Adaptive Q-routing with Random Echo and Route Memory

Maksim Kavalerov, Yuliya Shilova, Yuliya Likhacheva
Perm National Research Polytechnic University
Perm, Russia
mkavalerov@gmail.com, marissaspiritte@mail.ru, lijulia28@gmail.com

Abstract—Mobile ad hoc networks require routing algorithms that provide high performance in terms of delivery times of packets for dynamically changing topologies under various load conditions. A routing algorithm is proposed which is based on adaptive Q-routing technique with Full Echo extension. The proposed algorithm, called Adaptive Q-routing with Random Echo and Route Memory (AQRERM), has the improved performance in terms of overshoot and settling time of the learning. It also greatly improves stability of routing under conditions of high load for the benchmark example.

I. Introduction

Mobile ad hoc networks [1] require routing algorithms that maintain high performance, e.g., in terms of delivery times for packets, for dynamically changing topologies under various load conditions. Routing algorithms for ad hoc networks are surveyed in [2] [3]. Although, probably the most popular protocol for mobile ad hoc networks is AODV [4], its enhancement [5] based on reinforcement learning, namely Q-learning [6], seems very promising. Routing techniques based on reinforcement learning [7] can cope with dynamically changing conditions, using local information and globally gathered data, therefore they can be very efficient for mobile ad hoc networks, e.g., see the survey [8]. Q-routing algorithm [9] is a routing scheme which is based on Q-learning, model-free reinforcement learning method. Over the past several years Q-routing has been extended in many ways, for example, see Full Echo Q-routing [9], Dual Reinforcement Q-routing [10], Predictive Q-routing [11], Ant-Based Q-Routing [12], Gradient Ascent Q-routing [13], K-Shortest Paths Q-routing [14], Credence Based Q-routing [15], Q-probabilistic routing [16], Simulated Annealing Based Hierarchical Q-routing [17], Enhanced Confidence-Based Q-routing [18].

We focus on routing algorithms based on Q-routing because they provide flexible frameworks for implementation of reinforcement learning methods that can improve the performance of routing by using only local information or some data which pass through the network gathering the information about the global state of the network. In addition, these methods allow to balance exploration and exploitation during the learning process just by changing some parameters of the routing algorithm. The above-mentioned features of Q-routing based frameworks can be crucial for providing highest possible performance of routing for rapidly changing ad hoc networks with complex topologies. These networks can be estimated to be in great request in the near future.

This paper addresses the problem of balancing exploration and exploitation by introducing some enhancement to the previously developed routing algorithm, called Adaptive Q-routing Full Echo (AQFE) [19]. This algorithm is based on Full Echo extension of Q-routing, described in [9]. Full Echo extension significantly increases exploration by implementing the following method. Upon sending a packet, each node sends requests to its neighboring nodes in order to get the estimates of the delivery time for the routes provided by these neighbors. But under high load conditions, Full Echo Q-routing can lead to unstable delivery times caused by oscillating routes [9]. AQFE partly solves this problem by dynamically changing the learning rates that are used for updating estimates during Full Echo polling. This adaptation of learning rates allows to reduce exploration after the learning has settled. But instability in the average delivery time remains under some conditions, especially, when the load is high.

To overcome this, we propose the modification of AQFE based on Random Echo and Route Memory techniques. The Random Echo scheme implies that neighboring nodes are polled randomly taking into account the estimates of the average delivery time. The route memory technique is added to prevent the case when the packets return to the nodes already visited. The resulting algorithm, called Adaptive Q-routing with Random Echo and Route Memory (AQRERM), reduces instability under high load conditions and improves performance in terms of overshoot and settling time of the learning.

II. Background

A. Q-routing

The network consists of nodes, which can be considered as agents transmitting packets to their neighbors. The decision to transmit a packet is affected only by the packet's destination and so called Q-values stored in the table, also known as Q-table. Q-values are updated according to the estimates of the delivery time of packets. Let $Q_x(d,y)$ denote Q-value located at row d and column y in Q-table, and d is interpreted as the destination node and y corresponds to the neighbor y. Thus $Q_x(d,y)$ can be seen as the estimate of the delivery time that could be spent in transmitting a packet, destined for node d, by

ISSN 2305-7254

using neighboring node y as the proxy. By $P(s,d)$ we denote a packet originated at node s and destined for node d. Q-routing policy implies that $P(s,d)$ is sent to neighbor with minimal $Q_x(d,y)$ at row d in Q-table. By sending $P(s,d)$ to y, node x gets back y's estimate t for the time remaining in the route:

$$t = \min_{z \in N(y)} Q_y(d,z) \tag{1}$$

where $N(y)$ is the set of all y's neighbors. The following rule is used to update $Q_x(d,y)$:

$$Q_x(d,y) = Q_x(d,y) + \eta \cdot (q + s + t - Q_x(d,y)) \tag{2}$$

where η is called learning rate, q is the time spent in node x's queue, s is the transmission time between nodes x and y.

