Journal of High Speed Networks 29 (2023) 1–14
DOI 10.3233/JHS-222018
IOS Press

Double Q-learning based routing protocol for opportunistic networks

Jagdeep Singh [a,*], Sanjay Kumar Dhurandher [b], Isaac Woungang [c] and Leonard Barolli [d]

[a] *Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, India*
E-mail: jagdeepknit@gmail.com
[b] *Department of Information Technology, Netaji Subhas University of Technology, New Delhi, India*
[c] *Department of Computer Science, Toronto Metropolitan University, Toronto, Ontario, Canada*
[d] *Department of Information and Communication Engineering, Faculty of Information Engineering, Fukuoka Institute of Technology, Fukuoka, Japan*

Received 26 July 2022
Accepted 28 September 2022

Abstract. Opportunistic Delay Tolerant Networks also referred to as Opportunistic Networks (OppNets) are a subset of wireless networks having mobile nodes with discontinuous opportunistic connections. As such, developing a performant routing protocol in such an environment remains a challenge. Most research in the literature have shown that reinforcement learning-based routing algorithms can achieve a good routing performance, but these algorithms suffer from under-estimations and/or over-estimations. Toward addressing these shortcomings, in this paper, a Double Q-learning based routing protocol for Opportunistic Networks framework named Off-Policy Reinforcement-based Adaptive Learning (ORAL) is proposed, which selects the most suitable next-hop node to transmit the message toward its destination without any bias by using a weighted double Q-estimator. In the next-hop selection process, a probability-based reward mechanism is involved, which considers the node's delivery probability and the frequency of encounters among the nodes to boost the protocol's efficiency. Simulation results convey that the proposed ORAL protocol improves the message delivery ratio by maintaining a trade-off between underestimation and overestimation. Simulations are conducted using the HAGGLE INFOCOM 2006 real mobility data trace and synthetic model, showing that when time-to-live is varied, (1) the proposed ORAL scheme outperforms DQLR by 14.05%, 9.4%, 5.81% respectively in terms of delivery probability, overhead ratio and average delay; (2) it also outperforms RLPRoPHET by 16.17%, 9.2%, 6.85%, respectively in terms of delivery ratio, overhead ratio and average delay.

Keywords: Reinforcement learning, double Q-learning, opportunistic networks, routing, real mobility data

1. Introduction

Opportunistic Networks [10] are a subset of wireless networks consisting of mobile nodes having opportunistic connections that are discontinuous. One of the main drawbacks of OppNets is the lack of an end-to-end connection between the source node and the destination node, which arises due to the mobility of nodes and the possibility of intermittent connectivity. As a result of this, the store-carry-and-forward mechanism used in delay tolerant networks (DTNs) and the contact opportunity or mobility assistance among the nodes are often used for message forwarding purposes in the network [4]. The challenges encountered when designing a routing protocol for OppNets include high consumption of resources, long queueing time in the buffers of nodes, low data rate, poor delivery ratio, and high latency. Two well-known routing strategies for OppNets are forwarding-based [13] and

[*] Corresponding author. E-mail: jagdeepknit@gmail.com.

replication-based [19]. However, because the resources of the network are restricted, the challenge is to limit the number of copies of the messages on the network. These schemes do not handle changes in the topology of the network and the mobility of nodes. To address these issues, reinforcement learning-based routing protocols are utilized. Reinforcement learning (RL) uses a trial-and-error technique of learning the environmental attributes and it can be employed with the use of the Markov decision process [2]. Using this paradigm, a node attempts to gather information from its neighbors, for example, the buffer occupancy, number of hops from a particular destination, energy level, to name a few. The best possible next hop of a node is the node with the highest collected reward.

Q-learning [17] is an off-policy reinforcement learning algorithm that tells an agent what action to take under what conditions in order to learn the quality of the actions. It does not require an environmental model, and without requiring adaptation, it can handle issues with the stochastic transitions and rewards. In the routing, Q-learning is applied to predict the best possible next node that can be used to forward the message. However, this approach may lead to the issue of overestimation since it uses only the maximum action value for the selection or analysis of a particular node, which may result in higher than realistic expectations. Hence, to obtain unbiased results while improving the network performance, it has been advocated to use the double Q-learning technique.

In the double Q-learning approach [15], all nodes acquire the information about the network's connectivity and mobility patterns in a distributed fashion. To increase the accuracy in the next hop selection to forward the message, a dynamic reward mechanism is proposed, which is responsible for evaluating the dynamic changes occurring in the network information. The double Q-learning approach is implemented using two value functions. The first one uses the greedy strategy to establish a reward between two nodes and the second one is meant to approximate the value of this reward. To compute the future rewards of an intermediate node, an intermediate value mechanism is implemented. As the number of hops to transmit a message between two nodes increases, the reward decreases. Thus, the value of the reward between two nodes is calculated as a function of the number of hops that a message requires to reach its destination through that node. On the other hand, the reward and the number of hops are inversely proportional to each other. This approach can result in negative maximization bias when the action values are learned, leading to underestimation. In this paper, an off-policy Q-learning algorithm, which depends on a double-weighted estimator, is proposed to address this deficiency, the goal being to maintain an equilibrium between the underestimation in the double estimator and the overestimation in the single estimator.

The remainder of the paper is organized in the following way. The related work is included in Section 2. The Off-Policy Reinforcement-based Adaptive Learning for Routing in OppNets is explained in Section 3. Simulation findings are illustrated in Section 4. The conclusion of the paper is drawn in Section 5.

2. Related work

Few OppNets routing protocols utilizing RL techniques have been investigated in this section.

In [18] Wu et al. integrated the DTN mechanisms with the On-Demand Ad hoc Distance Vector protocol (AODV) using a Q-Learning based protocol. A simple reward function is utilized, and Q-Learning is employed to fetch the network's link status information. To verify the authenticity of the information obtained on the routes at regular intervals, a mechanism for route request/reply is proposed, which helps to initiate quick responses to network topology changes.

