A Transmission Design via Reinforcement Learning for Delay-Aware V2V Communications

Siyuan Yu, Nong Qu, Yizhong Zhang, Chao Wang$^{(\boxtimes)}$, and Fuqiang Liu

Department of Information and Communication Engineering,
Tongji University, Shanghai 201804, China
{1832914,1910624,1930707,chaowang,liufuqiang}@tongji.edu.cn

Abstract. We investigate machine-learning-based cross-layer energy-efficient transmission design for vehicular communication systems. A typical vehicle-to-vehicle (V2V) communication scenario is considered, in which the source intends to deliver two types of messages to the destination to support different safety-related applications. The first are periodically-generated heartbeat messages, and should be transmitted immediately with sufficient reliability. The second type are randomly-appeared sensing messages, and are expected to be transmitted with limited latency. Due to node mobility, accurate instantaneous channel knowledge at the transmitter side is hard to attain in practice. The transmit channel state information (CSIT) often exhibits certain delay. We propose a transmission strategy based on the deep reinforcement learning technique such that the unknown channel variation dynamics can be learned and transmission power and rate can be adaptive chosen according to the message delay status to achieve high energy efficiency. The advantages of our method over several conventional and heuristic approaches are demonstrated through computer simulations.

Keywords: Cross-layer transmission design · Vehicular communication · Deep reinforcement learning

1 Introduction

Traffic congestion, road safety, and energy shortage have become severe issues in the modern transportation system. Supported by the great progress made in the Internet of things (IoT), high-performance cloud/edge computing, and LTE/5G radio access technologies [10], intelligent transportation system (ITS) has been widely accepted as the promising solution and has attracted tremendous attentions in both academia and industry. As one of the key ITS technologies, vehicular networking enables traffic information, sensing data, and control demands to be shared, so that various ITS services can be developed [7]. However, such applications are in general safety-related and have diverse characteristics. Message transmissions have to be conducted among highly mobile terminals,

H. Gao et al. (Eds.): ChinaCom 2020, LNICST 352, pp. 613–627, 2021.
https://doi.org/10.1007/978-3-030-67720-6_42

subject to stringent delay and reliability requirements [6]. Realizing high-quality communication, especially vehicle-to-vehicle (V2V) transmission, is challenging.

Efficient transmission design in wireless systems has been extensively investigated for years. The conventional designing procedure is mainly based on channel state information (CSI) in the PHY layer. Although such an approach is sufficiently good for most modern wireless communication scenarios, it may not be able to satisfy the demands in vehicular communication systems due to a number of reasons. First, CSI does not reflect the time that source messages have already waited before satisfactory transmission opportunities are available. This may cause unbounded transmission delay. Involving the queue state information (QSI) in the MAC layer also into the decision-making process would enable *delay-aware transmission design*. Second, a sufficient amount of instantaneous CSI at the transmitter side is often assumed. Attaining such channel knowledge can be managed in relatively static wireless environments [14], but is challenging in dynamic ITS. Third, existing transmission design normally focuses on only one single type of application. In vehicular networks, however, multiple types of applications with diverse quality of service (QoS) requirements always co-exist. Therefore, new models and solutions are needed.

Our earlier work [4] investigates a multi-user V2V communication network in which each information source desires to send two types of delay-limited messages with different characteristics to its destination. We propose a cross-layer delay-aware transmission design using both CSI and QSI through the Lyapunov optimization theorem. It is shown that the performance can be much better than conventional CSI-based approaches. Nevertheless, the solution is established based on a knowledge of instantaneous transmit CSI (CSIT). In practice, a common approach to attaining CSIT is to exploit feedback from the receiver. In highly mobile networks, channel knowledge conveyed in the feedback normally can only reflect a delayed version of the true channel condition. In this case, transmission design based on instantaneous CSIT would not be applicable.

The past few years have witnessed explosive advances in artificial intelligence (AI) technologies. Utilizing machine learning techniques to facilitate wireless systems design has already attracted attentions in the communications research society [2,3]. As a main branch of machine learning, reinforcement learning, especially when combined with deep neural networks (DNNs), has been proven to be capable of solving a wide range of challenging sequential optimization problems (see, e.g., [5,8,11,12,15]). In this paper, we investigate the potential of applying reinforcement learning for enabling V2V transmission design.