B. Full Echo Q-routing

The drawback of Q-routing is that it does not update Q-values greater than the minimal Q-value at the same row of Q-table. Indeed, if the same Q-value remains minimal then Q-routing policy implies that the update rule (2) is applied only to this Q-value, and other Q-values remain unchanged. At the initial stage of learning, the minimal Q-value is likely to become greater due to congestions in the related routes. That is why other Q-values are usually updated at this stage. But later, after the learning has settled, Q-values do not change so much, and consequently, the routes remain unchanged even if they are not optimal. This can lead to the increase in the average delivery time. Additional exploration, based on polling the neighbors, could solve this problem. Moreover, it can speed up the learning at the initial stage.

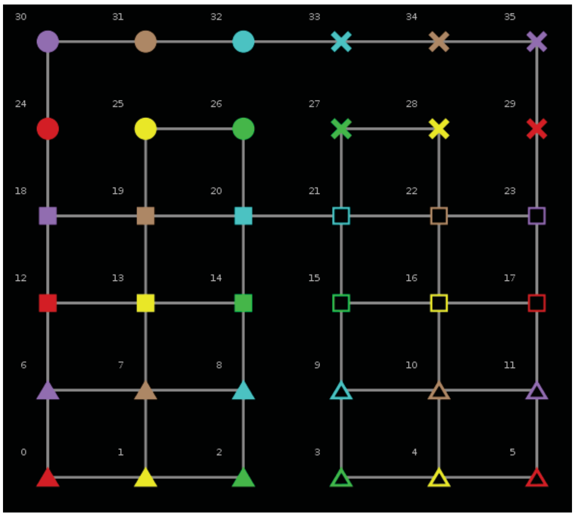

Fig. 1. The 6x6 grid network example used as a benchmark

To add such exploration to the policy, a modification called "full echo" has been proposed in [9]. This modification can be described as follows. When node x is going to choose the neighbor for forwarding the packet $P(s,d)$, all the neighbors are requested for the values t obtained by (1). The neighbors send back their values t and all the values in row d of node x's Q-table are updated accordingly. Q-routing with this modification is mentioned later as Full Echo Q-routing policy.

This policy can decrease the settling time and the overshoot of the initial stage of learning under low and medium load. But when the load gets higher, this approach can cause the routes oscillating between some bottlenecks of the network [9]. These oscillations may lead to significant variations in the average delivery time, thus increasing the average delivery time calculated for longer time intervals. Such effect can be seen in Fig. 2, which represents the average delivery time when Full Echo Q-routing is applied for the irregular grid network presented in Fig. 1. This network is a widely used benchmark, and it is also used for the experiments presented in the paper.

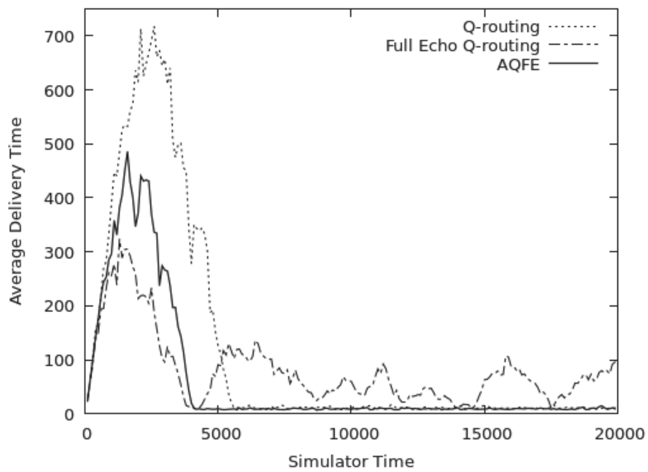

Fig. 2. Q-routing, Full Echo Q-routing, and AQFE under the same load conditions

C. Adaptive Q-routing Full Echo (AQFE)

The problem with oscillations under high load is addressed in [19] by modifying Full Echo Q-routing in the following way. The additional learning rate (η_2) is introduced, and it dynamically adapts to the estimate of the average delivery time obtained at the node. The basic learning rate η is used to update the Q-value of the neighbor to whom the packet is sent. And the additional learning rate is used to update Q-values for other neighbors during full echo procedure. The resulting algorithm is called Adaptive Q-routing Full Echo (AQFE).

The additional learning rate is adapted for each node as follows:

$$\eta_2 = \frac{T_{est}}{T_{\max}} \cdot \eta \cdot k \tag{3}$$

where T_{est} is an estimate of the average delivery time, $T_{\max}$ is an estimate of the maximum average delivery time, k is a parameter called echo rate which determines the ratio between learning rates when the average delivery time is at its maximum. The node x updates its estimate T_{est} as follows:

$$T_{est} = \frac{1}{n_D} \sum_{d \in D} \min_{y \in N(x)} Q_x(d, y) \qquad (4)$$

where D is the set of all destinations known to node x, n_D is the size of D, $N(x)$ is the set of all neighbors of node x. The estimate $T_{\max}$ is the current maximum value among all T_{est} obtained at node x. The estimates T_{est}, $T_{\max}$ allow to dynamically adjust the balance between exploration and exploitation. During the initial stage of the learning process, the average delivery time increases or remains high, therefore $T_{est} \approx T_{\max}$. The echo rate k usually is less than 1, and consequently $\eta_2 < \eta$. This reduces exploration comparing to Full Echo Q-routing, thus making the routing more stable. The adaptation rule shown in (3) allows to explore more, when the average delivery time is high, because stabilization of routing is not the main concern at this stage. For example at the initial stage the level of exploration is comparable to that of Full Echo Q-routing. But the overshoot under AQFE is greater than that of Full Echo Q-routing because $\eta_2 < \eta$ at initial learning stage. Later, η_2 reduces, thus eliminating some oscillations and improving routing stability. For example, Fig. 2 compares Full Echo Q-routing and Adaptive Q-routing Full Echo, and shows that the overshoot is higher but oscillations are eliminated under AQFE.

