensors,* Sensor Networks and Information Processing (ISSNIP), 2007.

[3] G. Pei, M. Gerla and T.-W. Chen, "Fisheye State Routing in Mobile Ad-Hoc Networks", *In Proceedings of the 2000 ICDCS Workshops,* Taipei, Taiwan, pp. D71-D78, Apr. 2000.

[4] L. Kleinrock and K. Stevens, "Fisheye: A Lenslike Computer Display Transformation", *Technical report, UCLA, Computer Science Department,* 1971.

[5] T.-W. Chen and M. Gerla, "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks", *In Proceedings of IEEE ZCC'98,* Atlanta, GA, pp. 171-175, Jun. 1998.

[6] Gyanappa A. Walikar, Rajashekar C. Biradar , A survey on hybrid routing mechanisms in mobile ad hoc networks PP. 48-63. doi>10.1016/j.jnca.2016.10.014

[7] Fahimeh Dabaghi, Zeinab Movahedi, Rami Langar, A survey on green routing protocols using sleep-scheduling in wired networks, PP. 106-122. doi>10.1016/j.jnca.2016.10.005.

[8] Shafiee, Kaveh, and Victor Leung. Connectivity-aware minimum-delay geographic routing with vehicle tracking in VANETs. Ad Hoc Networks 9.2 (2011): 131-141.

[9] Ding, Zhiguo, and Kin K. Leung. Cross-layer routing using cooperative transmission in vehicular ad-hoc networks. Selected Areas in Communications, IEEE Journal on 29.3 (2011): 571-581.

[10] Sofra, Nikoletta, Athanasios Gkelias, and Kin K. Leung. Link residual-time estimation for VANET cross-layer design. Cross Layer Design, 2009. IWCLD'09. Second International Workshop on. IEEE, 2009.

[11] Al-Rabayah, Mohammad, and Robert Malaney, "A new hybrid location-based ad hoc routing protocol", *Global Telecommunications Conference (GLOBECOM 2010),* 2010 IEEE. IEEE, 2010.

[12] Al-Sultan, Saif, Ali H. Al-Bayatti, and Hussein Zedan, "Context-aware driver behavior detection system in intelligent transportation systems", *Vehicular*

Technology, IEEE Transactions on 62.9 (2013): 4264-4275.

[13] Yang, Qing, et al. "ACAR: adaptive connectivity aware routing protocol for vehicular ad hoc networks", *Computer Communications and Networks,* 2008. ICCCN'08. Proceedings of 17th International Conference on. IEEE, 2008.

[14] Schaul, T. Bayer, J. D. Weirstra, Sun, T. "PyBrain", *Journal of Machine Learning Research,* vol 11, no.2, 2010.

[15] Kulkani, S., Rao, R., "Performance Optimization of Reinfrocement Learning Based Routing Algorithm Applied to Ad hoc Networks", *International Journal of Computer Networks and Communications,* vol. 2, pp. 46- 60, 2010.