International Journal for Traffic and Transport Engineering, 2023, 13(1): 125 - 137
DOI: http://dx.doi.org/10.7708/ijtte2023.13(1).10 UDC: 656.025.4

COMPARISON OF TRADITIONAL AND REINFORCEMENT LEARNING BASED ROUTING PROTOCOLS IN VANET SCENARIO

Nenad Jevtić[1], Pavle Bugarčić[2]
[1,2] *University of Belgrade, Faculty of Transport and Traffic Engineering, Vojvode Stepe 305, 11000 Belgrade, Serbia*

Received 12 January 2023; accepted 25 February 2023

Abstract: Vehicular ad hoc networks (VANETs) are characterized by high mobility of nodes and frequent changes in the network topology, which significantly complicates the process of routing data packets. It has been shown that traditional routing protocols are unable to promptly follow these changes and cannot be efficiently used in VANETs for vehicle to vehicle (V2V) communications. This is the reason why protocols based on reinforcement learning (RL) have been developed. These protocols enable constant monitoring of changes in the network environment, and adaptation of the routing process to those changes. In this paper, an analysis and comparison of the traditional and RL based routing protocols are performed in VANET scenario. The Ad hoc on-demand distance vector routing protocol (AODV) and AODV with Expected transmission count (ETX) metric are chosen as the representatives of traditional routing protocols, while the Adaptive routing protocol based on reinforcement learning (ARPRL) is chosen as the representative of routing protocols based on RL. The simulation results show that the ARPRL protocol has significantly better network performance in terms of packet loss ratio (PLR) and end-to-end delay (E2ED) in urban VANET scenario.

Keywords: routing protocols, reinforcement learning, VANET, ETX metric.

1. Introduction

With the development of smart cities and intelligent transportation systems, vehicular ad hoc networks (VANETs) are becoming increasingly important. VANETs represent a special category of wireless ad hoc networks (WANETs), where the network consists of a set of vehicles that communicate with each other (V2V) and with infrastructure (V2I) via wireless ad hoc links. The process of choosing the optimal route from source to destination in V2V networks is a challenging task since their topology is constantly changing, which can cause frequent link breaks. In these conditions, traditional routing techniques show significant limitations. This leads to the degradation of various network performances, such as end-to-end delay (E2ED), throughput, packet loss ratio (PLR), etc.

Due to the mentioned problems, many protocols developed for mobile ad hoc networks (MANETs), among which Ad hoc on-demand distance vector (AODV) (Perkins *et al.,* 2003) is certainly one of the most important, could not be successfully applied in the VANET scenario. Mubarek *et al.* (2018) and Bugarčić *et al.* (2019) have

[1] Corresponding author: n.jevtic@sf.bg.ac.rs

proposed certain modifications of this protocol, with the aim of adapting it to VANETs, achieving only partial success. Ardianto *et al.* (2022) and Jevtić and Malnar (2019) proposed the use of routing metrics, such as Expected transmission count (ETX) (De Coute *et al.,* 2005), as a method of improving the AODV protocol, while Malnar and Jevtić (2022) proposed the hybrid use of protocol modifications and metrics.

In order to overcome these problems and improve the routing process in VANETs, some authors propose the use of artificial intelligence when choosing the optimal route for sending data packets. The most commonly used field of artificial intelligence in routing protocols for VANETs is machine learning (ML) and the type of ML that gives the most promising results is reinforcement learning (RL). The main characteristic of this type of learning is the constant interaction of the learning agent with the environment, which allows monitoring and adapting to changes in the environment. Therefore, reinforcement learning is particularly suitable for use in highly dynamic networks where the topology changes frequently. Various RL algorithms have been used in the literature. For example, Wu *et al.* (2018) used the Q-learning (QL) algorithm introduced by Sutton and Barto (2018), while Saravanan and Ganeshkumar (2020) used the advanced deep RL (DRL) algorithm from Mnih *et al.* (2015). To further improve the performances and increase the stability of RL, Zhang *et al.* (2018) used the dueling DRL (DDRL) concept defined by Wang *et al.* (2016), which represents an improvement of the DRL algorithm. Another type of RL defined by Sutton and Barto (2018), the SARSA algorithm, is used in routing protocols for VANETs and implemented by Bi *et al.* (2020). The characteristic of all mentioned algorithms is that they are not based on the model of the environment, i.e. they all belong to the group of model-free algorithms. Jafarzadeh *et al.* (2020) proposed a model-based RL (MBRL) algorithm that first needs to create an internal model of the environment, and based on it, the optimal routing policy will be determined. Bugarčić *et al.* (2022) performed an overview and classification of all important routing protocols based on reinforcement learning in the VANETs.