III. Adaptive Q-routing with random echo and route memory (AQRERM)

AQFE, presented in the previous section, improves the performance of routing in comparison to Full Echo Q-routing, mainly under high load. Nevertheless, it suffers from instability of routing under higher load conditions or in the case of increased k, the parameter used in (3). For example, see Fig. 11 where the application of AQFE leads to frequent spikes in the average delivery time.

In this paper we propose the following modification of AQFE which is based on the following techniques. First, we modify the full echo procedure by requesting the neighbors randomly, i.e. each time only the random subset of the neighbors is requested for the estimates t. Second, each packet has the attachment with the information about the nodes visited by this packet. The proposed algorithm is called Adaptive Q-routing with Random Echo and Route Memory (AQRERM). Below we consider these modifications in more details.

A. *Random Echo*

When the parameter k becomes greater under AQFE, the performance is usually improved at the initial stage of the learning process. But higher values of k may cause instability of routing. The instability may be eliminated by reducing k, but smaller k leads to the increase of the settling time in many cases. That is why, we propose the technique called Random Echo instead of the original Full Echo procedure. According to this technique, each neighbor, that is not the proxy for the packet, is requested for its estimate t with probability equal to $T_{est}/T_{\max}$.

This modification leads to the following remarkable feature. The policy is almost like AQFE when the learning process at the initial stage because $T_{est} \approx T_{\max}$ if the average delivery time increases. But later on, it is almost like Q-routing, because the value $T_{est}/T_{\max}$ gets smaller, and this corresponds to low probability of requesting the neighbors. In this case the learning is mainly provided by (1) and (2) with constant learning rate η, thus making AQRERM similar to Q-routing.

As a result, AQRERM significantly reduces instability and oscillations for high values of k, because the probability of polling is low after the learning has settled, and consequently higher k do not affect the routing decisions so much.

B. *Route Memory*

The second modification of AQFE is called Route Memory and it can be described as follows. Each packet has the list of the visited nodes. This list of limited size L is updated upon the arrival of the packet to the new node. The predefined size of the list implies that the packet can keep the information about no more than L visited nodes. The route is determined by the sequence of visited nodes, thus each packet has the memory of its own route already established, and this gives the name for the method.

The information about visited nodes is used in the decision-making process. The node always chooses the neighbor y with minimum $Q_x(d,y)$ among Q-values that are at row d and not related to the nodes already visited by the packet. If all neighbors have been already visited then this rule is not applied. Also, node y's estimate t for the time remaining in the route is obtained by the formula:

$$t = \min_{z \in N^*(y)} Q_y(d, z) \qquad (5)$$

where $N^*(y)$ is the set of all y's neighbors except node x, or $N^*(y) = N(y)$ when $N(y)$ contains only node x. This rule takes into account that node y gets the estimate t for the packet arrived from node x. In most cases this packet will not return to x because of the Route Memory policy, and therefore $N^*(y)$ is used instead of $N(y)$.

IV. Experimental evaluation

We use simulation modeling for experimental evaluation of routing algorithms. The development of analytical bounds or measures for the performance of the algorithms, based on Q-routing, is a difficult task that can be addressed in future research.

Several results of the experimental evaluation of AQFE for the benchmark network presented in Fig. 1 are given in [19].

Particularly, in most cases AQFE leads to lower overshoot, and the settling time is almost the same as in the case of Q-routing, and Dual Reinforcement Q-routing [10] (DRQ-routing).

But these results are obtained for the settings where $k = 0.01$. Fig. 2 represents the result for AQFE under $\eta \cdot k = 0.22$.

Let us consider the experiments for even higher values of k and address the problem of instability of routing under this high load.

The results described in the paper are for η = 0.9. Notice that this choice of η is suggested by the results of the experiments with learning rates presented in [20]. It has been shown that the values η = 0.9 and η = 1.0 provide the best performance of Q-routing under most load conditions according to the overshoot and the settling time.

The average delivery time is updated with period equal to 100 time steps or “ticks”. The time is measured in ticks because the transition to real time units is straightforward, and the tick unit is more convenient at this stage of performance analysis in its general form when we do not consider many implementation details such as transmission rate or processor clock speed. Following [9], [10] and other works, we use some usual simplifications related to the packet size and the transmission time between nodes. Namely, all packets have the same size and are transmitted in 1 tick between nodes in any neighboring pair.

The irregular grid network shown in Fig. 1 is used in all experiments. Packets are destined for random nodes and originate in the network at random nodes. The creation of packets are driven by Poisson distribution with parameter λ. This parameter also indicates the network load. For example, $\lambda = 1$ refers to conditions of low load, and $\lambda \geq 3$ is considered as high load because, as suggested by our experiments, when $\lambda \approx 3.7$ the network, in most cases, becomes congested, and the routing algorithms are unable to find efficient routes.