In [5] Elwhishi et al. employed a cost function between nodes to make the forwarding decisions between cooperative groups of nodes by proposing a RL-based routing scheme for DTNs. In this protocol, a cost function is used for reward calculation, and the factors determining the value of this cost function are buffer occupancy, congestion, and node mobility statistics.

In [16] Hasselt et al. proposed a deep RL with a double Q-learning scheme for DTNs (so-called DQN), which addresses the overestimations problem encountered in the Q-learning approach. The proposed algorithm utilizes a combination of Q-learning and deep neural networks approach to overcome this problem. The authors demonstrated that the double Q-learning methods have relevance in large-scale function approximations. Taking this work further, Hesselt *et al.* proposed an enhanced DQN algorithm Hasselt et al. [6], in which the over-optimistic behavior of Q-learning methods are observed in large-scale scenarios and the double Q-learning method is used to tackle the problem of overestimation. In [20] Yuan et al. introduced a method to balance the overestimation problem due to the Q-learning algorithm by using a double Q-learning based routing protocol for DTNs. In their scheme, each node learns the mobility patterns and link-state information in a distributed manner, which helps in avoiding overestimation by optimally selecting the next hop.

In [7] Hu et al. attempted to utilize the generic MAC protocols to increase the lifetime of the network by evenly distributing the residual energy of the nodes. This was achieved by proposing a RL-based adaptive routing protocol (called QELAR) for underwater sensory networks. In their scheme, the forwarders of the packets are selected using a reward function which involves the node residual energy and energy distribution among a group of selected nodes. In [9] Nelson et al. proposed a routing protocol for DTNs based on a quota function which limits the consumption of resources by imposing an upper bound over the number of messages created by the node. Indeed, let X denotes the upper bound, whenever a new message is generated, its source initiates the so-called spray phase, where X number of copies are delivered to other nodes present in the network.

In [3] Balasubramanian et al. introduced a resource allocation mechanism for the intentional DTNs, which specifically optimizes an administrator-specific routing metric. In their scheme, a packet is routed by duplicating it opportunistically until the target node receives a copy of it. In this process, the routing metric is translated to per-packet utilities, which are meant to check whether replicating a packet is worth or not depending on the resources used. In doing so, these network resources are tracked by means of a dedicated control plane.

In [12] Rolla et al. introduced an infrastructure-less routing protocol designed for wireless networks, in which multi-agent RL techniques are used in the route selection process as well as in the message forwarding process.

In [14] Sharma et al. introduced a dynamic programming based reinforcement learning approach for optimal routing in opportunistic IoT networks and has solved the MDP using policy iteration to get the optimal policy which is used to design the routing protocol.

The Double Q-Learning algorithm is employed to improve the performance of networks and acquire an unbiased estimation by decoupling the selection from the evaluation. DQLR protocol [20] helps each node learn information regarding link states along with movement patterns from the network. This is done in a distributed fashion. Double Q-Learning Routing protocol uses two value functions to decouple selections from the evaluation. Here, the greedy strategy is determined by one of the value functions whereas the other function is used to determine its value. However, the action values can sometimes be underestimated by DQLR. This paper introduces a Off-Policy Reinforcement-based Adaptive Learning (ORAL) algorithm, which is based on the construction of the weighted double estimator. The primary objective is to maintain a balance among the underestimation in the double estimator and the overestimation in the single estimator.

3. System model

In this section, we first present the Off-Policy Reinforcement-based Adaptive Learning model and its unbiased balanced behavior. Next, the reward mechanism and the method used to calculate the future reward used in the suitable next-hop selection to carry the message forward to destination are presented. The description of different blocks used in the ORAL architecture diagram are as follows:

Source Node: The node which wants to transmit the data. The number of neighbors to which the message is forwarded by the sender is calculated using a formula that takes into account the probability of delivering the message to the destination.

Destination Node: When an intermediate node finds the destination, the message is delivered to it and the process stops.

Forwarder/Relay Node: The forwarding of message to a certain node during its journey in the network is determined by the match between the sender information and the context information of the nodes. Matches are assigned certain weights representing the precision with which that attribute (belonging to a class) can identify the destination.

ORAL Router: In the Off-Policy Reinforcement-based Adaptive Learning approach, all nodes acquire the information about the network's connectivity and mobility patterns in a distributed fashion. The ORAL approach is implemented using two value functions. The first one uses the greedy strategy to establish a reward between two nodes and the second one is meant to approximate the value of this reward. To compute the future rewards of an intermediate node, an intermediate value mechanism is implemented.

Summary Vectors: The proposed protocol aims at effectively utilizing the context information of the node in order to decrease the overhead of flooding. The Current Context of a user is a snapshot of the local environment of the user. It is stored in the form of Identity Tables. The user has complete autonomy on the information shared in the identity tables.

Q-Learning offers a framework by which a network (in our case, OppNet) can learn how to execute a specific task, such as transmitting the packets from source to destination through an optimal route based on its experience. For this reason, an OppNet can be considered as an agent environment, where messages are to be transmitted from a source to a destination. All mobile nodes present in the network which are participating in the routing process can be considered as states, and the transmissions among these states can be considered as actions. But, simply adopting Q-learning as a basis for the routing protocol does not provide an accurate estimation of future rewards and may lead to an overestimation of the action values. To overcome this issue, Yuan *et al.* proposed that the double Q-learning approach relies on a double estimator method. But, this tends to underestimate the discovered routing paths.

