Research Journal of Engineering & Technology. Vol., 1 (1), 13-21, 2024
Available online at http://www.rjetjournal.com
ISSN 2149-3340 ©2024

A Intelligent Wireless Ad hoc Routing Protocol to Increase Throughput in Mobile Ad hoc Networks

Shayesteh Tabatabaei[1*], Mohammad Teshnehlab[2]

[1]*Department of Computer, Science and Research branch, Islamic Azad university, Tehran, Iran*
[2]*Control center of Excellence, faculty of Electrical and computer Engineering, K.N.Toosi University of Technology, Tehran, Iran*

[*]Corresponding Author Email: shtabatabaei@malekaniau.ac.ir

Abstract: Ad hoc networks are dynamically configurable wireless networks that have no fixed infrastructures and do not require predefined configurations. In large, distributed systems, like ad hoc networks, centralized learning of routing or movement policies may be impractical. We need to employ learning algorithms that can learn independently, without the need for extensive coordination. A search for alternative methods of routing packets has resulted in reinforcement learning (RL) as a good approach to adaptive routing. RL methods are able to learn and adapt to a unknown and changing environment. In This paper we propose a new intelligent routing protocol with RL approach for wireless ad hoc networks. The Intelligent Wireless Ad Hoc Routing protocol (IWARP) is an on-demand protocol and self-configuring. It selects optimal routes based on local information and past experience. The proposed scheme uses the distributed Q-Learning framework, to select a stable route in order to enhance system performance. Our study also compares the performance of the IWARP protocol with the well-known Ad hoc On-Demand Distance Vector (AODV) protocol, Results obtained by a simulation campaign show that IWARP increases the throughput and decreases the data dropped and number of hops per route.

Keywords: Mobile ad hoc Networks (MANET), Ad Hoc Network, IWARP, RL.

Introduction

A Mobile Ad Hoc Network (MANET) is a network of wireless mobile nodes that does not rely on any base stations or fixed infrastructure. Communications are made using multi hop wireless links. Therefore each node also acts as a router, forwarding data packets for other nodes. Low-cost mobile devices such as laptops and palmtops are becoming widely available, making large scale ad hoc networks in the civil field

A realistic possibility: Dynamism, absence of costly and often cumbersome infrastructure, decentralization and robustness are the main advantages of MANETs. This makes them particularity adapted to disaster recovery situations, interactive courses or meetings. Future application possibilities are endless, from sensor dust networks to multi hop wireless broadband Internet access, wireless imaging, cooperative car traffic monitoring, Personal Area Networks, etc. The difficulties inherent in this type of network come from the limited computing resources and bandwidth, as well as the potential high degree of node mobility and energy limitations. Therefore, ad hoc routing protocols must be efficient, with a low load on the network and at the same time he accurate in their decision-making mechanism (Chetret et al., 2004).

Efficiently routing in a dynamic network is an important problem due to network's frequent and unpredictable change in topology and characteristics. There is no center server which knows all node information. Therefore, it is not appropriate to apply conventional routing algorithms (such as "Routing Information Protocol" or "Open Shortest Path First"). In order to solve these problems, routing algorithm based on reinforcement learning is proposed. This learning algorithm use only local information stored in each node to decide routing path.

Learning is advantageous because it allows the mobile nodes to quickly react to changes in network configuration and conditions. Moreover, network configuration and conditions are by definition expected to change frequently. This partial observably is inherent to both the routing and movement portions of the mobile nodes, since there is no central network administrator. Mobile nodes do not have access to the global network topology or communication patterns. Even with this limited knowledge, learning is useful because it would otherwise be difficult for a human designer to create an optimized movement policy for each network scenario.
This type of problem lends itself to reinforcement learning techniques, where the goal of the learner is to maximize long-term reward by learning an optimal behavior policy through simulation (Oužecki & Jevti, 2010).

In this paper we proposed an algorithm that presents a new intelligent routing protocol with Q-Learning algorithm for wireless ad hoc networks. an algorithm that attempts to optimize paths taken through a network by optimizing aggregate end-to end latency through a reinforcement learning approach. Q-Learning attempts to optimize the end-to-end performance of network paths and balance load across the network by using learned information.

The Intelligent Wireless Ad Hoc Routing protocol (IWARP) selects optimal routes based on the current network status and past experience. The IWARP selects routes based on mobility, available bandwidth, Hop count and Power of battery. IWARP adapts its routing decision based on the current network status and past experience.