A. *Low load conditions*

The results of three independent trials with different random seeds are presented in Fig 3. The settings for these trials were: $\lambda = 1$, $\eta = 0.9$, $\eta \cdot k = 0.5$, $L = 3$. The plots clearly indicate that AQRERM significantly outperforms Q-routing, DRQ-routing and AQFE in all trials. For the trials with other random seeds the results are pretty the same.

B. *High load conditions*

The results of three independent trials with different random seeds are presented in Fig 4. The settings for these trials differ from the previous case only by load level, which is determined here by $\lambda = 3$. The better performance of AQRERM is quite obvious.

C. *Instability under high load conditions*

Under high load conditions, when $\lambda = 3$, wild variations of the average delivery time may occur even after the initial learning has settled. These variations are mostly in the form of high spikes, for example see Fig. 11.

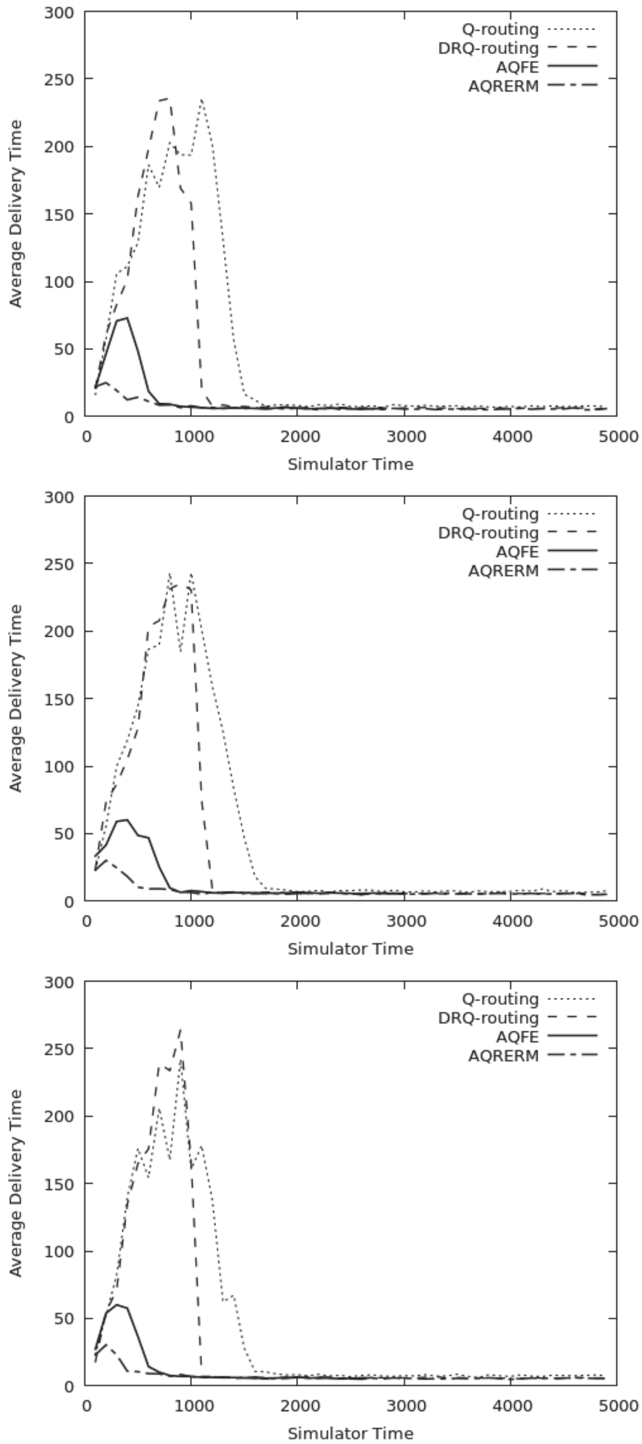

Fig. 3. Performance of routing algorithms under low load conditions

Probably, this effect of the delivery time variations can be explained by the oscillation of routes between the top path and the central one, which connect two parts of the network [9].

