Q-Learning Relay Placement for Alert Message Dissemination in Vehicular Networks

Raoua Chakroun, Slim Abdellatif, Thierry Villemur

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 203 (2022) 222–230

Procedia Computer Science

www.elsevier.com/locate/procedia

The 19th International Conference on Mobile Systems and Pervasive Computing (MobiSPC)
August 9-11, 2022, Niagara Falls, Canada

Q-Learning Relay Placement for Alert Message Dissemination in Vehicular Networks

Raoua Chakroun*, Slim Abdellatif[a], Thierry Villemur[a]

[a]*Laas-CNRS, 7 avenue du colonel roche, Toulouse 31400, France*

Abstract

Alert message (AM) dissemination is a fundamental yet challenging issue in vehicular networks, as it relies on wireless transmissions in a highly mobile, potentially dense, and changing environment. Emerging network infrastructure-based vehicular networks are being considered as an alternative to Vehicular Ad-hoc NETworks (VANETs) for alert message dissemination. Indeed, assuming that Vehicle to Infrastructure (V2I) communication links are nominally available, with some transient and time-limited connectivity losses, recent alert message dissemination schemes primarily rely on V2I links to widely broadcast AMs. Vehicle to Vehicle (V2V) rebroadcasts performed by some selected relay vehicles located within pre-computed rebroadcast zones are then used to ensure the full dissemination within an area of interest. This paper focuses on rebroadcast zones placement. It proposes a Q-learning-based method that computes the minimum number and optimal locations of rebroadcast zones. From these computed zones, the combination of V2I broadcasts with V2V rebroadcasts allows the delivery of AMs in a whole area, even in the presence of locations with poor wireless connectivity. The performance results show that high information coverage and low delivery delays are achieved with our proposed Q-learning based placement. Useless duplicate rebroadcasts and collisions are also avoided, saving network resources.

Keywords: Reinforcement Learning, Vehicular Communications, Alert Message Dissemination, Safety Applications;

1. Introduction

To enable the future Intelligent Transportation System (ITS) applications and services, effective data dissemination in vehicular communication networks is crucial, especially for safety services like cooperative awareness (e.g., emergency vehicle, warning for an accident, etc.). Historically, emergency message dissemination techniques relied on Vehicular Ad hoc NETworks (VANET) with the challenge of mitigating the broadcast storm problem and its undesir-

* Corresponding author. Tel.: +33-561-336-905 ; fax: +0-000-000-0000.
E-mail address: rchakrou@laas.fr

10.1016/j.procs.2022.07.029

able effects [4]. Emerging vehicular networks pave the way to new and more effective approaches that can leverage the nominal presence of access networks (i.e., vehicle to network infrastructure wireless links). Some of the work from the literature have proposed to primarily exploit vehicle to infrastructure (V2I) links to widely broadcast Alert Messages (AM) and complement these transmissions with parsimonious selected vehicle to vehicle (V2V) rebroadcasts [8, 1]. Hence, a relay selection technique is needed to select relay vehicles that may rebroadcast emergency messages to achieve full message delivery within an area of interest without cluttering the wireless medium with useless wireless transmissions. For instance, in [1], a network controller computes rebroadcast points (i.e., locations), and only the vehicles that sit in their close vicinity (i.e., rebroadcast zones) are allowed to rebroadcast an AM. The computation of these rebroadcast zones takes advantage of the centralized view that the network controller builds on the wireless network coverage, radio propagation environment, as well as of the information related to road traffic and vehicles (their characteristics and potentially their road trip).

This paper proposes a Reinforcement Learning (RL) based rebroadcast zones placement algorithm, which provides the minimum number and optimal locations of rebroadcast zones that help achieve full AM dissemination with low delays.

The rest of the paper is organized as follows: Section 2 presents an overview of existing work in the literature. Section 3 describes the problem addressed in this paper. Section 4 presents the proposed method. Section **??** evaluates and discusses our simulation results. Section 6 concludes the paper.