In this paper, we analyze and compare the results of the application of traditional routing protocols and routing protocols based on reinforcement learning in VANETs. The AODV protocol is chosen as a representative of traditional routing protocols, as one of the most popular reactive routing protocols for WANETs. Also, the performance of an improvement of AODV protocol that uses ETX metric (AODV-ETX) (Jevtić and Malnar, 2019) is considered in comparison. On the other hand, the Adaptive routing protocol based on RL (ARPRL) (Wu *et al.,* 2018) is chosen as the representative of routing protocols which use reinforcement learning. These protocols are compared according to network performance indicators PLR and the average E2ED.

The rest of the paper is organized as follows. In the second section, the basic principles of RL are explained. In the third section, the fundamentals of the AODV, AODV-ETX, and ARPRL routing protocols are described. In the fourth section, a comparison and analysis of the network performance obtained when using these three protocols are performed. After that, a discussion based on all the previous results and observations is given. In the final section, concluding considerations are summarized.

2. Reinforcement Learning

RL (Sutton and Barto, 2018) is the most common type of ML in routing protocols for dynamic WANETs. This type of learning involves learning through constant interaction with the environment to achieve a certain goal. To describe this process, it is first necessary to define the most important elements in the learning process. The decision-maker in the RL process is called the agent. Everything surrounding the agent and what he interacts with is called the environment. At any discrete moment *t*, the environment can be in a certain state s_t, which belongs to the finite set of possible states *S*. The agent decides to take a certain action a_t, from a finite set of actions *A*, which are available to the agent in the current state s_t. The environment responds to this action with feedback to the agent, which contains the new state of the environment after the action is taken, s_{t+1}, as well as the numerical reward for the taken action, r_{t+1}. In this way, the environment reinforces the agent with knowledge about the usefulness of the actions he takes. Over time, the agent tries to maximize the reward by optimizing the choice of possible actions. A schematic representation of the agent's interaction with the environment is shown in Figure 1.

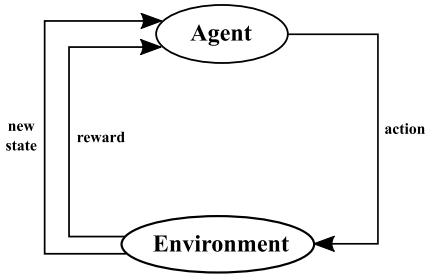

Fig. 1
Agent-environment Interaction in the RL Process

The RL process in one WANET can be modeled in several ways. The most commonly used approach is that each node in the network that sends packets represents a learning agent, while the entire network represents the environment. Sending packets to one of the neighboring nodes represents a potential action that the agent can take. Since each node has a finite set of neighbors, it represents a set of possible actions that the node can take. The feedback received by the sender contains a reward for the taken action and the new state of the environment.