This paper is organized as follows. In section II, we review the AODV Routing Protocol. Section III describes the Reinforcement learning. Section IV describes the newly introduced protocol, Intelligent Wireless Ad Hoc Routing protocol (IWARP), Section V presents the simulation results based on a mobile network example, and conclusions are presented in Section VI.

Ad-hoc on-demand distance vector (AODV) routing protocol OVERIEW

The starting point for our protocol was to modify an existing on demand routing protocol for best effort Ad hoc networking.

We took the well-known Ad hoc On-Demand Distance Vector protocol (AODV). AODV uses a broadcast route discovery mechanism, and it relies on dynamically established routing table entries at intermediate nodes. The functions performed by AODV protocol include local connectivity management, route discovery, route table management and path maintenance. Local connectivity management may be summarized as follows. Nodes learn about their neighbors by either receiving or sending broadcast packets from or to their neighbors. Receiving the broadcast or HELLO from a new neighbor or failing to receive HELLO packets from a node that was previously in the neighborhood, indicates that the local connectivity has changed.

The source node initiates path discovery by broadcasting a route request, RREQ, packet to its neighbors. When a node receives an RREQ, in case it has routing information, it sends the reply packet, RREP, back to the destination. Otherwise, it rebroadcasts the RREQ packet further to its neighbors. As the RREQ packet travels from the source to the destination it automatically sets up the reverse path for all nodes back to the source. As the RREP travels back to the source, each node along the path sets up a forward pointer to the node from which the RREP came and updates its timeout information for route entries to the source and destination.

For each destination of interest a node maintains a single route table entry that contains the address of the destination, the next hop along the path to that destination, the number of hops to the destination, and other route related parameters. If a node is present with two different routes to the destination it chooses the fresher route. If both routes were discovered simultaneously, the route with fewer hops is preferred. Path maintenance is performed in several ways. When any node along an established path moves, so that some of the nodes become unreachable, a special RREP packet is sent to affected source nodes. Upon receiving notification indicating a broken link, the source node restarts the path discovery process, if it still needs that route (Afandi, 2006).

Reinforcement Learning

Reinforcement learning is about learning from interaction how to behave in order to achieve a goal. The reinforcement learning agent and its environment interact over a sequence of discrete time steps. In every state, the agent chooses an action a_t according to his policy and receives reinforcement r_t, also known as reward, from the environment evaluating his actions. On the basis of the reinforcement received and the state reached after taking the chosen action, the agent modifies his policy in order to attain his goal, which is typically a discounted sum of all rewards (Afandi, 2006).

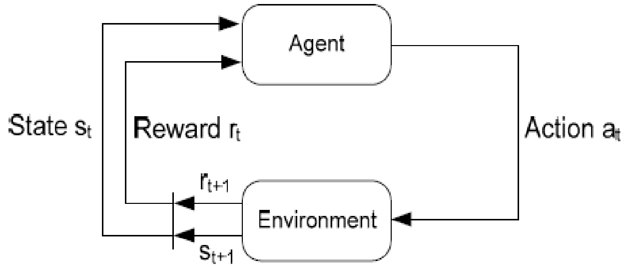

Figure 1. The framework of reinforcement learning.

An effective way to achieve reinforcement learning is to use the Q-learning algorithm (Watkins & Dayan, 1992).

The Q-Learning problem model consists of an agent, states' S and a number of actions a per state A. By performing an action a, where $a \varepsilon A$, the agent can move from state to state. Each state provides the agent a reward (a real or natural number) or punishment (a negative reward). The goal of the agent is to maximize its total reward. It does this by learning which action is optimal for each state. The algorithm therefore has a function which calculates the quality of a state-action combination

$$Q: S \times A \rightarrow R \qquad (1)$$

Where goal is:

$$a_t = \arg \ \max_a Q(s_t, a_t) \qquad (2)$$

Before learning has started, Q returns a fixed value, chosen by the designer. Then, each time the agent is given a reward (the state has changed) new values are calculated for each combination of a state's from S, and action a from A. The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information.

$$Q(s_t, a_t) \leftarrow r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \qquad (3)$$

Where r_t is the reward given at time t and γ is discount factor $(0 < \gamma < 1)$.

The discount factor determines the importance of future rewards. A factor of 0 will make the agent "opportunistic" by only considering current rewards, while a factor approaching 1 will make it strive for a long-term high reward. If the discount factor meets or exceeds 1, the Q values will diverge.