2. Related Work

Alert message dissemination has been widely studied in a VANET/V2V context, as shown in survey papers [5, 6, 7]. All try to address the «broadcast storm problem» caused by massive successive and unnecessary vehicle rebroadcasts, which lead to radio resource wastage, increased delivery delays, and decreased delivery ratios. The idea of exploiting V2I wireless links and enlarging the delivery area of AM thanks to appropriately selected V2V rebroadcasts has not been investigated, except for [8, 1]. In [8], the authors propose a framework for the dissemination of alert messages within an integrated system which comprises a Hybrid VANET-Cellular architecture where the buses act as mobile gateways and a cloud infrastructure which enables rapid data acquisition of road traffic flow and the geographical position of all mobile gateways. This choice efficiently provides essential traffic information (accident, route recommendation, etc.) to the vehicles in the targeted area. Gateways play the role of rebroadcasters by relaying back and forth to the farthest receivers. In [1], V2V rebroadcasts are performed by any vehicle. The selection of relay vehicles is based on their location with respect to pre-identified and computed rebroadcast points. A vehicle receiving an AM is eligible for rebroadcast if it is in the vicinity of a rebroadcast point (i.e., within the associated rebroadcast zone). The paper proposes a simple and general method to compute the rebroadcast points/zones. To the best of our knowledge, there is no other work addressing the problem of rebroadcast zones/points placement. However, some works as [16, 17] are focusing on the problem of optimal RSU placement, where the goal is to find the optimal positions of RSUs to cover a targeted area while considering various constraints, such as the deployment cost and feasibility, applications requirements, etc. They can be seen as complementary to the work presented in this paper, which addresses the problem of rebroadcast points/zones placement once the placement of RSUs is defined.

3. Problem Description

We consider in this work a location-based alert message dissemination scheme as in [1], which in addition to the AM broadcast from the network infrastructure, i.e., RSU, selects a set of relay vehicles that can re-transmit the AM in order to reach vehicles in areas that are not covered (white zones) or poorly covered (gray zones) by the transmissions from the network infrastructure. Indeed, a vehicle that receives an AM individually decides whether it is eligible for rebroadcast by checking its location with respect to pre-defined rebroadcast points sent to the vehicle during handover. The closer a vehicle is from a rebroadcast point, the higher is its priority to rebroadcast the AM. When an AM is rebroadcast by a relay vehicle, surrounding vehicles resume their rebroadcast attempt to avoid wasting network resources.

This paper addresses the problem of rebroadcast points/zones placement. A network controller is assigned to a region of interest wirelessly served by a set of RSUs; it identifies white and gray zones in its region (either from

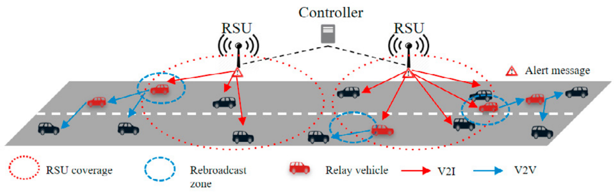

Fig. 1: Location-based alert messages dissemination.

Fig. 2: Environment representation.

a wireless site survey, some network planning, or some form of prediction). It selects the optimal number K and placement of rebroadcast zones where vehicles can rebroadcast an alert message sent by an RSU. These rebroadcast zones are defined by point $P_i(x_i, y_i), i \in [1, K]$, with x_i and y_i are the GPS coordinates of P_i and a radius d_{max} in the order of a few meters from the rebroadcast point.

For illustration, Figure 1 depicts a road section with two RSUs under the control of a network controller. The red dash ellipses delimit the RSUs' wireless coverage. All the other road portions are considered gray/white zones. Three rebroadcast zones (delimited by the blue dashed ellipses) are shown. Vehicle transmissions from these rebroadcast zones are expected to cover the gray/white zones.

4. Proposed RL-Based Rebroadcast Zones Placement

We first summarize the environment representation, the state space, reward function, and the action space used in our RL framework before ending with the Q-learning algorithm.