The Q-learning model can be improved by introducing a balance between the overestimation caused by the Q-learning approach and the underestimation caused by the double Q-learning approach, which makes an estimator truly unbiased. Therefore, the weighted double Q-learning algorithm proposed in Zhang et al. [21] is used in the design of our proposed ORAL protocol. Similar to the double estimator method, the weighted double estimator method also uses two value functions. Whenever a transmission of a packet occurs between two nodes, one of these value functions, selected at random, is utilized for updating the values of the future node's rewards. But, instead of using only one function, say Q^X for the greedy strategy and one function, say Q^Y, for the reward update purpose, a linear combination of Q^X and Q^Y is used to calculate a balancing factor β, which is then used to define a trade-off function Q^{Wx} or Q^{Wy} that updates the values in Q^X or Q^Y depending on the selected value function. More precisely, in the greedy strategy, the selection of the next hop is done on the basis of calcualted Q-value. Based on these, we propose the update rules in Q^{Wx} and Q^{Wy} as per Eq. (1) and Eq. (2) respectively. If Q^X is selected as value function, the following equations are involved in the evaluation:

$$Q^{Wx} = \beta^X Q^X\left(i, a_X^*\right) + \left(1 - \beta^X\right) Q^Y\left(i, a_X^*\right)) \tag{1}$$

$$Q_C^X(C, a) = Q^X(C, a) + \alpha(R(C, a) + \gamma\left(Q^{Wx}\right) \tag{2}$$

Otherwise, if Q^Y is selected as value function, the following equations are involved in the evaluation:

$$Q^{Wy} = \beta^Y Q^Y\left(i, a_Y^*\right) + \left(1 - \beta^Y\right) Q^X\left(i, a_Y^*\right)) \tag{3}$$

$$Q_C^Y(C, a) = Q^Y(C, a) + \alpha(R(C, a) + \gamma\left(Q^{Wy}\right) \tag{4}$$

where a_X^* and a_Y^* are the actions with the maximum future rewards from state i, a_X^L and a_Y^L are the actions with the minimum future rewards from state i respectively, where i is the chosen next-hop from the current state C. Here, the value of C is greater than 0 and the tuning factor $\beta \in [0, 1]$, meaning that the double-weighted estimator will behave like a single estimator if $\beta = 1$ and like a double estimator if $\beta = 0$. Therefore, we can observe that an optimal selection of C results in an unbiased estimation of future rewards.

3.1. Reward mechanism, value aging, and transitivity

The reward function determines the performance and behavior of the agent. Therefore, it is a crucial part of the process of estimating the future rewards for all state-action pairs. In this paper, we have used a probabilistic approach to calculate the reward for all state-action pairs in which the delivery probability is calculated at every node for each destination. This probability information is stored at each node, then exchanged when the nodes encountered each other. These internal probabilities are updated using the information received from the neighbor nodes. To formulate the reward function, let's assume that s is the current state of an agent and this agent chooses an action 'a' to reach a state C, then the probability with which the agent can reach the final destination 'd' through i as next-hop will determine the reward R that the agent will get from the state-action pair (C, a).

Then, the reward $R_C(d, i)$ that the sender node C receives when the receiver d is reached through the next hop i can be defined as: $R_C(d, i) = P(i, d)$ where $P(i, d)$ is obtained as:

$$P(i, d) = P(i, d)_{\text{old}} + \bigl(1 - P(i, d)_{\text{old}}\bigr) \times P_{\text{init}} \tag{5}$$

It should be noted that $P(i, d)$ is modified whenever a node is visited, so that frequently visited nodes have a high predictability of delivery. Here, $P_{\text{init}} \in [0, 1]$ is a constant. In case a pair of nodes have not exchanged their information for a while, their predictability will decrease, and hence, the values of the expected reward functions Q^X and Q^Y will also decrease. This mechanism is referred to as aging, and the Eq. (6) and Eq. (7) are referred to as aging equations, where $\gamma \in [0, 1]$ is the aging constant.

$$Q_C^X(d, i) = Q_C^X(d, i)_{\text{old}} \times \gamma^{\mu} \tag{6}$$

$$Q_C^Y(d, i) = Q_C^Y(d, i)_{\text{old}} \times \gamma^{\mu} \tag{7}$$

Another factor that affects the delivery predictability of a node is the transitivity property. According to this, if a neighbor, say i, of a node C is frequently visited by a node j, there is a probability that j is a good forwarder for C. Therefore, it is important to incorporate this factor of transitivity into the Q^W function. This relationship is shown in Eq. (8), where $\beta \in [0, 1]$ is a scaling constant that defines the effect of transitivity on the ORAL protocol.

$$Q_C^W(d, j) = Q_C^W(d, j)_{\text{old}} + \bigl(1 - Q_C^W(d, j)_{\text{old}}\bigr) \times Q_C^W(d, i) \times Q_i^W(d, j) \times \beta \tag{8}$$

The network architecture of the proposed ORAL scheme is depicted in Fig. 1, where the transmission of the data packets through the nodes are illustrated. As shown, the data received by the convergence layer is forwarded to the ORAL router. In this process, the delivery predictability and summary vector databases are used to calculate the reward function and other factors involved in the routing procedure. Next, the convergence layer out-ducts are used to forward the data packet to the most suitable next hop until it reaches its intended destination. Here, the next-hop node is expected to return the maximum reward in the future from the source S to reach the destination D. The pseudo-code of the ORAL protocol is given in Algorithm 1.

3.2. *Illustrative example*

The following example is used to illustrate the working of the proposed ORAL algorithm. Nodes are represented by circles and connections are represented by the edges. First, the Q^X and Q^Y values and the delivery probability of all state-action pairs are initialized to 0. Now, let's assume that the agent randomly chooses an action a from C to i and decides to update Q^X, which is also a random selection. The values of hyperparameters for routing are represented in the Table 1.

We have

$$Q^X\bigl(i, a_X^*\bigr) = Q^Y\bigl(i, a_Y^*\bigr) = Q^X\bigl(i, a_X^L\bigr) = Q^Y\bigl(i, a_Y^L\bigr) = 0 \tag{9}$$

Thus

$$\beta = \frac{|Q^Y(i, a_Y^*) - Q^Y(i, a_Y^L)|}{|c + Q^Y(i, a_Y^*) - Q^Y(i, a_Y^L)|} = 0 \tag{10}$$