One of the simplest RL algorithms is QL, in which each agent maintains a table of Q-values that refer to the usefulness of taking a specific action at a particular moment. Based on these values, the agent makes decisions about future actions. Each element of this table is calculated using following equation:

$$Q(s_t, a_t) \leftarrow (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot (r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a)) \quad (1)$$

where r_{t+1} represents the reward for the action taken in the appropriate state, α represents the learning rate that affects the learning speed and can take a value in the range $[0,1]$, γ represents the discount factor that determines the importance of future rewards and can also take a value in the range $[0,1]$, while $\max_a Q(s_{t+1}, a)$ is the maximum possible Q-value that an agent can achieve by taking an action a, from the set of possible actions A, in the state s_{t+1}. The agent must strike a balance between the exploitation of acquired knowledge and the exploration of the environment, which is necessary to update its knowledge based on changes in the state of the environment. This can be done by defining an action selection policy, and the most commonly used is ε-greedy. According to this policy, an agent with a probability ε takes the action with the highest Q-value (knowledge exploitation), while with a probability (1-ε) takes a randomly selected action from a set A (environmental exploration).

3. Routing Protocols for VANETs

Routing protocols are responsible for determining and maintaining the optimal route for packet forwarding in a network. In VANETs, all nodes participate in packet routing. The main challenge faced by routing protocols in these networks is the high mobility of network nodes, which lead to frequent changes in network topology. In such conditions, there may be a late detection of a link break on the path used to send packets, which results in packet loss and reduced network throughput. For this reason, new approaches in routing are necessary, which include faster detection of changes in the network and timely selection of a new route in case of link breakage on the old route. This section will show, on the one hand, the traditional approach to packet routing, and on the other hand, the newer approach that involves the inclusion of artificial intelligence in the routing process. First, the principle of functioning of the AODV protocol, as one of the most popular traditional reactive routing protocols, will be described, and then the AODV-ETX that includes well known ETX metric in this protocol. Finally, special attention is given to the explanation of the basic principles of the ARPRL protocol, which belongs to the RL-based routing protocols.

3.1. AODV Protocol

The AODV protocol enables dynamic, multi-hop routing between mobile nodes attempting to establish and maintain communication in an ad hoc network. The protocol consists of two main mechanisms: "route discovery" and "route maintenance", which work together so that nodes can discover and maintain routes to arbitrary destinations in the network.

When there is a need to send data, the AODV protocol activates a route discovery mechanism using Route request (RREQ) and Route reply (RREP) control packets. A node that wants to send data packets first sends broadcast RREQ packets with the aim of finding the best route to the destination. An RREQ packet can be received by an intermediate node or a destination node. If the packet is received by an intermediate node, it will first update the route to the source node in its routing table, check if it has a suitable route to the desired destination and if it does, send it to the source node via an RREP packet. Otherwise, it will rebroadcast the RREQ packet further into the network. When the destination node receives the RREQ packet, it first creates (or

updates) its routing table with information about the route to the source node, and then sends the RREP packet to the source node along the same route by which the RREQ packet arrived. When it receives the RREP packet, the source node starts sending the data packets to the destination along the same path that it received the RREP packet. If it receives multiple RREP packets, the source node chooses the route with the least number of hops to the destination.

The route maintenance mechanism uses Hello control packets to check route validity. Nodes periodically send Hello packets to their neighbors to inform other nodes that they are active. If a node does not receive a Hello packet in a predefined time interval from one of its earlier neighbors, it will consider that the link to that neighbor is broken. In the event of a link break, the node sends Route error (RERR) packets to inform all of its neighbors who used that link that it is no longer available. Each node has its routing table where data about routes are stored following a predefined time (Delete period). After this time, the data from the routing tables in the nodes will be deleted.

3.2. AODV-ETX Protocol

AODV implicitly uses the shortest path (hop-count) metric, considering that upon receiving the RREP packets, it chooses the route that has the fewest hops to the destination. This is known to be sub-optimal, especially in the fast-changing environments such as VANETs. Therefore, AODV protocol can be significantly improved by using more advanced routing metric that take into account link state, channel interference, movement of the network nodes, etc. One of the simplest and mostly used metrics is ETX. ETX metric of a link is represented using the probability of successful transmissions of packets over that link. If p_t represents the probability of successful packet transmission, and p_r the probability of successfully received packet, then the probability of a packet to be successfully sent and acknowledged will be $p_t \cdot p_r$. ETX metric for a link l, is given by:

$$ETX_l = \frac{1}{p_t \cdot p_r}. \quad (2)$$

Both probabilities (p_t and p_r) are typically measured using dedicated Link probe packets (LPPs), which are broadcasted every τ seconds. Every node remembers the number of received LPPs during the last w seconds allowing it to calculate the probability p_r at any time t as:

$$p_r = \frac{count(t-w,t)}{w/\tau} \quad (3)$$

Count $(t-w, t)$ is the number of LPPs received during the window w, and w/τ is the number of LPPs that should have been received. This way, some node A can easily measure p_r by counting successfully received LPPs from its neighbor B. However, due to the lack of acknowledgments for broadcast packets, node A cannot determine the probability p_t for transmission to his neighbor B. Therefore, each LPP sent by node B contains the number of LPPs received from all of its neighbors (including A) during the last w seconds. With this value node A can calculate the p_t for node B. The metric ETX_r of a route r from source to destination node is calculated as the sum of the ETX_l values for each link l in the route:

$$ETX_r = \sum_{l \in r} ETX_l \quad (4)$$

Although ETX metric has proven to provide better results than hop-count, its main drawback is increased overhead.

3.3. ARPRL Protocol

To improve the network performance that degrades due to frequent changes in the network topology that the AODV protocol cannot successfully follow, the ARPRL routing protocol based on reinforcement learning is proposed. Specifically, this protocol uses the QL algorithm, which is one of the most commonly used types of reinforcement learning.

For packet routing, each node uses its Q-table, which consists of Q-values that are updated by exchanging Hello packets between neighboring nodes, receiving data packets, and using feedback from the medium access control (MAC) layer. In addition to the Q-table, each node maintains a neighbor table, so that it always knows the set of available neighbor nodes. This table is only updated by exchanging Hello packets. When a node wants to send data packets to a destination, it first checks its Q-table to see if it has a next hop to that destination. If there is no next hop, it initiates the route discovery process using Learning probe request (LPREQ) and Learning probe reply (LPREP) control packets, which is very similar to the route discovery process of the AODV protocol.

Each vehicle in the network acts as a learning agent and continuously collects information about the state of the links in the network, which is exchanged by periodically sending Hello packets between neighbors. The structure of the Hello packet is shown in Figure 2. Each Hello packet contains the following fields: ID, position and speed of the source node, creation time of the Hello packet, the number of maximum Q-values contained in the Hello packet, which is followed by a series of maximum Q-values. If a node has multiple potential routes to a certain destination, it will choose the one with the highest Q-value that represents the maximum Q-value for that destination. By repeating this process for all destination nodes in the network, a sequence of maximum Q-values is created. Each maximum Q-value field contains the IP address of the destination node, the corresponding Q-value, and the IP address of the next hop on the path to the destination.

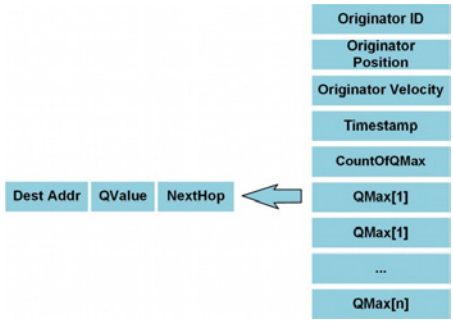

Fig. 2.
Hello Packet Structure

Each node that has packets to send to a specific destination selects the next hop with the highest Q-value to choose the best route to send the packets. That is why it is very important to update these values frequently, due to the dynamic nature of VANETs. The first way to update the Q-values is through control packets. Upon receiving a Hello packet, each vehicle updates its Q-values based on the information from this packet. If a node does not receive a Hello packet from a particular neighbor within a predefined time interval, the Q-value for that neighbor will be reset to 0. After receiving a Hello packet from its neighbor n, the current vehicle c will update its Q-table according to the following equation:

$$Q_c(d,n) = \left(1-\alpha_{c,n}\right)\cdot Q_c(d,n) + \alpha_{c,n}\left[R_{c,n} + \gamma_{c,n}\cdot max_{y\epsilon Nei(n)} Q_n(d,y)\right] \tag{5}$$