The Intelligent Wireless Ad Hoc Routing protocol (IWARP)

In this paper, we propose an Intelligent Wireless Ad Hoc Routing Protocol (IWARP) for MANETs.

The Intelligent Wireless Ad Hoc Routing Protocol (IWARP) is an on-demand protocol; it periodically adjusts routing traffic based on the network status.

Our goal is to improve routing quality in MANETs by using the distributed Q-Learning framework, based on Reinforcement Learning (RL) methods. RL, a machine learning technique, is used to evaluate routes and assign values to different parameters for each route (Afandi et al., 2006). The goal is making the quality of path finding in MANET networks with using accessible data in network. This idea desire for choosing a stable path for:

Decreasing overhead for path finding, decreasing number of hop for presenting in path finding and optimized use of powerful and energetic paths. Before representing the suggested method, introduce some important parameters:

Hop count (HC): this parameter as the distance (in hops) between the routers otherwise the HC is the number of hops for a feasible path. The smaller the HC is, the more reliable the routing path. Their route selection algorithm prefers the route that reaches the destination node first. The reason for selecting this route is that this route is less congested than the others. Of course, this route may contain highly mobile nodes, which make link failure more likely. It is, therefore, more reasonable to look at the hop count to make the final selection of a route (Tseng & Chang, 2003). If each intermediate host has a large roaming area, and the MANET has many nodes (and hops), then a feasible path with a low hop count is preferred. When a source node wants to send data to a destination node for which there is no entry in its route cache, route discovery is initiated and a route request is broadcasted. The destination node D knows the HC value of the three feasible paths via the hop count in the RREQ packet. As shown in Fig. 2 for Path I (S, A, K, H, D), HC1 = 4. For Path II (S, B, E, D), HC2 = 3. For Path III (S, C, F, G, D), HC3 = 4.

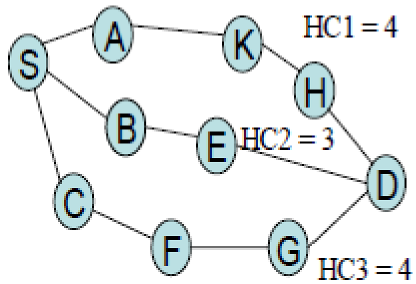

Figure 2. Hop count of each feasible path.

Bandwidth metric (BW): this parameter show, account necessary bandwidth for sending a file. Generally, the method to define the bandwidth metric is to calculate the available bandwidth between two adjacent nodes with the same speed which are separated by a direct and symmetric link. A simple method in (Badis, 2003) gives the available bandwidth based on the transmission speed which can be used to measure the bandwidth for a (i,j) link. This link has an available bandwidth of:

Available _Bandwidth (i , j) = (1-u)*Bandwidth(i, j)   (4),

Where u is the link utilization (i.e. u = A (t)/t, A(t) is the total amount of time where the link is used by nodes during an interval of time t).

Power of battery (PB): With using the parameter, account necessary energy for complete sending a file or data before transforming data packages with considering the size of packages.

Speed (SP): this parameter show rate of mobility for mobiles.

The reliability of a feasible path is based upon four factors: the mobility of nodes, the power level of battery, the Bandwidth, and the hop count.

We also suppose quantity Threshold value for each four parameters as shown in Figure 3. If we parameters there is greater than Threshold Then the reward of this parameter can be made +1 else be made -1.

For example, the power level of battery is very important in MANET. We prefer the value of power level of battery to be Maximum. Thus, if the value of power level of battery be smaller than itself threshold, the reward of this parameter can be made -1 else be made +1, or we suppose the threshold's for the Bandwidth, 5500000.0. We want to own high Bandwidth thus if the value of Bandwidth be greater than 5500000.0, in other words, we want this parameter to be in Maximum span. Thus, if this parameter to be in Maximum span, the reward of this parameter can be made +1 else be made -1, also we want to own small Hop count thus if the value of Hop count be greater than itself threshold, the reward of this parameter can be made -1 else be made +1.

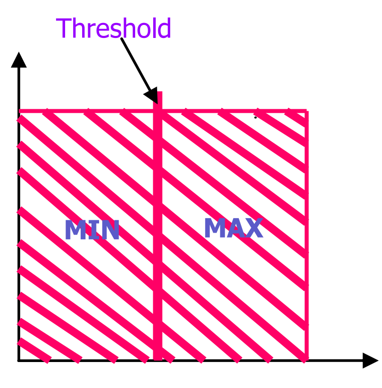

Figure 3. Threshold value for parameters.