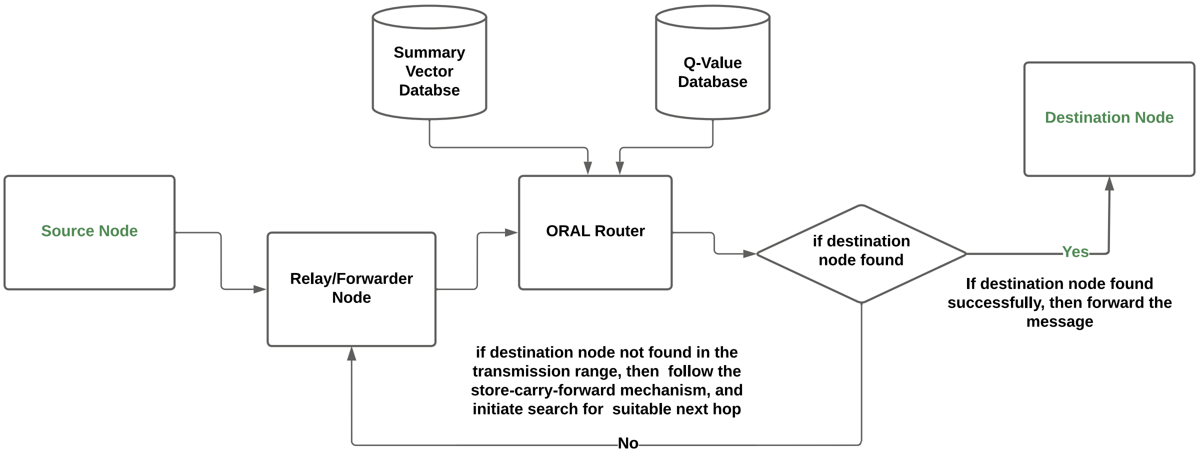
Summary Vector Databse
Q-Value Database
Destination Node
Source Node
ORAL Router
Relay/Forwarder Node
if destination node found
Yes
If destination node found successfully, then forward the message
if destination node not found in the transmission range, then follow the store-carry-forward mechanism, and initiate search for suitable next hop
No

Fig. 1. Architecture of off-policy reinforcement-based adaptive learning.

Table 1
Values of the hyper-parameters

Hyper-parameters	Values
Learning rate	0.9
Value aging factor	0.13
Scaling factor	0.1
P_{init}	0.75
Discount factor	0.9
c	10

and

$$R(C,i) = P(C,i) = P(C,i)_{\text{old}} + \big(1 - P(C,i)_{\text{old}}\big)P_{\text{init}} = 0.75 \tag{11}$$

Using these values, we get:

$$Q^X(C,i) = Q^X(C,i) + \alpha\big(R(C,i) + \gamma\big[\beta^X Q^X\big(i,a_X^*\big) + \big(1-\beta^X\big)Q^Y\big(i,a_Y^*\big)\big] - Q^X(C,i)\big) \tag{12}$$

$$Q^X(C,i) = 0 + 0.9\big(0.75 + 0.9 * (0) - 0\big) = 0.9 * 0.75 = 0.675 \tag{13}$$

Now, assume that the agent chooses this path again from C to i and decides to update Q^X again. This time, we will get more rewards from this path because it has been visited before by the agent, i.e.

$$R(C,i) = P(C,i) = P(C,i)_{\text{old}} + \big(1 - P(C,i)_{\text{old}}\big)P_{\text{init}} \tag{14}$$

$$R(C,i) = 0.75 + (1 - 0.75)0.75 = 0.9375 \tag{15}$$

Thus,

$$Q^X\big(i,a_X^*\big) = Q^Y\big(i,a_Y^*\big) = Q^X\big(i,a_X^L\big) = Q^X\big(i,a_X^L\big) = 0 \tag{16}$$

Algorithm 1 Off-policy reinforcement-based adaptive learning

```
1:  **Begin** the source node creates the message to destination.
2:  **for** (all neighbors $i$ in $S_C$) **do**
3:      $S_C$ starts neighbor discovery.
4:      Update the delivery probability and calculate the reward for the same.
5:      Randomly select $Q_C^X$ or $Q_C^Y$.
6:      **if** ($Q_C^X$ is selected) **then**
7:          Update the estimated reward using Eqs (1)–(2).
8:      **else**
9:          **if** ($Q_C^Y$ is selected) **then**
10:             Update the estimated reward using Eqs (3)–(4).
11:         **end if**
12:     **end if**
13:     Calculate $\beta^X$ or $\beta^Y$.
14:     **if** ($Q_C^X$ is selected) **then**
15:         $$\beta^X = \frac{|Q_C^Y(i,a_Y^*) - Q_C^Y(i,a_Y^L)|}{|c + Q_C^Y(i,a_Y^*) - Q_C^Y(i,a_Y^L)|}$$
16:     **else**
17:         **if** ($Q_C^Y$ is selected) **then**
18:             $$\beta^Y = \frac{|Q_C^X(i,a_X^*) - Q_C^X(i,a_X^L)|}{|c + Q_C^X(i,a_X^*) - Q_C^X(i,a_X^L)|}$$
19:         **end if**
20:     **end if**
21:     Repeat Step 3 to 20 for explorations and calculations to get a complete set of delivery probabilities and Q-values for all state-action pairs, which can then be used to estimate the expected rewards for all paths available at any state.
22:     Calculate $Q_C^W$ for all paths obtained in the previous Step.
23:     **if** ($Q_C^X$ is selected) **then**
24:         $$Q_C^W = \beta^X Q^X(i,a_X^*) + (1-\beta^X) Q^Y(i,a_Y^*))$$
25:     **else**
26:         **if** ($Q_C^Y$ is selected) **then**
27:             $$Q_C^W = \beta^Y Q^Y(i,a_Y^*) + (1-\beta^Y) Q^X(i,a_X^*))$$
28:         **end if**
29:     **end if**
30: **end for**
31: Node with the highest value of $QC^W$ is selected.
32: **if** ($i$ is not receiver) **then**
33:     Create: duplicate the message copy and transfer it to node $i$.
34:     Add the transitive factor if possible.
35:     Perform routing on $i$ for further transmission.
36: **else**
37:     Transfer the message to node $i$ successfully.
38: **end if**
```

and

$$\beta = \frac{|Q^Y(i,a_Y^*) - Q^Y(i,a_Y^L)|}{|c + Q^Y(i,a_Y^*) - Q^Y(i,a_Y^L)|} = 0 \tag{17}$$