where d is the destination vehicle, $Nei(n)$ represents the set of neighbors of node n, and $\alpha_{c,n}$, $\gamma_{c,n}$, and $R_{c,n}$ represent the learning rate, discount factor, and reward, respectively, and are defined as follows:

$$\alpha_{c,n} = max\left(0.2,\ \frac{\left||v_c|-|v_n|\right|}{v_{max}-v_{min}}\right) \tag{6}$$

where v_c and v_n represent the speed of nodes c and n, and v_{max} and v_{min} the maximum and minimum speed in the network;

$$\gamma_{c,n} = \begin{cases} \frac{\sum_{i=1}^{N} R_{c,n}}{N}, & N \neq 0 \\ 0, & N = 0 \end{cases} \tag{7}$$

where N is the total number of nodes in the network;

$$R_{c,n} = C + HMRR_{c,n} + LET_{c,n} \tag{8}$$

where C is a constant with a value of 100, while $HMRR_{c,n}$ and $LET_{c,n}$ represent the Hello message reception ratio and the link stability factor, respectively, and are calculated based on the following equations:

$$HMRR_{c,n} = \begin{cases} 100\cdot\frac{CNT_r(c,n)}{CNT_s(n)}, & CNT_s(n) \geq 15 \\ 100\cdot\frac{CNT_r(c,n)}{CNT_s(n)}\cdot\left(1-\left(\frac{1}{2}\right)^{CNT_s(n)}\right), & otherwise \end{cases} \tag{9}$$

where $CNT_r(c,n)$ is the number of Hello messages received by node c from node n, and $CNT_s(n)$ is the number of Hello messages sent by node c to node n;

$$LET_{c,n} = \begin{cases} 100, & A = 0\ and\ B = 0 \\ min\left(100, \frac{-(AB+CD)+\sqrt{\left(A^2+C^2\right)\cdot R^2-(AD-BC)^2}}{A^2+B^2}\right), & otherwise \end{cases} \tag{10}$$

where A, B, C and D are calculated using the following equations:

$$A = v_c\, cos(\Theta_{v_c}) - v_n cos(\Theta_{v_n}) \quad (11)$$

where (Θv_c) and (Θv_n) represent the projections of the speeds of nodes c and n on the x axis, respectively;

$$B = x_c - x_n \quad (12)$$

where x_c and x_n represent the x coordinates of nodes c and n respectively;

$$C = v_c\, sin(\Theta_{v_c}) - v_n sin(\Theta_{v_n}) \quad (13)$$

where (Θv_c) and (Θv_n) represent the projections of the speeds of nodes c and n on the y axis, respectively;

$$D = y_c - y_n \quad (14)$$

where y_c and y_n represent the y coordinates of nodes c and n respectively.

It is obvious that nodes with higher *LET* will have higher Q-values, which means that more stable routes are preferred when choosing the next hop. In order to demonstrate this, a simple VANET scenario is presented in Figure 3. Communication between vehicles S and D is possible via two potential routes: one via vehicle A (S→A→B→D) and the other via vehicle C (S→C→E→D). As vehicle A moves further and further away from vehicle S, while vehicle C continues straight as vehicle S, the first route will lose connectivity after a while due to a link break (S'→A'), and the route via vehicle C remains valid. Accordingly, the neighbouring vehicle C is more suitable to be selected as the next hop on the path from vehicle S to vehicle D.

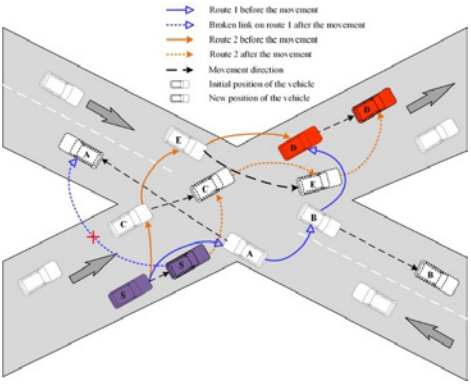

Fig. 3.
VANET Scenario showing the Role of LET in Packets Routing

Second way to update the Q-value is after receiving data packets. When it receives the data packets sent from the source node s via the neighbour node n, the node c updates the Q-values via the equation:

$$Q_c(s,n) = \left(1 - \alpha_{c,n}\right) \cdot Q_c(s,n) + \alpha_{c,n}\left[R_{c,n} + \gamma_{c,n} \cdot max_{y \in Nei(n)} Q_n(s,y)\right] \quad (15)$$