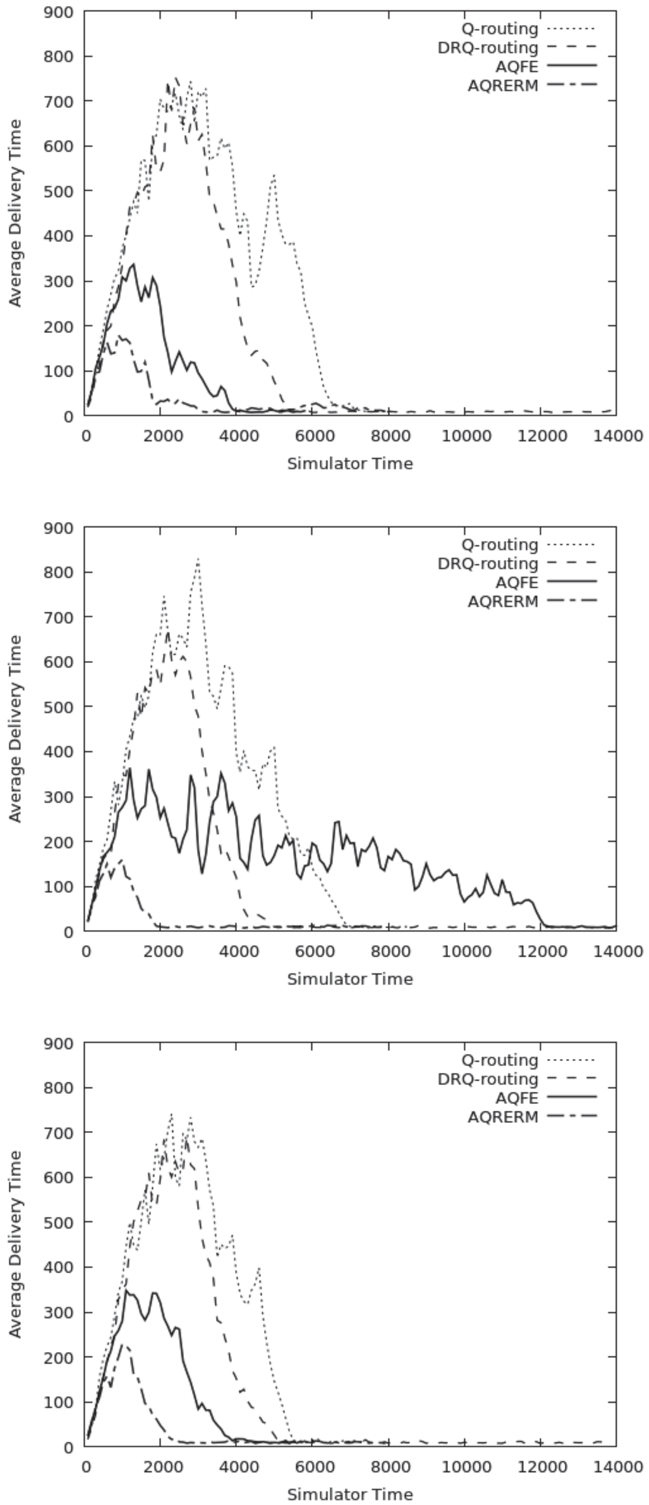

Fig. 4. Performance of routing algorithms under high load conditions

Fig. 5 demonstrates the queues oscillating during the spike of the average delivery time. The lengths of the queues change over short period of time, from tick 10383 to tick 11234. This indicates that routes are switching between the top and central path of the network. This maintains the average delivery time much higher than the lowest possible level.

These pictures, as well as the results of the experiments in this paper, are obtained by using multi-agent modeling environment NetLogo. The routing algorithms are also implemented and tested in this environment.

The main concern is to eliminate the variations of the average delivery time or at least significantly reduce the height of the spikes and the probability of their occurrences.

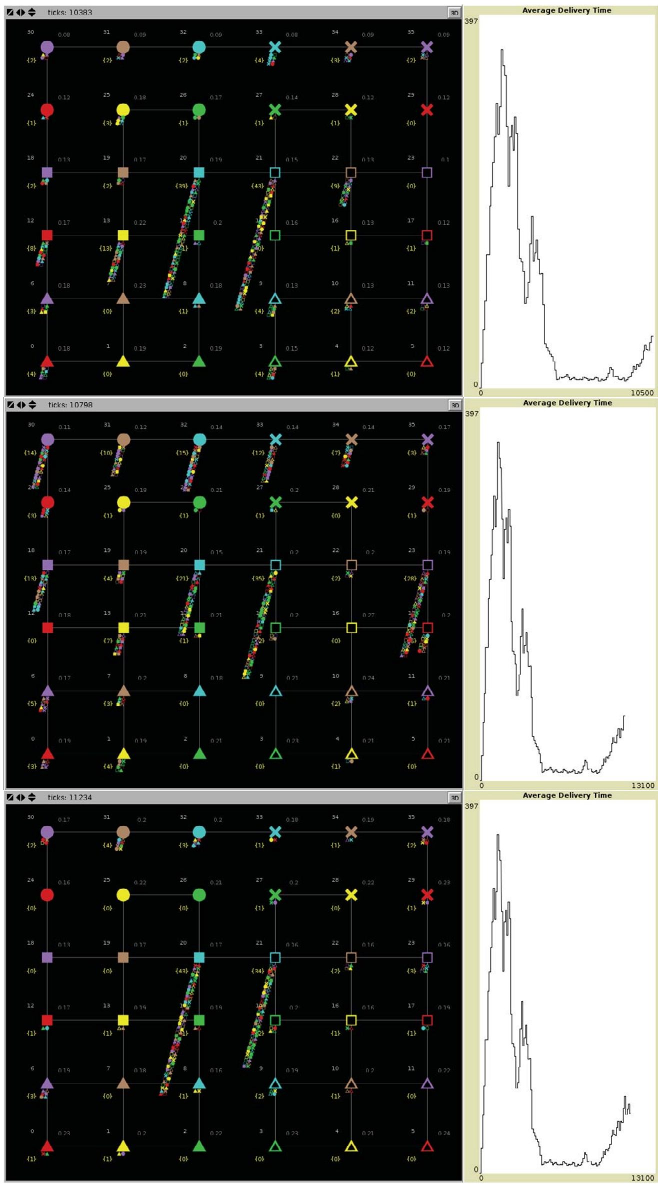

Fig. 5. Oscillating routes lead to oscilating lengths of the queues when AQFE is used under high load

The importance of this issue is supported by the fact that in many cases a high spike can occur after a period of stable routing even when the network load remains constant, as depicted in Fig. 11. For instance, this makes harder to estimate the possibility of high delivery times after initial learning stage and, consequently, guarantee that some predefined timing constraints will be met.