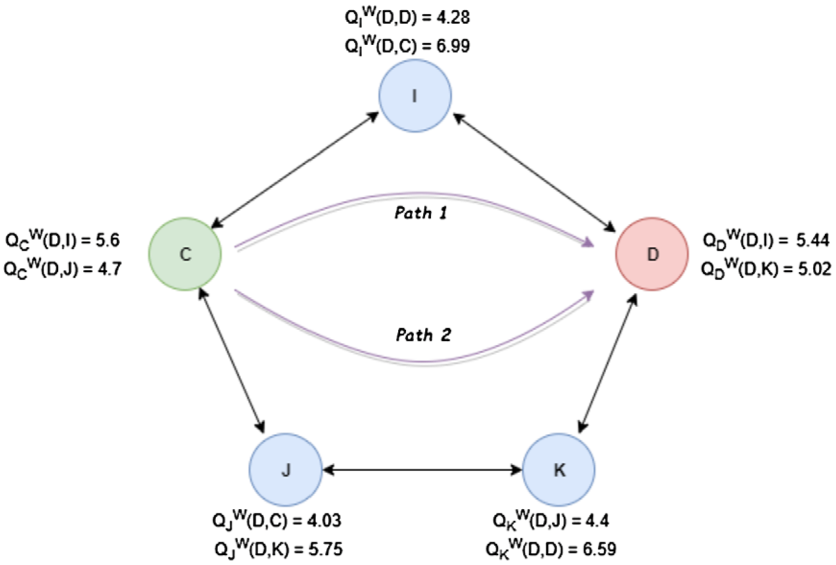

Fig. 2. Illustrative example of ORAL protocol.

Using the values in these equations, we get:

$$Q^X(C,i) = Q^X(C,i) + \alpha\big(R(C,i) + \gamma\big[\beta^X Q^X\big(i,a_X^*\big) + \big(1-\beta^X\big)Q^Y\big(i,a_Y^*\big)\big] - Q^X(C,i)\big) \tag{18}$$

$$Q^X(C,i) = 0.675 + 0.9\big(0.9375 + 0.9(0) - 0.675\big) = 0.9 * 0.75 = 0.91125 \tag{19}$$

After a few such explorations and calculations, we will get a complete set of delivery probabilities and Q-values for all pairs of state-action, which can then be used to estimate the expected rewards for all paths available at any state, and to decide which one to choose for message delivery to destination. While comparing the expected rewards from the available number of paths, a Q^W function is invoked as a function of Q^X and Q^Y. In Fig. 2, $Q_C^W(D,i)$ represents the estimated expected reward if an agent moves from C towards D using i as next-hop node.

Now, let's suppose that a message is to be sent from nodes C to D, then according to Fig. 2, it can be concluded that it will be better for the agent to choose I as next-hop node from C because it is expected to generate more reward value since $Q_C^W(D,i) > Q_C^W(D,j)$.

4. Simulations

The proposed ORAL protocol is simulated using the ONE simulator version 1.6.0 [8]. The capabilities of the proposed protocol are compared with that of four recent benchmark routing protocols for OppNets, namely, the Double Q-Learning Routing (DQLR) [20], QLR, RLPRoPHET [14] and Epidemic [9]. The performance metrics considered for analysis are delivery ratio, overhead ratio and average delay. The simulation parameters are given in Table 2.

- *Delivery Ratio*: Delivery ratio is defined as the ratio of the number of successfully delivered messages to that of the total number of messages being generated in a network. High values of delivery probability prove high levels of reliability of the routing protocol. Figures 4 and 5 show the delivery ratio vs learning rate and discount rate respectively.
- *Overhead Ratio*: Overhead ratio is defined as the number of duplicate copies of every message being buffered by all the devices of the network. A lower overhead ratio is indicative of better utilization of network storage space.

Table 2

Simulation parameters

Parameter	Synthetic Model	Real data Model
Movement model	SPMBM	Infocom2006 [1]
Simulation time	43200 s	259200 s
Dimension	4500 × 3200 m^2	10000 × 8000 m^2
TTL	50–300 Min.	60–2880 Min.
Number of nodes	80	98
Buffer space	5 Mbytes	10 Mbytes
Velocity of nodes	0.5, 1.5 m/s	0.5, 1.5 m/s
Transmission Range	10 m	10 m
Size of Message	1 Kbytes	25 Kbytes

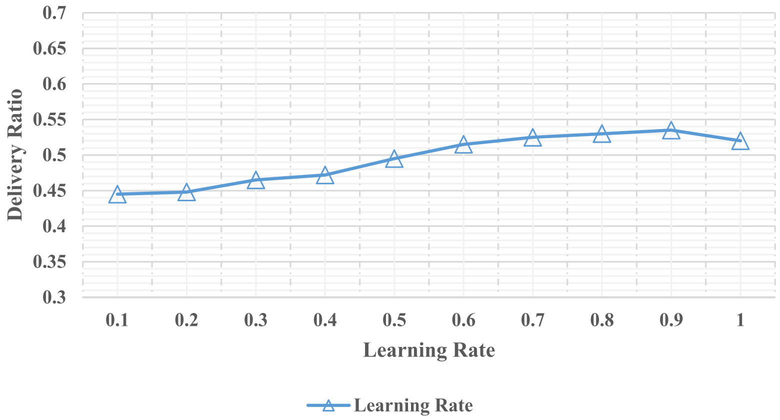

Fig. 3. Delivery ratio vs learning rate (δ).

- *Average Delay*: The mean latency in transmission of message to destination is called average delay.

4.1. Results

The value of TTL is initialized to 100 minutes. As variables, the learning rate δ and discount factor η are taken, and the effect of this variation is evaluated on the delivery ratio. Figure 3 demonstrates that if the agent is only considering the previous learning information, δ becomes zero. It is found that with an increase in the learning rate, the amount of new learning information considered by the agent increases. Additionally, the agent only considers the new learning information as the learning rate tends to 1.