Path finding table structure is the working of this protocol that when a source node wants to send a data to an away node these steps must followed:

a. When the source S receives a packet from the transport layer in direction of destination D, it checks if a route exists to the destination. If it already has a route, it transmits the packet to the next hop node. Else, it transmits a RREQ request.

b. When a node X receives a RREQ request with a source S, a destination D and a source-sequence-number sqn_1:

X calculates the reward Q_i with the formula (5)

$$Q_i(S,X) \leftarrow R_i + \gamma \text{ Max } Q_{i+1}(X,D) \qquad (5)$$

Where R_i is the reward given with a source S and a node X , $(R_i = R_{HC} + R_{BW} + R_{PB} + R_{SP})$ it is sum of immediate reward of four factors and γ is discount factor $(\gamma = 0.9)$.

If the pair < S, D> exists in the reverse route entry table with a reward Q_1 and a source- source-sequence-number denoted sqn_2

If ($Q < Q_1$and $sqn_1 = sqn_2$) or ($sqn_1 > sqn_2$)
X updates this reverse route entry table
• If X # D
X broadcasts the RREQ request up to date
• Else
X sends a RREP request
Else
X discards the RREQ request
Else
X creates a new pair < S, D > with the reward equal to
Q and a source-sequence-number equal to sqn_1
If X = D
X sends a RREP request

c. When a node X receives a RREP request from node Y with a source S, a destination D, a reward Q_i, a source-sequence-number sqn_1:
X calculates the reward Q_i with the formula (5)
If the pair < S, D > exists in the routing table with a reward Q_2 and a source-sequence-number denoted sqn_2
If ($Q_1 < Q_2$ and $sqn_1 = sqn_2$) or ($sqn_1 > sqn_2$)
X updates the routing table with the next hop y
• If X # D
X forwards the RREP to the previous node
Else
X discards the RREQ request
Else
X creates a new tuple in the routing table with the next hop y
If X # S
X forwards the RREP to the previous node.

Results

In this section, we show the network model that we use and compare our algorithm with the AODV protocol.

A. Simulation Model

To simulate our algorithm, we use the OPNET modeler 10.5. The initial positions of the nodes were uniformly distributed throughout the network. Node mobility was simulated according to the random waypoint mobility model, in which each node travels to a randomly selected location at a configured speed and then pauses for a configured pause time, before choosing another random location and repeating the same steps. Node transmission range was 250m. We ran simulations for constant node speeds from 0 to 10 m/s, with pause time fixed at 200 seconds. We simulated 20 CBR sessions in each run, with random source and destination pairs. Each CBR session generates 10 packets per second with data packets of 512 bytes. In the simulation, the network coverage area is a 2117m * 2117m square with 35 mobile nodes. . We will use a simple topology, and Process model, as shown in Figure 4 and Figure 5. Simulation time is 720 seconds.

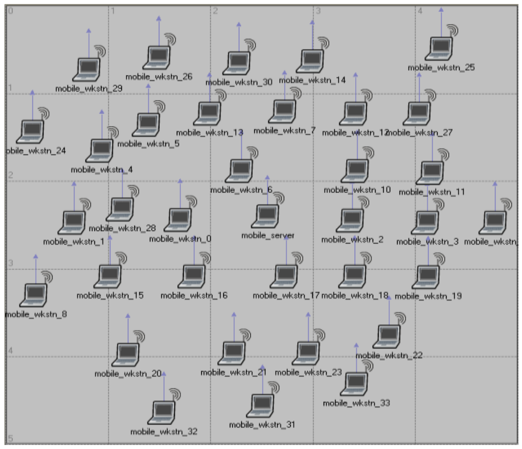

Figure 4. A sample topology for MANET.

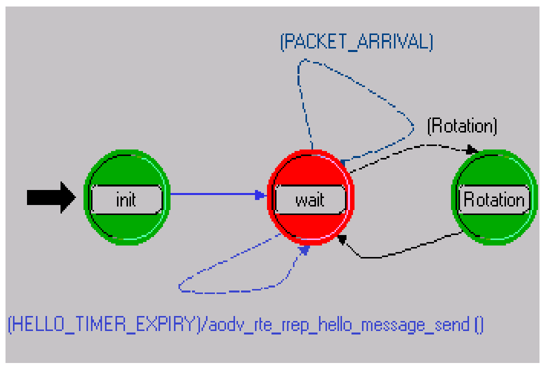

Figure 5. Process model for sample topology.