4.1. *Markov Decision Processes*

RL is the science of decision-making or the optimal way of making decisions in an environment with which an agent (that implements the RL algorithm) can interact. In RL, the learning agent can be studied by adopting Markov Decision Process (MDP) formalism. An MDP is defined as a (S, A, r) triples, where S stands for the set of possible states, A_{s_t} is the set of possible actions from state $s_t \in S$ to $s_{t+1} \in S$, $r_a(s_t, s_{t+1})$ is the immediate reward, earned from the transition from state s_t to state s_{t+1} by performing an action a. The decision policy π is a function that maps state to action (the agent's brain); basically, a policy function says what action to perform in each state. The ultimate objective with an MDP lies in finding the optimal policy π^* which specifies the correct action $\pi(s) \in A$ to perform in each state $s \in S$, which maximizes the sum of reward. In other words, we would like our agent to learn a function that enables it to map S to A ($\pi : S \leftarrow A$).

4.2. *Environment, States, Actions, and Rewards*

- *Environment*: Our environment is represented by the geographical map as a matrix of size $L \times M$ (L and M depend on the size of the geographic map) of small squares, by default of $20 \times 20m^2$ (as shown in Fig. 2). Each square has an actual state out of five, which are numbered as follows: "0": road zone could be elected as rebroadcast zone, "1" : non-road zone or zone X which could not be elected as rebroadcast zone, "0.5': gray/white zone, "1.5" : gray/white zone covered by a rebroadcast zone elected, "2": rebroadcast zone elected.
- *State Space S*: The state space contains all possible positions and numbers of rebroadcast points. At iteration t, the computed placement s_t is $s_t = \{N, (x_1, y_1), ..., (x_N, y_N)\}$, where, N is the number of rebroadcast points and (x_i, y_i) is the coordinate of the i^{th} rebroadcast point;

- *Action Space A* : The number of rebroadcast zones can be incremented by adding ($A(x_j, y_j)$) a new rebroadcast zone $j(x_j, y_j)$, decremented, or maintained ($\overline{A}$). The position of each rebroadcast zone in a current state can be deleted (DL), maintained (M), or moved up (U), down (D), right (R), or left (L). The number and positions of rebroadcast points are updated in each iteration by performing one of the following actions.
$$s_{t+1} \xleftarrow{N \times a^i \in \{DL,M,U,D,R,L\} \cup a \in \{A(x_j,y_j),\overline{A}\}} s_t$$
We have added a constraint to avoid collisions between two adjacent rebroadcast points. The distance between two rebroadcast zones must be greater than or equal to a threshold φ that depends on vehicles' average coverage in the rebroadcast zone.
- *Reward Function*: For state $s_t = \{\cup_{i=1}^{N}(x_i, y_i)\}$ at iteration t, the reward r_t is the sum of rewards of all elected rebroadcast points: $r_t = \sum_{i=1}^{N} r_t^i$, where $r_t^i =$ is derived by counting the number of new gray squares (not yet covered) that can be reached by a vehicle sitting in rebroadcast point i.

4.3. Q-learning and Problem Formulation

Q-learning is model-free reinforcement learning which provides agents with the ability to learn how to act optimally in MDP domains by experiencing the consequences of their actions without requiring maps of these domains. Following the above system description, we can model the problem as a discrete-state MDP, where an agent (the network Controller) in a state s_t takes action $a_t \in A$ and transitions to another state s_{t+1}. As a result of the execution of this action, the environment returns a rebroadcast zone's position dependant reward r_t, which allows the local update of a Q-value, $Q(s_t, a_t)$, indicating the appropriateness of selecting action a_t in-state s_t. The Q-value is computed according to the rule [13]:
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (1)$$
Where α quantifies to what extent the newly acquired information will override the old information. An agent with $\alpha = 0$ will learn nothing, while $\alpha = 1$ would consider only the most recent information, and $\gamma \in [0, 1]$ is the discount factor that determines the current value of the future state costs.
Authors in [3] proved that Q-learning converges to the optimum action-values with probability "1" as long as all actions are repeatedly sampled in all states and the action-value pairs are represented discreetly. First, the algorithm randomly selects N rebroadcast points. Then, the greedy policy regarding the Q-values tries to exploit continuously. During the learning phase, the agent selects the corresponding action based on the ϵ-greedy policy, i.e., it selects with probability $1 - \epsilon$ the action associated with the maximum Q-value and with probability ϵ a random action less frequently ($\epsilon = 0.1$). This means that the controller uses the optimum Q_value 90% of the time and makes exploratory actions 10% to gain new experience. This balancing between exploitation and exploration can guarantee convergence and often provides good performance. Hence, the controller explores all possible actions and avoids local minima. For more details on RL and Q-learning, the reader is referred to, e.g., [13].