A third type of Q-value update is based on feedback from the MAC layer. Upon receiving information from the MAC layer about packet loss from the neighbouring node *n*, node *c* for each destination d_i updates the Q-values through the following equation:

$$Q_c(d_i, n) = 0.5 \cdot Q_c(d_i, n) \quad (16)$$

which means that the Q-values for routes via the neighbour node *n* will decrease after each packet loss notification at the MAC layer.

4. Simulation Results

The simulations are performed using the Network simulator v3 (NS-3) and the well-known Simulation of urban mobility (SUMO). The simulation scenario uses a common Manhattan Grid mobility model with 20 horizontal and 20 vertical streets in a 2000 m x 2000 m field. Each street has 2 lanes for movement in both directions. In order to make the scenario as realistic as possible, every fifth intersection along the horizontal and vertical axis contains a traffic light. The SUMO simulator generates the movement of vehicles in a defined space, where the vehicles move through the streets with a speed limit of 15 m/s. The described simulation scenario is shown in Figure 4, where one of the intersections with traffic lights is shown in detail.

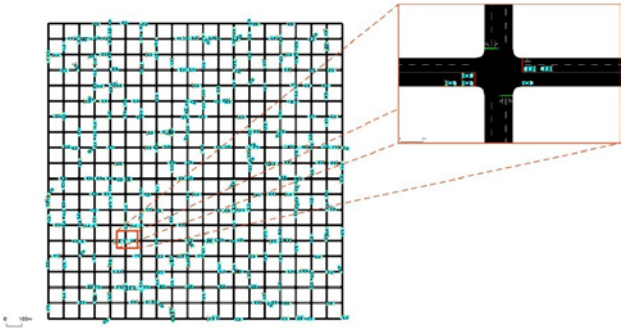

Fig. 4.
Simulation Scenario: 2000 m x 2000 m, 20 Streets on Each Axis, Every Fifth Intersection with Traffic Lights

The following values are assumed for the communication parameters of the simulation. The IEEE 802.11p standard for wireless networks is used, with a bandwidth of 10 MHz and a throughput of 6 Mb/s. Two ray ground is selected for the propagation model. The traditional AODV protocol, AODV-ETX protocol with ETX metric, and the previously described ARPRL protocol, which represents a modification of the AODV protocol based on reinforcement learning, are tested and compared. The User datagram protocol (UDP) is used at the transport layer. The application layer generates packets of size 512 B using Constant bit rate (CBR) traffic, whereby the achieved application throughput is 4 kb/s. Packet traffic at the application layer is generated by ten randomly selected vehicles, while another ten vehicles, also randomly selected, receive the generated packets. Two indicators of network performance are observed (PLR

and E2ED), 200 iterations of simulations are performed with different settings for the random number generator and then the mean values for both indicators are calculated. Other parameters have default settings for the network simulator. An overview of the most important simulation parameters is shown in Table 1.

Table 1

Overview of Simulation Parameters

Parameter	Value
Simulation area	2000 m x 2000 m
Streets	20 horizontal and 20 vertical streets with traffic lights
Simulation duration	600 s
Number of vehicles	50 - 300, with a step of 50
Maximum vehicle speed	15 m/s
Mobility model	Manhattan grid
Propagation model	Two ray ground
MAC standard	IEEE 802.11p
Channel width	10 MHz
Channel throughput	6 Mb/s
Routing protocols	AODV, AODV-ETX, ARPRL
Transport layer protocol	UDP
Application throughput	4 kb/s (CBR, 10 vehicles)
Packet size	512 B
Observed performances	PLR, E2ED
Iterations of simulations	200

Vehicle density in VANETs has a significant impact on protocol performance. In this paper, the analysis is performed with a fixed maximum vehicle speed of 15 m/s, while the total number of vehicles in the network is varied from 50 to 300, which increased the traffic density in the network. The results of the simulations are shown in Figures 5 and 6.