The high spikes can be eliminated by reducing *k*. But the settling time increases when *k* increases. The proposed algorithm, AQRERM, solves this problem by eliminating high spikes under high *k*.

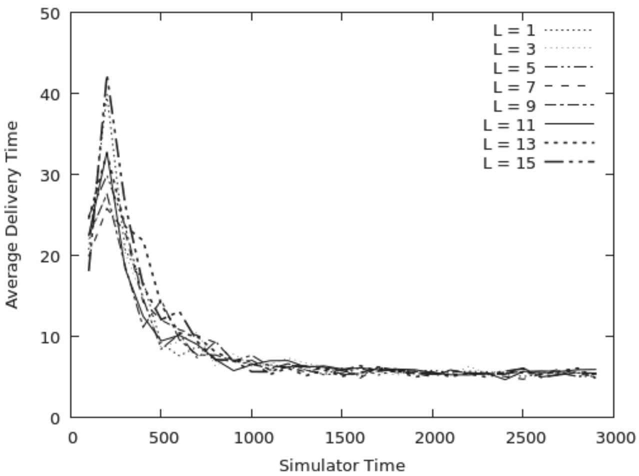

Fig. 6. Performance of AQRERM for various *L* under low load

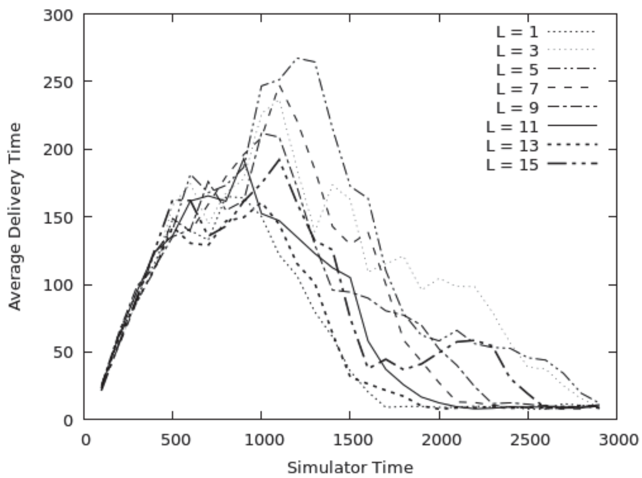

Fig. 7. Performance of AQRERM for various *L* under high load

D. Influence of L on overshoot and settling time

The parameter *L*, the size of the list, affects the decisions of AQRERM, therefore some experiments are needed to estimate the influence of this parameter on the performance and stability of routing. Fig. 6 shows how the performance of AQRERM depends on *L* for $\lambda = 1$, $\eta = 0.9$, $\eta \cdot k = 0.5$. This set of parameters is the same as in the case of the experiment presented in Fig. 3 and can be considered as a condition of low load. These results demonstrate that under conditions of low load there is no much difference among the cases with different *L*.

The performance under varying *L* and $\lambda = 3$, $\eta = 0.9$, $\eta \cdot k = 0.5$ is shown in Fig. 7. Under conditions of high load the effects of various *L* can be seen more clearly.

E. AQRERM with constant k and AQFE with fine-tuned k

Instability of routing under AQFE can be almost eliminated by using lower values of k, and AQFE can provide the stable routing as well as the proposed AQRERM with the same *k*, see Fig. 8, where AQFE has $\eta \cdot k = 0.125$ and AQRERM has $\eta \cdot k = 0.5$ under the same settings: $\lambda = 3$, $\eta = 0.9$, $L = 3$. Notice that AQFE provides the stable routing for a long period. But even in this case, AQRERM performs better in terms of overshoot.

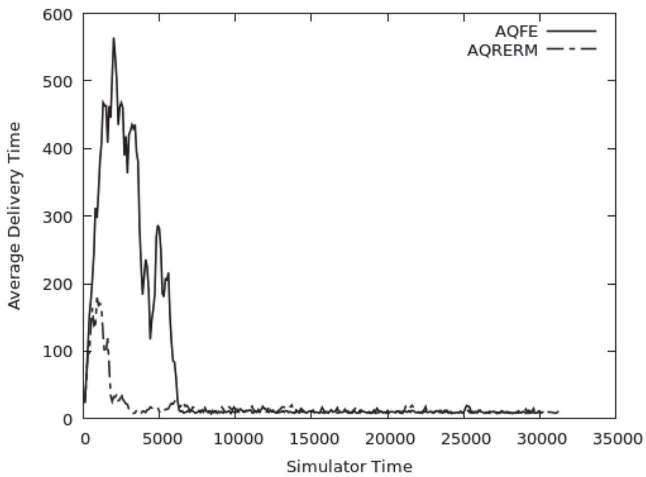

Fig. 8. Performance AQFE with fine-tuned *k* against AQRERM with the same settings