4.4. Q-learning Algorithm

The Q-Learning algorithm is described in Algorithm 1. The set of gray squares is taken as input as we assume that the controller has a prior and updated vision of the quality of links in each road segment. This can be achieved thanks to some wireless site surveying, simulations, or some prediction as in [2]. The first step of the algorithm randomly selects N feasible rebroadcast points. After that, at each iteration t, the position of each rebroadcast zone ($(x_i, y_i), \forall i \in N$) and then the number of rebroadcast zones make an exploratory move with probability ϵ or picks the best-known action to date (highest Q_value) with probability $1 - \epsilon$. The algorithm explores different states during the learning phase (a fixed simulation/iteration run) to find the optimal policy that maximizes the expected action-value function (Q_value), and, hence, the total coverage of gray/white zones. The distance between a new candidate and other elected rebroadcast zones is always checked before choosing and executing an action.

To conclude this section, compared to traditional approaches, our method exhibits the following advantages while using decent memory and computational resources. RL algorithms are applicable to environments where no prior information, assumptions, or requirements about the region considered are available (e.g., heuristic methods wait to receive triggers before taking a decision. Learning-based methods are used when there is a possibility of further optimizing the system based on learning from the history of the aforementioned triggers). The decision-making process

Algorithm 1 Q-learning-based Rebroadcast zone placement

Input:
$L \times M$ Environment Matrix
N: Initial number of rebroadcast zones
Output:
$L \times M$ Environment Matrix with optimal rebroadcast zones placement and gray zones covered

Initialize the first state with N positions randomly selected, $Q_0(s, a) = 0, \forall s \in S$ and $\forall a \in A$ at iteration $t = 0$
while *Learning* **do**
  Visit state $s_t = (\{\cup_{i=1}^{N}(x_i, y_i)\})$
  for $(x_i, y_i) \in s_t$ **do**
    Select an action a_t^i using ϵ-greedy rule
    Update The values of Environment Matrix elements Calculate the reward r_t^i
    Observe next state s_{t+1}^i
    Update the Q_value $Q(s_t^i, a_t^i)$ from (1)
  Select an action a (to increment or not N)
  if $a_t = A(x_j, y_j)$ *(add a new broadcast zone j)* **then**
    Update The values of Environment Matrix elements
    Calculate the reward r_t
    Observe next state s_{t+1}
    Update the Q_value $Q(s_t, a_t)$

of Q-placement is on-demand. Unlike traditional algorithms with fixed optimization levels, Q-learning lets one decide the optimization level. This is achieved by tuning the number of iterations the algorithm runs. It is a desirable feature because the controller can precisely decide how much computation power it commits to achieving a certain performance level. This flexibility is essential, especially in situations where the controller is time-constrained. Moreover, the Q-learning algorithm is fully compatible with a centralized architecture, e.g., the Q-learning algorithm itself can be regarded as an application running on the controller. All the information needed by Q-learning is collected during routine network status updates between the controller and vehicles [21].

5. Performance analysis

This section details the performance results of our algorithm and compares its performance to the one proposed in [1] using the location-based dissemination procedure described in 3. We describe hereafter the experimental setup, the placement method of [1] before diving into the performance analysis of our placement method taken alone and also combined with the location-based dissemination procedure.

5.1. Simulation Setup

The simulation environment is based on the microscopic road traffic simulator SUMO [10] coupled with the event-based network simulator NETSIM as described in [9]. An area of $2 \times 2km^2$ of an European-like city center (namely, Toulouse, France) using Open Street Maps (OSM) is considered. It exhibits an irregular road structure and the presence of large buildings affecting the quality of wireless transmissions. The vehicle density is varied between 30 to 500 vehicles. The maximum transmission range of each vehicle is set to $R_{max} = 250m$. The number of RSUs is varied from 2 to 8, as well as their position. Our algorithm is implemented using the Python language and is run on an Intel Corei5 2GHz and 8GB RAM system. The convergence time of all simulations is below 0.7s. Table 1 lists the parameters used in the evaluation.