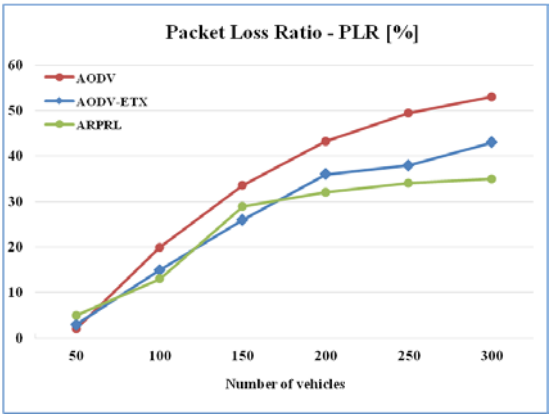

Fig. 5.

Dependence of Packet Loss Ratio on Vehicle Density

Figure 5 shows the average PLR for all three routing protocols depending on the number of vehicles in the simulation environment. With the increase in the vehicles density, the probability of finding a high-quality route increases, and in certain situations, an improvement in the results, i.e. a decrease in the value of the PLR, can be expected. However, in the considered scenario, due to the high dynamics of the vehicles, the network topology changes rapidly and there are frequent link interruptions, causing sending of RERR packets and the resending of RREQ packets. Frequent activation of the route discovery mechanism leads to flooding and network degradation in terms of loss of user packets due to an increased number of collisions. This phenomenon is especially pronounced when the number of RREQ packets is large, that is, when there are many vehicles in the network. Therefore, a clear trend of increasing PLR for AODV protocol with increasing vehicle density is observed in Figure 5. Inclusion of ETX metric in AODV protocol shows improvements, since this metric enables the selection of high-quality routes, and therefore the number of RREQ and RERR packets is significantly reduced. However, with the increment of the number of vehicles in the VANET, packet losses still increase significantly. The ARPRL protocol uses proactive Q-learning that enables it to choose even better and more stable routes than with ETX metric, as well as keep track of several routing options to quickly change the route if one of the links on that route is interrupted. Therefore, the ARPRL protocol shows significantly less performance degradation in terms of PLR.

Figure 6 shows the average network E2ED for each routing protocol as a function of the number of vehicles in the simulation environment. For similar reasons as above, network performance results show significant degradation of average E2ED with AODV protocol. As with PLR, the average E2ED is also improved with the introduction of the ETX metric in the AODV protocol. But the approximately exponential rise of the delay with the increase in the number of vehicles has not been eliminated. On the other hand, the ARPRL protocol does not show significant degradation when increasing the number of vehicles in the network, keeping average packet delays within acceptable limits.