The value of η is small, the future reward value is small. This results in a reduced value of delivery ratio. Additionally, the delivery ratio increases with an increase in the discount factor. Further, the proposed ORAL scheme is compared against the DQLR and QLR schemes with respect to the overhead ratio, the delivery ratio and the delay as the TTL is altered. The TTL is initialized by the source node. This value is lessened after every node encounter. The message is discarded when the TTL reaches 0. Here, the overhead ratio is defined as the average per message number of copies that are forwarded. In addition, the significance of the weight of the single and double estimates is controlled using the parameter c.

The results for the estimated values are illustrated in Fig. 5. For the weighted double estimator method, the results are obtained based on the input parameters $c = 1, 10, 100$. Additionally, it is observed that in all domains, as N increases from 2 to 256, the magnitude of overestimation by the single estimator increases. Moreover, the proximity of c to the double estimator increases with an increase in its value. Similarly, the proximity of c to the

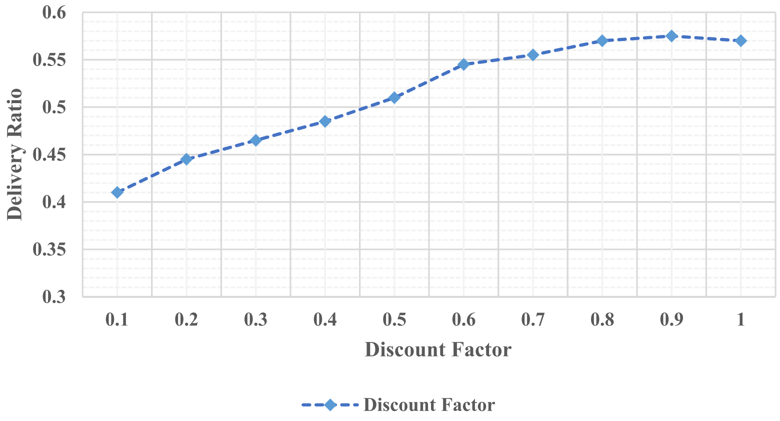

Fig. 4. Delivery ratio vs discount factor (η).

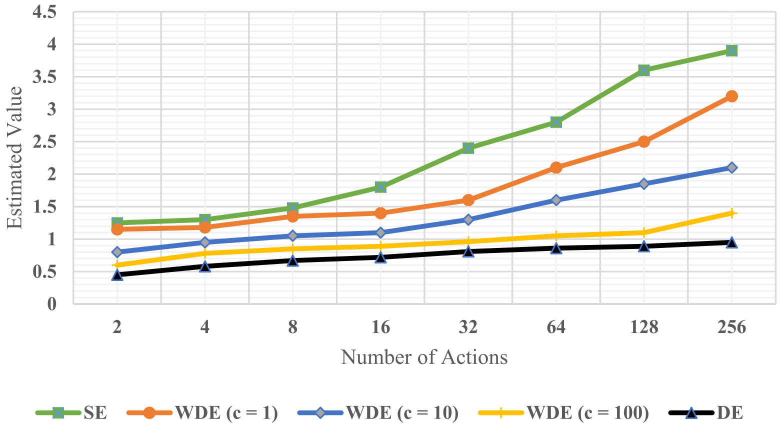

Fig. 5. Estimated value vs number of actions.

single estimator increases as the value of c tends to 0. Figure 5 also reveals that there is no established best value of c.

4.1.1. Results using the SPMBM model

The effects on the above-mentioned performance metrics using the SPMBM model under varying TTL are shown in Fig. 6, and Fig. 7 respectively. Here, the simulation time is 43200 seconds. It is observed that when the TTL is increased, the average latency also increases. This is due to the fact that a substantial of TTL value also increases the stay of the message in the node's buffer. In Fig. 7, the overhead ratio also increases as the TTL increases, and the ORAL ratio yields the smallest overhead ratio compared to DQLR, QLR, RLPRoPHET and Epidemic. In addition, Fig. 6 shows that the average delay increases with an increase in TTL, and ORAL delivers the lowest delay relative to other schemes.

Figure 6 shows the dependence of the average latency with the TTL for all the scenarios. From the graph, it has been monitored that with the increase in TTL the average latency also increases. This occurs due to substantial TTL value increases the stay of the message in the node's buffer.

The proposed ORAL protocol is more purposeful than benchmark protocols, and, takes advantage of information learned in the past. Compared to Q-Learning protocol, ORAL protocol utilizes two value function to solve Q-Learning protocol problem, ensuring that the selection of next hop is optimal.

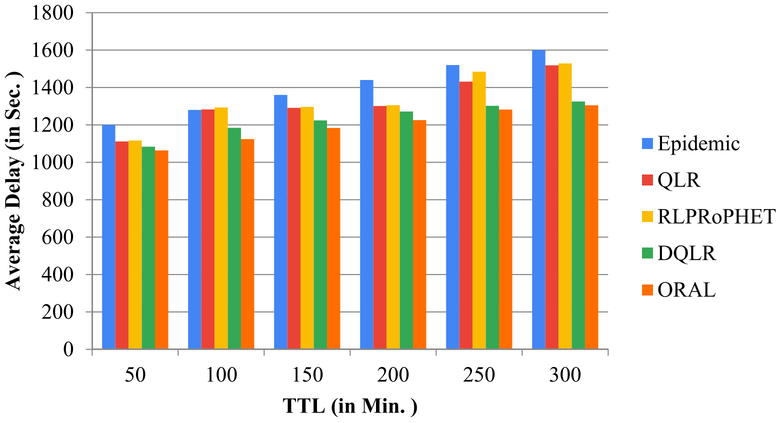

Fig. 6. Average delay vs TTL.

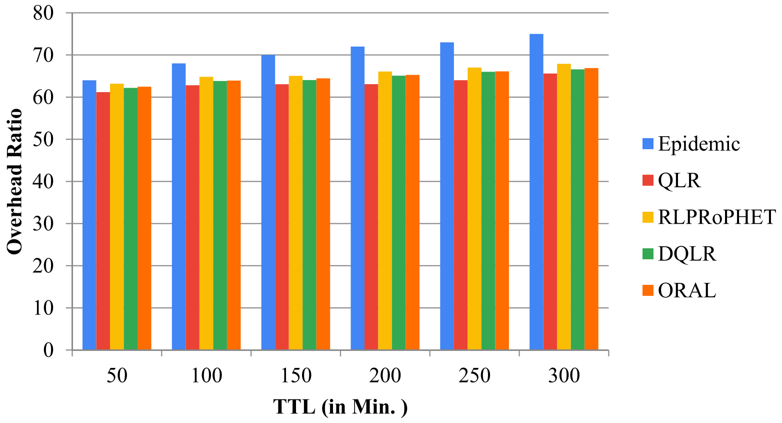

Fig. 7. Overhead ratio vs TTL.