Finally, gray/white zones in the considered map were identified by simulation as follows. RSUs are configured to broadcast alert messages every $100ms$ for $500s$. The average Message Delivery Ratio (MDR) is computed for each

Table 1: Simulation configuration parameters

Parameter	Value
Discount rate γ	0.9
Learning rate α	0.1
Epsilon ϵ	0.1
Simulation time	500 s
AM generation start	10 s
AM generation rate	10 packets/s
AM packet size	1024 bytes
Propagation model	Nakagami m = 3
d_{max}	16 m

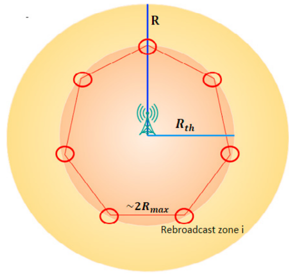

Fig. 3: Default rebroadcast zones placement method [1]

(a) Initial RSU placement (Scenario3) (b) with the default method [1] (c) with our Q-learning method

Fig. 4: rebroadcast points placement

square. As road safety applications require high reliability between 90% and 95% [11], squares with a MDR below 90% are considered gray.

5.2. *Rebroadcast Zones Placement Method proposed in [1]*

This method assumes that the controller can define a threshold distance R_{th} for each RSU from which the packet loss rate significantly increases. From this threshold, the controller builds a regular polygon with r equal sides ($5 \leq r \leq 17$), each with a length greater than $2R_{max} \pm 100m$, where R_{max} is the maximum transmission range of vehicles (as shown in figure 3). This ensures reduced interference between relay vehicles associated with two adjacent rebroadcast zones. Then, for each polygon vertices, the controller derives the closest point on a roadside that falls within a distance of d_{max}. If such a point exists, it is added to the set of rebroadcast points.

5.3. *Performance evaluation of our Q-learning placement method*

Three scenarios are considered with 2, 4, and 8 RSUs to address three different situations: an insufficient number of RSUs to decently cover the whole area (with numerous large gray zones), a decent number of RSUs, and finally, a high number of RSUs leading to multiple scattered small gray zones. The first scenario (S1) corresponds to the simulation settings of [1] with 2 RSUs. Our RL algorithm computes 11 rebroadcast points covering ≈ 84% of gray/white zones while with [1] 13 are obtained with a coverage of ≈ 80%. Full coverage is not achieved as 2 RSUs are not enough to reach all gray zones, assuming one rebroadcast from a relay vehicle.
The second scenario (S2) considers 4 RSUs placed at different locations. Our Q-learning algorithm leads to the full coverage of gray/white zones with only 9 rebroadcast zones, while 17 are needed with [1].

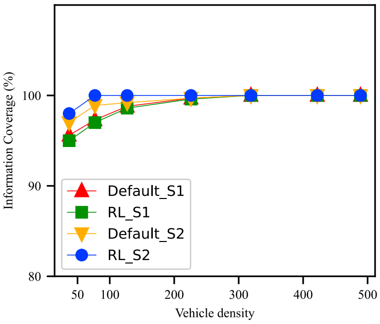

Fig. 5: Information Coverage

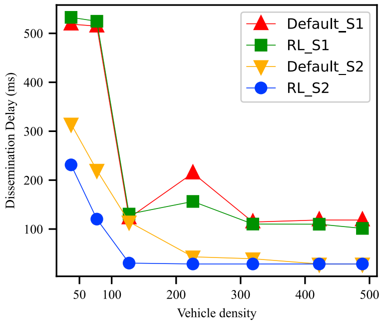

Fig. 6: Dissemination Delay

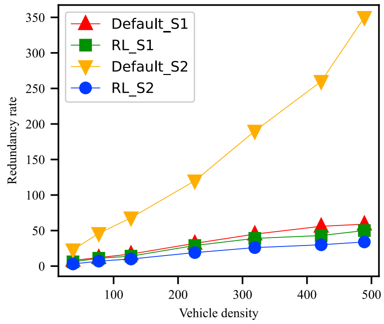

Fig. 7: Redundancy Ratio

As illustrated in Fig. 4a, the last scenario (S3) considers 8 RSUs. The transmission power (i.e., transmission range) of RSUs and vehicles is reduced. With the method of [1], we have 48 rebroadcast zones vs. 15 (Fig. 4c) with our algorithm, which achieves a better coverage than [1], around ≈ 93% of gray/white squares.