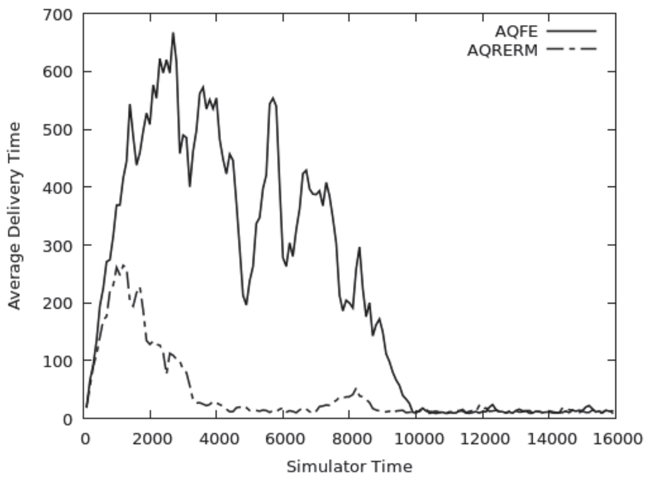

Fig. 9. Performance of routing algorithms under higher load conditons

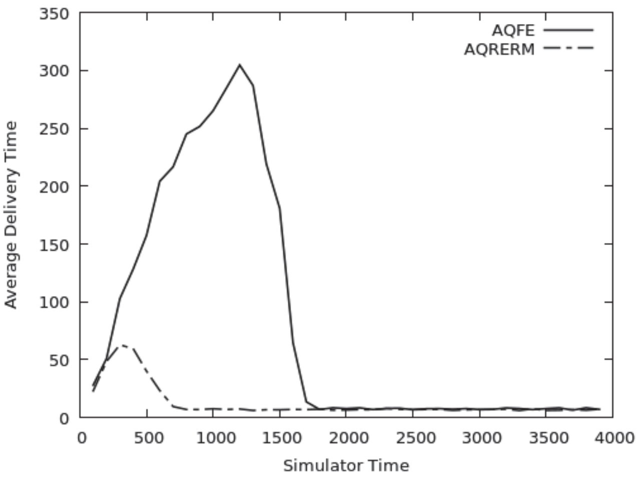

Fig. 10. Performance of routing algorithms under medium load conditons

Under higher load, see Fig. 9, when $\lambda = 3.2$, AQRERM with the same settings outperforms AQFE with the settings previously fine-tuned, namely $\eta \cdot k = 0.125$ .Under medium load, see Fig. 10, when $\lambda = 2$, AQRERM with the same settings also significantly outperforms AQFE with $\eta \cdot k = 0.125$. The results show that under various load conditions, AQRERM with unchanged k outperforms AQFE with k fine-tuned for a specific load range.

The instability of routing can be provided by using AQFE with smaller k but this can significantly increase overshoot and settling time. The tradeoff between performance and stability in the case of AQFE can be achieved by fine-tuning k for each load condition. But in the case of AQRERM one value of k, e.g. such that $\eta \cdot k = 0.5$, provides stability of routing and gives significantly better performance for all load conditions in comparison to AQFE with fine-tuned k. Thus AQRERM provides more convenient and efficient way to route packets without adapting k for each load condition.

F. *Influence of L on stability*

We estimate stability of routing under high load, $\lambda = 3$, on larger time intervals such as 80000 ticks as depicted in Fig. 11. In the case of AQFE, high spikes of the average delivery time occur at regular intervals during routing. AQRERM provides stability of routing under high load conditions by eliminating high spikes of delivery time variations.

Notice that under higher load conditions, e.g., when $\lambda = 3.2$, small variations of the average delivery time may occur even if AQRERM is applied, for example, see Fig. 9. To test this in more details, similar experiments were carried out for even higher load when $\lambda = 3.5$, and the results are presented in Fig. 12. Notice that AQFE is unable to find efficient routes in this case and the learning has not settled, while under AQRERM the learning has settled but with larger settling time and overshoot. The routing is stable under AQRERM in this experiment.