4.1.2. Results using the real mobility data traces

Using the INFOCOM 2006 Real Mobility Data Traces, the results obtained using the above-mentioned performance metrics under varying TTL are given in Fig. 8, Fig. 9 and Fig. 10 respectively. In Fig. 8 show that the delivery ratio of ORAL, DQLR, QLR, RLPRoPHET and Epidemic increases when the TTL is increased. This is because when TTL is increased, the lifespan assigned to each message is enlarged, and the message delivery probability decreases as more messages are kept in the buffers of nodes. The considered simulation time is 259200 seconds, and it is found that the same observations made in the case of the SPMBM model prevail. First, Fig. 8 is displayed between delivery probability and TTL (Time to Live) and observed that the delivery probability of ORAL, DQLR, RLPRoPHET, QLR, and, Epidemic increases as the TTL is increased. This is due to the fact that the time duration allotted to each message gets increased as the TTL increases, and when more messages gets stored in the node's buffer, the message delivery probability increases. Figure 9 shows the dependence of the average latency with the TTL for all the scenarios. From the graph, it has been monitored that with the increase in TTL the average latency also increases. This occurs due to substantial TTL value increases the stay of the message in the node's buffer. The mean average latency value of ORAL is the lowest among all the scenario/techniques that are 5375.2161 seconds. In context to this the performance of ORAL is 2.16% better than DQLR, and 3.95% better than RLPRoPHET respectively. Hence, when time-to-live is varied, (1) the proposed ORAL scheme outperforms DQLR by 14.05%, 9.4%, 5.81% respectively in terms of delivery probability, overhead ratio and average delay; (2)

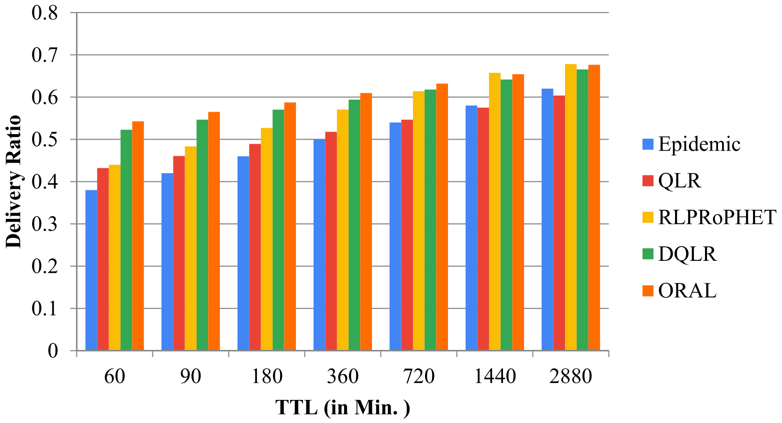

Fig. 8. Delivery ratio vs TTL (using the real dataset).

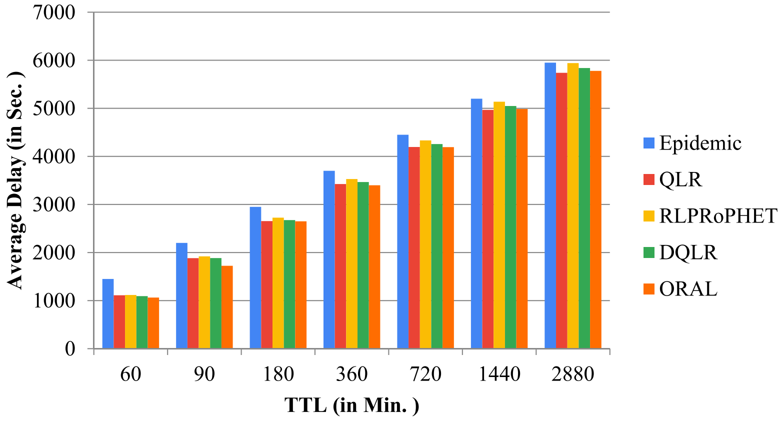

Fig. 9. Average delay vs TTL (using the real dataset).

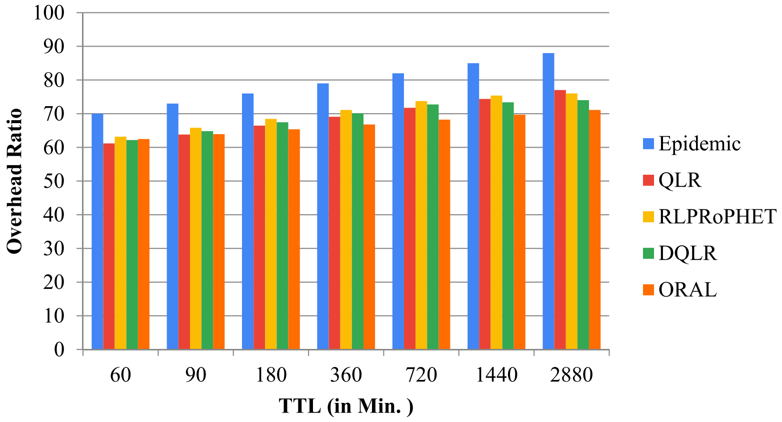

Fig. 10. Overhead ratio vs TTL (using the real dataset).

it also outperforms RLPRoPHET by 16.17%, 9.2%, 6.85%, respectively in terms of delivery ratio, overhead ratio and average delay.