5.4. Performance Evaluation of the location based AM dissemination with our placement method

We present how our placement method positively impacts the location-based AM dissemination performance when used to select rebroadcast zones. We first present the considered performance metrics, then our analysis.

5.4.1. Performance metrics

- **Information Coverage (IC):** It is computed as the total number of vehicles that successfully receive an AM at the end of the simulation divided by the number of vehicles averaged on all generated AMs. This metric shows how successful the dissemination is after a decent period of time.
- **Dissemination delay:** The dissemination delay is the total time required to deliver the AM to all the vehicles in the area of interest that receive the AMs. The vehicles that do not receive any AM are excluded from the computation. This metric measures how fast the dissemination can reach the vehicles within the area of interest.
- **Collision Ratio (CR):** The collision ratio is the percentage of MAC collisions divided by the number of packets sent computed over the simulation duration.
- **Redundancy Rate (RR):** The average number of AM rebroadcasts out of all sourced AM.

5.4.2. Performance Analysis

Figure 5 presents the Information Coverage as a function of vehicle density in both scenarios (S1 and S2) for the default (i.e., [1]) and our RL-based placement methods. For S1, our method is slightly better, with an improvement that ranges between 0.2 to 0.5%. Notably, as noted above, this improvement is achieved with fewer rebroadcast zones (11 vs. 13 with the default method). As cited above, for S2, only 9 rebroadcast zones are needed with the RL algorithm vs. 17 with the default. Despite requiring approximately half the number of zones, Figure 5 shows a significantly better IC for low traffic densities. Compared to the default method, our proposed method ensures a much more effective placement of rebroadcast zones. This effective placement also affects the dissemination delays, as shown in Figure 6. For low vehicle densities, where the rebroadcast zone placement particularly matters, the difference between the dissemination delays of the two methods is at least 100ms, meaning that at least one additional rebroadcast (i.e., of the next instance of an AM) is needed with the default method compared to our method. When the traffic density increases, the probability of the presence of vehicles in or around the rebroadcast zones increases. As the RL-based placement ensures the full coverage of gray zones, more vehicles are reached from the first AM rebroadcast. Indeed, starting from a vehicle density of 100, with the RL-based placement, on average, all vehicles are reached within 100ms. In comparison, more than twice this density is needed to achieve such performance with the default method.

Figures 7 shows the number of duplicated AM transmissions versus vehicle density in both scenarios for both placement methods. As expected, with the RL-based placements, the redundancy is significantly reduced. Indeed, by minimizing the number of rebroadcast zones and optimally choosing their locations to serve the gray zones, fewer relay vehicles are eligible for an AM rebroadcast. This avoids useless redundant rebroadcasts.

Whatever the placement method, in all the considered scenarios, the collision ratio remains very low ($< 0.03\%$ vs $< 0.0001\%$ with the RL-based method), even for high vehicle densities. This is mainly due to the AM dissemination procedure, which drastically limits the contention when rebroadcasting an AM. Indeed, the relay vehicle selection limits the set of vehicles that can act as a relay to those in close vicinity to rebroadcast points. In addition, it further manages contention between nearby eligible relay vehicles by assigning different back-off waiting periods before pursuing with a rebroadcast attempt.

6. Conclusion

In this paper, we have proposed a Q-learning-based method that provides location-based AM dissemination procedures with the minimum number and optimal locations (rebroadcast zones) where vehicles are invited to rebroadcast an AM in order to deliver it on a pre-defined region, which may include multiple gray zones. Our method provides the best possible AM coverage, fast AM delivery, and very limited redundant and useless AM re-transmissions (i.e., network overhead). Our simulations assess the performance gains of our placement method on a real portion of a European city center with realistic road traffic models.