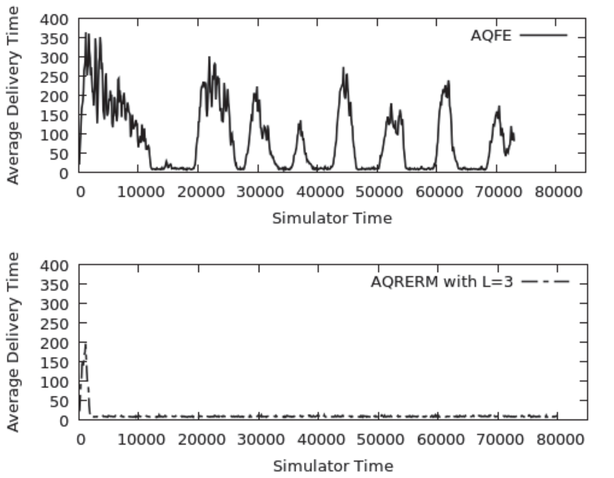

Fig. 11. Stability of routing under AQFE and AQRERM for $\lambda = 3$

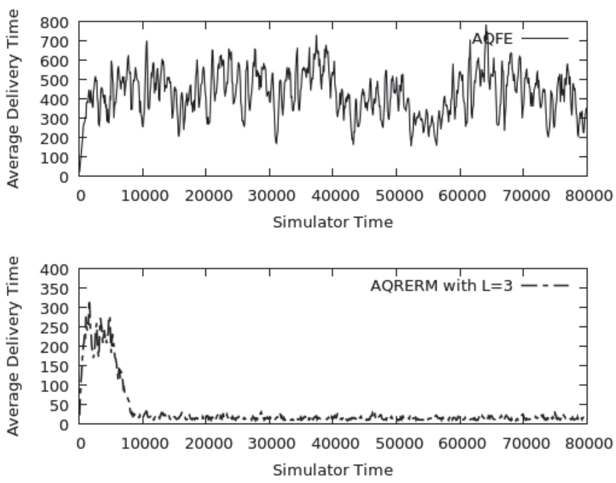

Fig. 12. Stability of routing under AQFE and AQRERM for $\lambda = 3.5$

V. Conclusion

A modification of the routing algorithm AQFE [15] is proposed and called Adaptive Q-routing with Random Echo and Route Memory (AQRERM). The algorithm outperforms AQFE, Q-routing [9] and DRQ-routing [6] under various load conditions for the irregular 6x6 grid network widely used as a benchmark.

AQRERM does not require finding the best settings, e.g., the value of the parameter k, for specific load conditions. This can be required in the case of AQFE.

AQRERM greatly improves stability of routing under high load conditions comparing to AQFE. This justifies the proposed modifications of AQFE and the practical value of the resulting algorithm called AQRERM.

The results of the experiments obtained for the irregular 6x6 grid network, see Fig. 1, support the claims made above.

Additional experiments are needed to test AQRERM under various conditions. For example, other network topologies can be used, including the networks with dynamically changing topologies. Also, the performance of AQRERM under different load patterns can be tested.

REFERENCES

[1] R.A. Haraty, and B Traboulsi, "MANET with the Q-Routing Protocol," ICN The Eleventh International Conference on Networks. 2012.

[2] B. Russell, "Learning-based route management in wireless ad hoc networks", PhD Thesis, New Brunswick Rutgers, The State University of New Jersey, 2008.

[3] A. Hinds, M. Ngulube, S. Zhu, and H. Al-Aqrabi, “A review of routing protocols for mobile ad-hoc networks (manet)”. International journal of information and education technology, 3(1), 2013.

[4] C. E. Perkins and E. M. Royer, “Ad-hoc On-Demand Distance Vector Routing,” Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, Feb. 1999, pp. 90–100.

[5] M. Elzohbi, “Flexible and Scalable Routing Approach for Mobile Ad Hoc Networks by Function Approximation of Q-Learning”. Master Thesis, University of Calgary, 2016.

[6] C. J. Watkins and P. Dayan, "Q-learning," Machine learning 8.3-4, 1992, pp. 279–292.

[7] R.S. Sutton and A.G. Barto, Reinforcement learning: An introduction. Cambridge: MIT press, 1998.

[8] S. Chettibi and S. Chikhi. "A Survey of Reinforcement Learning Based Routing Protocols for Mobile Ad-Hoc Networks." Recent Trends in Wireless and Mobile Networks. Springer Berlin Heidelberg, 2011. pp 1-13.

[9] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," Advances in neural information processing systems, 1994, pp. 671–671.

[10] S. Kumar and R. Miikkulainen, "Dual Reinforcement Q-Routing: An On-Line Adaptive Routing Algorithm," Artificial neural networks in engineering, 1997.

[11] S. Choi and Dit-Yan Yeung, "Predictive Q-routing: A memory-based reinforcement learning approach to adaptive traffic control." Advances in Neural Information Processing Systems 8, 1996.

[12] D. Subramanian, P. Druschel, and J. Chen. "Ants and reinforcement learning: A case study in routing in dynamic networks," IJCAI (2), 1997.

[13] L. Peshkin and V. Savova "Reinforcement learning for adaptive routing," Proceedings of the 2002 International Joint Conference on Neural Networks, Vol. 2, 2002.

[14] S. Hoceini, A. Mellouk, and Y. Amirat, "K-shortest paths Q-routing: a new QoS routing algorithm in telecommunication networks," Networking-ICN 2005, pp. 164–172.

[15] N. Gupta, M. Kumar, A. Sharma, M. S. Gaur, V. Laxmi, M. Daneshtalab, and M. Ebrahimi, "Improved route selection approaches using Q-learning framework for 2S NoCs," ACM 3rd International Workshop on Many-core Embedded Systems, 2015, pp. 33-40.

[16] R. Arroyo-Valles, R. Alaiz-Rodríguez, A. Guerrero-Curieses, and J. Cid-Sueiro, "Q-Probabilistic Routing In Wireless Sensor Networks," IEEE 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, 2007.

[17] A. M. Lopez and D. R. Heisterkamp, "Simulated annealing based hierarchical Q-routing: a dynamic routing protocol," IEEE Eighth International Conference on Information Technology: New Generations, 2011, pp. 791–796.

[18] S. T. Yap and M. Othman. "An adaptive routing algorithm: enhanced confidence-based Q-routing algorithm in network traffic." Malaysian Journal of Science 17.2, 2004, pp 21-29.

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

[20] Y. Shilova and M. Kavalerov, "Issledovanie vliyaniya parametra skorosti obucheniya na rezultaty raboty algoritma marshrutizacii Q-routing [Study of influence of the learning rate parameter on the results of algorithm Q-routing]", Innovacionnye tehnologii: teoriya, instrumenty, praktika, 2015, pp. 172–179.