5. Conclusion

This paper has proposed an Off-Policy Reinforcement-based Adaptive Learning for Routing in Opportunistic Networks, called ORAL, in which the mechanism to select the most optimal nodes to forward the message to the destination is performed using a weighted double Q-estimator. A probability-based reward strategy has been implemented to calculate the state-action rewards based on the delivery predictability of the nodes. Simulation results have shown that the proposed ORAL protocol can perform reasonably well while achieving a better delivery ratio without any overestimation or underestimation in predicting the optimal next-hop for the agent. In addition, compared to four benchmark routing protocols for OppNets, namely the DQLR, QLR, RLPRoPHET and Epidemic schemes, the ORAL scheme has shown superior performance in terms of delivery ratio, delivery delay, and overhead ratio. As future work, it would be beneficial to further assess the effectiveness of the ORAL under various different mobility models [11].

Conflict of interest

The authors have no conflict of interest to report.

References

[1] CRAWDAD Dataset Cambridge HAGGLE. https://crawdad.org/uoi/haggle/20160828/one, last accessed on December 01, 2019.

[2] Z. Ali Khan, O. Abdul Karim, S. Abbas, N. Javaid, Y.B. Zikria and U. Tariq, Q-learning based energy-efficient and void avoidance routing protocol for underwater acoustic sensor networks, *Computer Networks* (2021), 108–309. doi:10.1016/j.comnet.2021.108309.

[3] A. Balasubramanian, B. Levine and A. Venkataramani, DTN routing as a resource allocation problem, *ACM SIGCOMM Computer Comm. Review* **37** (2007), 373–384. doi:10.1145/1282427.1282422.

[4] L. Chancay García, E. Hernández Orallo, P. Manzoni, A. Vegni, V. Loscrì, J. Carlos Cano and C. Tavares Calafate, Optimising message broadcasting in opportunistic networks, *Computer Communications* **157** (2020), 162–178, https://dx.doi.org/10.1016/j.comcom.2020.04.031. doi:10.1016/j.comcom.2020.04.031.

[5] A. Elwhishi, P.-H. Ho, K. Naik and B. Shihada, ARBR: Adaptive reinforcement-based routing for DTN, *IEEE Intl. Conf. on Wireless and Mobile Computing, Networking and Communications* (2010), 376–385. doi:10.1109/WIMOB.2010.5645040.

[6] M. Hessel, J. Modayil, H.V. Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan Bilal Piot, M.G. Azar and D. Silver, Rainbow: Combining improvements in deep reinforcement learning, *Intl. AAAI Conf. on Artificial Intelligence* (2018), 3215–3222, https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewFile/17204/16680.

[7] T. Hu and Y. Fei, QELAR: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks, *IEEE Trans. on Mobile Computing* **9** (2010), 796–809. doi:10.1109/TMC.2010.28.

[8] A. Keränen, J. Ott and T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Rome, Italy, 2009. doi:10.4108/ICST.SIMUTOOLS2009.5674.

[9] S.C. Nelson, M. Bakht and R. Kravets, ncounter-based routing in DTNs. IEEE INFOCOM (2009), 846–854 doi:10.1109/INFCOM.2009.5061994.

[10] L. Pelusi, A. Passarella and M. Conti, Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks, *IEEE Communications Magazine* **44** (2006), 134–141. doi:10.1109/MCOM.2006.248176.

[11] I. Rhee, M. Shin, S. Hong, K. Lee, S.J. Kim and S. Chong, On the Levy-walk nature of human mobility, *IEEE/ACM Trans on Networking (TON)* **19**(3) (2011), 630–643. doi:10.1109/TNET.2011.2120618.

[12] V.G. Rolla and M. Curado, A reinforcement learning-based routing for delay tolerant networks, *Engineering Applications of Artificial Intelligence* **26** (2013), 2243–2250. doi:10.1016/j.engappai.2013.07.017.

[13] D.K. Sharma, S. Kumar Dhurandher, D. Agarwal and K. Arora, kROp: k-means clustering based routing protocol for opportunistic networks, *Journal of Ambient Intelligence and Humanized Computing* **10**(4) (2019), 1289–1306. doi:10.1007/s12652-018-0697-3.

[14] D.K. Sharma, J. Rodrigues, V. Vashishth and A. Khanna, RLProph: A dynamic programming based reinforcement learning approach for optimal routing in opportunistic IoT networks, *Wireless Networks* **26**(6) (2020), 4319–4338, https://link.springer.com/article/10.1007/s11276-020-02331-1. doi:10.1007/s11276-020-02331-1.

[15] H. van Hasselt, Double Q-learning, in: *Advances in Neural Information Processing Systems*, Vol. 88, 2010, pp. 2613–2621, http://papers.nips.cc/paper/3964-double-q-learning.

[16] H. van Hasselt, A. Guez and D. Silver, in: *Deep Reinforcement Learning with Double Q-learning. Intl. AAAI Conf. on Artificial Intelligence*, Phoenix, AR, USA, 2016, pp. 2094–2100, https://ojs.aaai.org/index.php/AAAI/article/view/10295.

[17] C. Watkins and P. Dayan, Q-learning, *Machine learning* **8** (1992), 279–292. doi:10.1007/BF00992698.

[18] C. Wu, K. Kumekawa and T. Kato, Distributed reinforcement learning approach for vehicular ad hoc networks, *IEICE Trans. on Communications* **93** (2010), 1431–1442. doi:10.1587/transcom.E93.B.1431.

[19] J. Wu, Z. Chen and M. Zhao, An efficient data packet iteration and transmission algorithm in opportunistic social networks, *Journal of Ambient Intelligence and Humanized Computing* **11** (2020), 3141–3153. doi:10.1007/s12652-019-01480-2.

[20] F. Yuan, J. Wu, H. Zhou and L. Liu, A Double Q-Learning Routing in Delay Tolerant Networks, IEEE ICC 2019 (2019), 1–6. doi:10.1109/ICC.2019.8761526.

[21] Z. Zhang, Z. Pan and M.J. Kochenderfer, in: *Weighted Double Q-learning. Intl. Joint Conf. on Artificial Intelligence*, Melbourne, Australia, 2017, pp. 3455–3461, https://dl.acm.org/doi/abs/10.5555/3172077.3172372.