For the experimental evaluation, we have assumed the performance metrics in order to analyze the performance of the proposed IWARP protocol for ad hoc routing:

Throughput: Also called packet delivery ratio in (Broch et al., 1998) and throughput in (Michiardi & Molva, 2002), this is the ratio of the number of packets received by the CBR sink to the number of packets sent by the CBR source, both at the application layer. Packets that are sent but not received are lost in the network due to malicious drops, route failures, congestion, and wireless channel losses.

Hop Count: we determined the destination (hop count) for all possible next hops.

In figure 6 we show the average performance in terms of throughput, according to the Simulation Time in a network of 35 nodes. X-axis represents the Simulation time and Y-axis represents the throughput. The simulations of figure 6 show that IWARP performs better from the AODV. Most of the ad hoc routing protocols presuppose the presence of bidirectional links between the nodes in the network. In reality, the ad hoc network may consist of heterogeneous nodes with different power capabilities and different transmission ranges. When this is the case, a given node might be able to receive the transmission of another given node but might not be able to successfully transmit data to the latter (Wang et al., 2007). In IWARP protocol with using the Q-Learning determine the suitability of link and if the link is suitable, a given node is able to successfully transmit data to the latter. Thus, throughput of our algorithm is higher than the AODV algorithm.

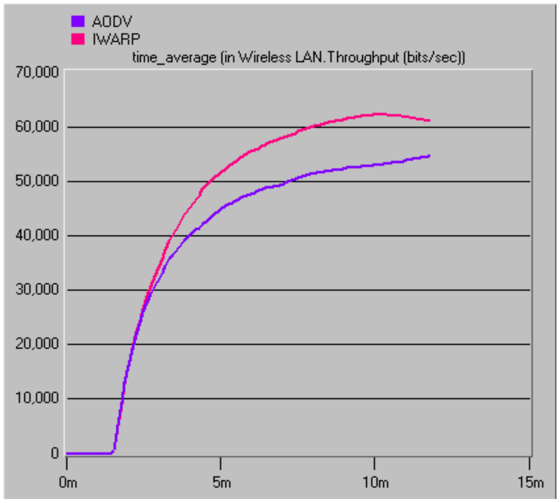

Figure 6. Throughput vs. Simulation time for 35 mobile nodes.

Figure 7 shows the number of RREQ requests sent by both algorithms. X-axis represents the Simulation time and Y-axis represents the total route request send. Our algorithm for all Simulation Time, generates more requests because when a node receives a request and the sequence number is lower than the sequence number the reverse route entry table, it forwards the request up to date.

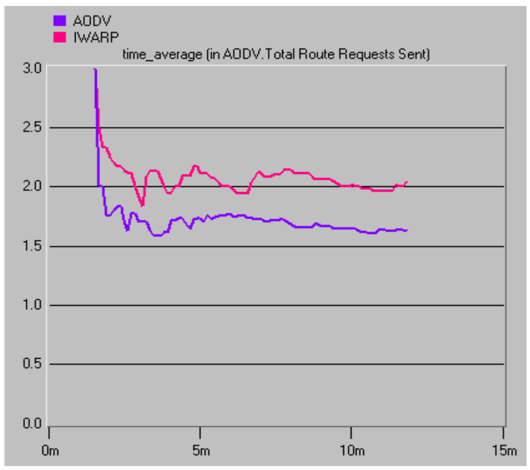

Figure 7. Packets control overhead.

Figure 8 shows the hop count of IWARP, and AODV for different Simulation Time. X-axis represents the Simulation time and Y-axis represents the number of hops per route. The simulations of Figure 8 show that finally, hop count of IWARP less than AODV.

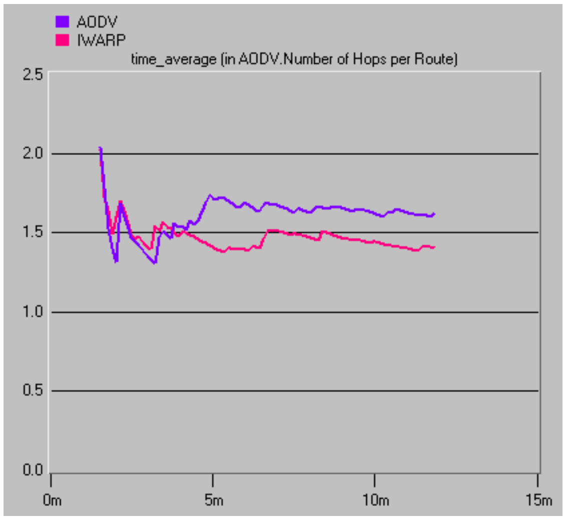

Figure 8. Hop count vs. Simulation Time for 35 mobile nodes.