Specifically, we consider a typical V2V transmission system, in which two types of messages with different QoS requirements are delivered to support safety-related applications. The first type are heartbeat messages which should be transmitted immediately with high reliability. The second type are environment sensing messages that should be sent with limited delay. The small-scale channel fading coefficients change across time-division slots following a fixed but unknown distribution. Only a delayed version of the CSIT is available. It is expected that the transmission can be realized with

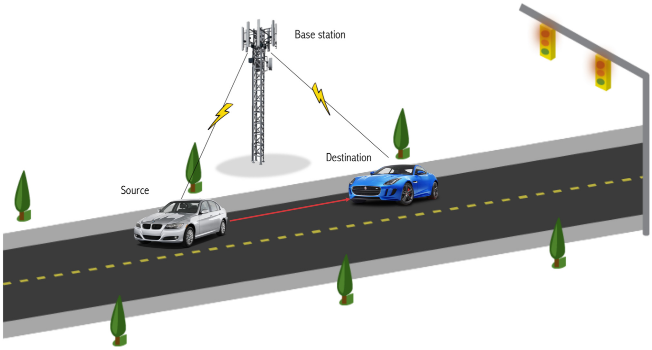

Fig. 1. System model

the maximal energy efficiency, i.e., one joule energy enables the maximum information delivery. We propose formulating the design optimization problem as a finite Markov decision process (MDP), and apply a deep Q-network (DQN) algorithm to solve it. Simulation results show that, our method is able to achieve close performance as using the method presented in [4] with perfect channel knowledge, and outperform three heuristic solutions in the imperfect CSI environment. The advantages of machine-learning-based cross-layer delay-aware transmission design are thus demonstrated.

2 System Model

We consider a typical V2V communication scenario with source S and destination D, as shown in Fig. 1. S desires to send two types of delay-limited messages to D. They have very different characteristics and QoS requirements, to support different safety-related ITS applications. The operations of the whole system are conducted in multiple time-division *transmission blocks*, each of which consists of T unit time slots. At the beginning of each time slot, S executes transmission. At the end of the slot, D provides certain feedback to update the knowledge of S regarding channel and message delivery status.

The first type of messages, termed *type-1 messages*, arrive in S periodically with a fixed data rate r bit/slot. An example of such messages is the heartbeat messages that provide D with the real-time status of S. They should be transmitted immediately. Otherwise the contained information would become stale. For each transmission block, a sufficient proportion of the messages (e.g., 70%) are expected to successfully reach D, as a transmission *reliability requirement*. Let $\phi[t] = 1$ denote that, at time slot t, S successfully delivers a type-1 message to D. Otherwise, $\phi[t] = 0$. The reliability requirement can hence be written as

$$\frac{1}{T}\sum_{t=1}^{T}\phi[t] \geq \phi_0, \tag{1}$$

where ϕ_0 is a constant specified according to the application.

The second type of messages, termed *type-2 messages*, arrive in S randomly. The arrival data volume $a[t]$ (in bits) at time slot t follows a stationary random process (e.g., Poisson with parameter λ bit/slot). An example of these messages is the environment sensing data collected from S's on-board sensors. Sharing them with D can help extend the environment perception capability of D. These messages can be temporarily stored in the source queue $\mathcal{Q}$. But the queuing delay must be limited. Such a *latency requirement* is posed by a finite maximum queue length Q_0 (queue length exceeding Q_0 results in overflow and loss of data). Let $Q[t]$ denote the instantaneous queue length at S and $b[t]$ denote the data volume that the type-2 messages leave the queue, at time slot t. Within each transmission block, at the end of time slot t, the queuing dynamics of $\mathcal{Q}$ under the latency requirement is

$$Q[t] = \max\{Q[t-1] - b[t], 0\} + a[t] \leq Q_0, \tag{2}$$

for all $t \in \{1, 2, \cdots, T\}$ and some $Q[0] \leq Q_0$.