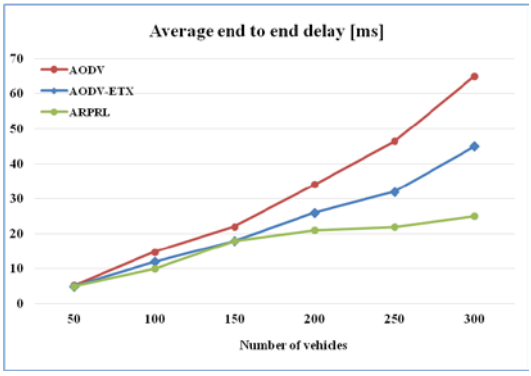

Fig. 6.
Dependence of End-to-end Delay on Vehicle Density

6. Conclusion

In this paper, an analysis and comparison of AODV, AODV-ETX, and ARPRL routing protocols in urban VANETs for V2V applications, is performed. The AODV is a traditional reactive routing protocol, which means that the route discovery mechanism is triggered only when there is a need to send data packets. It is obvious that the AODV protocol is not able to keep up with frequent topology changes in VANETs in a timely manner, which results in a high percentage of lost packets and a large delay of packets in the network. This is especially evident with the increase in the number of network nodes, where due to frequent interruptions of network links, it is necessary to restart the route discovery procedure, which leads to network overload and network performance degradation. ETX metric improve performance of AODV protocol since it helps in choosing high-quality routes. This reduces link breakage, but cannot help in fast adaptation of routing protocol to frequent topology changes. On the other hand, the ARPRL protocol constantly monitors alternative routes and their quality with the help of reinforcement learning, which allows fast adaptation to changes in the network topology and high probability of choosing the optimal path from source to destination nodes. This results in significantly better network performance in terms of the PLR and E2ED, which is especially noticeable as the number of vehicles in the network increases.

As part of future research, further analysis and testing of reinforcement learning-based routing protocols for VANETs is planned, as well as the development of a new protocol that would be even better adapted to the dynamic nature of VANETs. Also, the research can be extended to flying ad hoc networks (FANETs), which are becoming more and more attractive with the accelerated development of unmanned aerial vehicles. In addition, it is possible to test the application of advanced reinforcement learning algorithms, such as deep reinforcement learning and dueling deep reinforcement learning.

References

Ardianto, B. et al. 2022. Performance Comparison of AODV, AODV-ETX and Modified AODV-ETX in VANET using NS3. In *Proceedings of the IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 156-161.

Bi, X. et al. 2020. A reinforcement learning-based routing protocol for clustered EV-VANET. In *Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 1769–1773.

Bugarčić, P. et al. 2019. Modifications of AODV protocol for VANETs: performance analysis in NS-3 simulator. In *Proceedings of the 27th Telecommunications Forum (TELFOR)*, 1-4.

Bugarčić, P. et al. 2022. Reinforcement learning-based routing protocols in vehicular and flying ad hoc networks – a literature survey, *Promet* 34(6): 893-906.

De Couto, S.J. et al. 2005. A high–throughput path metric for multi-hop wireless routing, *Wireless Networks* 11(4): 419-434.

Jafarzadeh, O. et al. 2020. A novel protocol for routing in vehicular ad hoc network based on model-based reinforcement learning and fuzzy logic, *International Journal of Information and Communication Technology Research* 12(4): 10-25.

Jevtić, N.; Malnar, M. 2019. Novel ETX-Based Metrics for Overhead Reduction in Dynamic Ad Hoc Networks, *IEEE Access* 7(2019): 116490-116504.

Malnar, M.; Jevtić, N. An improvement of AODV protocol for the overhead reduction in scalable dynamic wireless ad hoc networks, *Wireless Networks* 28(3): 1039 - 1051.

Mnih, V. et al. 2015. Human level control through deep reinforcement learning, *Nature* 518(7540): 529–533.

Mubarek, F.S. et al. 2018. Urban-AODV: an improved AODV protocol for vehicular ad-hoc networks in urban environment, *International Journal of Engineering & Technology* 7(4): 3030-3036.

Perkins, C.; Belding–Royer, E.; Das, Ss. 2003. Ad Hoc On demand Distance Vector (AODV) routing. *RFC 3561, IETF.*

Saravanan, M.; Ganeshkumar, P. 2020. Routing using reinforcement learning in vehicular ad hoc networks, *Computational Intelligence* 36(2): 682–697.

Sutton, R.; Barto, A. 2018. *Reinforcement learning: An introduction, second edition.* MIT Press, Inc. USA. 526 p.

Wang, Z. et al. 2016. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International conference on machine learning*, 1995-2003.

Wu, J. et al. 2018. Reinforcement Learning Based Mobility Adaptive Routing for Vehicular Ad-Hoc Networks, *Wireless Personal Communications* 101: 2143-2171.

Zhang, D. et al. 2018. A deep reinforcement learning-based trust management scheme for software-defined vehicular networks. In *Proceedings of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet)*, 1-7.