Figure 9 shows the numbers of data dropped by both algorithms. X-axis represents the Simulation time and Y-axis number of data dropped. From Figure 9, we can conclude that IWARP algorithm can greatly decrease the numbers of data dropped. It shows that the improvements become more significant with the increase of Simulation time.

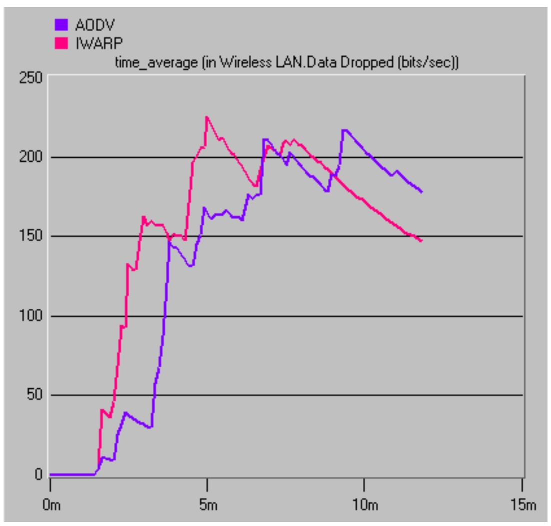

Figure 9. Numbers of data dropped vs. Simulation Time for 35 mobile nodes.

Discussion and Conclusion

In this paper, we provided a new intelligent routing protocol with RL approach for wireless ad hoc networks. Simulation results show that the IWARP algorithm improves the throughput and hops per route significantly, and also improves the network performance. For this, our algorithm adds overhead only in RREQ packets. We show simulation results of our algorithm under random variation of the network. From a performance point of view, our heuristic gives a path with a higher throughput than the original AODV protocol. Moreover, it decreases the hop count and numbers of data dropped.

References

Afandi H, 2006. An Intellig ent Wireless Ad hoc Routing Protocol. In University of Wisconsin_Milwaukee. [Google Scholar] [Publisher]

Afandi H, Hosseini H, Vairavan K, 2005. A Simulation Study of The Intelligent Wireless Ad hoc Routing Protocol. PE-WASUN'05. [Google Scholar]

Badis H, 2003. QoS for Ad hoc Networking Based on Multiple Metrics: Bandwidth and Delay. MWCN'03. [Google Scholar] [Publisher] https://doi.org/10.1142/9789812791146_0004

Broch J, Maltz D A, Johnson D B, Hu Y, Jetcheva J, 1998. A performance comparison of multi-hop wireless ad hoc network routing protocols. In MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking. New York, NY, USA, pp:85–97. [Google Scholar] [Publisher] http://dx.doi.org/10.1145/288235.288256

Chetret D, Tham C K, Wong L W C, 2004. Reinforcement Learning and CMAC-based Adaptive Routing for MANETs. IEEE. In Elsevier Inc. [Google Scholar] [Publisher] https://doi.org/10.1109/ICON.2004.1409226

Michiardi P, Molva R, 2002. Simulation-based analysis of security exposures in mobile ad hoc networks. In Proceedings of EuropeanWireless Conference. [Google Scholar] [Publisher]

Oužecki D, Jevti D, 2010. Reinforcement Learning as Adaptive Network Routing of Mobile Agents. MIPRO 2010, pp: 24-28. [Google Scholar] [Publisher]

Tseng Y C, Li Y F, Chang Y C, 2003. On route lifetime in multihop mobile ad hoc networks. IEEE Transactions on Mobile Computing. 2 (4): 366–376. [Google Scholar] [Publisher] http://dx.doi.org/10.1109/TMC.2003.1255651

Wang N, Huang Y, Chen J, 2007. A stable weight-based on-demand routing protocol for mobile ad hoc networks. [Google Scholar] [Publisher] https://doi.org/10.1016/j.ins.2007.05.030

Watkins J C H, Dayan P, 1992. Q-Learning, Machine Learning. Kluwer Academic Publishers, Vol 8. [Google Scholar] [Publisher]