Future work will focus on assessing the ability of the proposed algorithm to effectively adapt to changes in wireless links' quality by dynamically adjusting the placement of rebroadcast zones.

References

[1] R. Chakroun, S. Abdellatif, T. Villemur, "An effective location-based alert messages dissemination scheme for software defined vehicular networks", in: International Conference on Advanced Information Networking and Applications, Springer, 2021, pp. 542–552.

[2] W. Benrhaiem, A. S. Hafid, "Bayesian networks based reliable broadcast in vehicular networks", Vehicular Communications 21 (2020) 100181.

[3] C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine Learning, vol. 8, no. 3, pp. 279–292, 1992.

[4] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network", Wireless networks 8 (2002) 153–167.

[5] A. Ullah, S. Yaqoob, M. Imran, H. Ning, "Emergency message dissemination schemes based on congestion avoidance in vanet and vehicular fog computing", IEEE Access 7 (2018) 1570–1585.

[6] J. A. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T.Calafate, "A survey and comparative study of broadcast warning message dissemination schemes for vanets", Mobile Information Systems 2016(2016).

[7] B. F. Ibrahim, M. Toycan, H. A. Mawlood, "A comprehensive survey on vanet broadcast protocols, in: 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)", IEEE,2020, pp. 298–302.

[8] B. Liu, D. Jia, J. Wang, K. Lu, L. Wu, "Cloud-assisted safety message dissemination in vanet–cellular heterogeneous wireless network", ieee systems journal 11 (2015) 128–139.

[9] J. S. Weber, M. Neves, T. Ferreto, Vanet simulators: "an updated review", Journal of the Brazilian Computer Society 27 (2021) 1–31.

[10] S. Busanelli, G. Ferrari, V. A. Giorgio, N. Iotti, "Information dissemination in urban vanets: Single-hop or multi-hop?, in: Roadside Networks for Vehicular Communications: Architectures, Applications, and TestFields", IGI Global, 2013, pp. 237–263.

[11] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, W. Xu, "Use cases, requirements, and design considerations for 5g v2x", arXiv preprint arXiv:1712.01754 (2017).

[12] R. Meireles, M. Boban, P. Steenkiste, O. Tonguz, J. Barros, "Experimental study on the impact of vehicular obstructions in vanets", in: 2010 IEEE Vehicular Networking Conference, IEEE, 2010, pp. 338–345.

[13] Richard S. Sutton and Andrew G. Barto. "Introduction to Reinforcement Learning". MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[14] G. Sato, A. Sakuraba, N. Uchida, Y. Shibata, "A new road state information platform based on crowed sensing on challenged network environments", Internet of Things (2020) 100214.

[15] V. G. Menon, S. Jacob, S. Joseph, P. Sehdev, M. R. Khosravi, F. Al-Turjman, "An iot-enabled intelligent automobile system for smart cities", Internet of Things (2020) 100213.

[16] B. Aslam, F. Amjad and C. C. Zou, "Optimal roadside units placement in urban areas for vehicular networks," 2012 IEEE Symposium on Computers and Communications (ISCC), 2012, pp. 000423-000429.

[17] E. S. Cavalcante, A. L. L. Aquino, G. L. Pappa, and A. A. F.Loureiro, "Roadside unit deployment for information dissemination in a VANET," in Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion '12, 2012, p. 27.

[18] S. Mehar, S. M. Senouci, A. Kies and M. M. Zoulikha, "An Optimized Roadside Units (RSU) placement for delay-sensitive applications in vehicular networks," 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), 2015, pp. 121-127.

[19] O. Trullols, J. M. Barcelo-Ordinas, M. Fiore, C. Casetti, and C.-F. Chiasserini, "A Max Coverage Formulation for Information Dissemination in Vehicular Networks," in 2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2009, pp. 154–160.

[20] B. Aslam and C. C. Zou, “Optimal roadside units placement along highways,” in 2011 IEEE Consumer Communications and Networking Conference (CCNC), 2011, pp. 814–815

[21] Zhang, Z., Ma, L., Leung, K. K., Tassiulas, L., Tucker, J. "Q-placement: Reinforcement-learning-based service placement in software-defined networks". In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), pp. 1527-1532. 2018.