The message transmissions are conducted in a narrow-band block fading environment. The fading coefficient between S and D at time slot t is denoted by $h[t]$, which remains fixed in each time slot, but changes across different slots. To simplify the problem, we assume that the channel gain $|h[t]|$ can be discretized into L different levels, the set of which is denoted by $\mathcal{H} = \{g_1, g_2, \cdots, g_L\}$. At time slot $t-1$, if $|h[t-1]| = g_i$, then at the next time slot t, the channel gain changes to $|h[t]| = g_j$ $(i, j \in \{1, 2, \cdots, L\})$ with transition probability $\Pr\{|h[t]| = g_j \big| |h[t-1]| = g_i\} = p_{i,j}$. We consider a stationary environment such that the channel transition probabilities remain fixed, but are unknown at both the source and destination. Due to the mobility of the vehicles, the source has only a delayed channel knowledge. Specifically, at time slot $t-1$, the destination estimates the channel coefficient using the training sequence sent by the source. Assume the estimation is sufficiently accurate. At the end of the slot, D feeds its estimation result (and also D's decoding status) to S. Hence at the beginning of time slot t, S knows only $h[t-1]$.

Based on available knowledge regarding channel and queue conditions, and the past transmission status of the two types of messages, at the beginning of any time slot t, S chooses to use one of M codebooks to encode its message to a unit-power signal $x[t]$. Let the set of data rates of the M codebooks be $\mathcal{R} = \{R_1, R_2, \cdots, R_M\}$. In addition, S selects a power level from set $\mathcal{P} = \{P_1, P_2, \cdots, P_N\}$ to send the signal. As a result, at time slot t, the received signal at D can be expressed as

$$y[t] = \sqrt{P[t]}h[t]x[t] + n[t], \tag{3}$$

where $P[t]$ is the transmit power, and $n[t]$ denotes additive white Gaussian noise (AWGN) with power N_0. The mutual information between S and D is thus (with bandwidth B)

$$I[t] = B\log_2\left(1 + \frac{P[t]|h[t]|^2}{N_0}\right). \tag{4}$$

Further, the source S can choose its *encoding action*, i.e., whether encoding only one type of messages or both types. Specifically, if $R[t] \geq r$, then S has two choices to form $x[t]$. First, both types of messages are transmitted. In this case, $x[t]$ represents r bits of a type-1 message and $R[t]-r$ bits of type-2 message (i.e., reducing the queue length by $b[t]=R[t]-r$ bits). Second, only the type-2 message is encoded so that $x[t]$ represents $R[t]$ bits of type-2 message (i.e., reducing the queue length by $b[t]=R[t]$ bits). Certainly, if the chosen data rate $R[t]<r$, there is only one encoding action: All the $R[t]$ bits in $x[t]$ are from the type-2 message, so that the queue length is reduced by $b[t]=R[t]$ bits. We use a binary indicator $\sigma[t]=1$ to denote the encoding action that S transmits both messages (including the case $b[t]=0$, i.e., only a type-1 message is encoded), and use $\sigma[t]=0$ to denote that S transmits only the type-2 message.

Due to the imperfect CSIT, the transmission of $x[t]$ may not be successful. Assume that the M channel codes adopted by S are sufficiently strong. Then if $R[t] \leq I[t]$, the destination D can correctly recover the transmitted source message. Otherwise, if $R[t]>I[t]$, correct decoding is not possible. Use binary indicator $\psi[t]$ to represent these events. At the end of each time slot t, the destination D feeds $\psi[t]=1$ (decoding success) or $\psi[t]=0$ (decoding failure) to S. (By this means, $\phi[t]$ in (1) can be found by $\phi[t]=\psi[t]\sigma[t]$.)

We aim to find an energy-efficient transmission strategy, such that the V2V link can choose its encoding action, transmission power and rate based on its queue state and delayed channel knowledge, to satisfy the desired message delivery requirements in each transmission block with maximized energy efficiency η, defined as the ratio of sum effective data rate to sum power consumption. Mathematically, we aim to solve the following optimization problem.

$$\text{maximize}: \quad \eta=\frac{\bar{R}}{\bar{P}}=\frac{\sum_{t=1}^{T}\psi[t]R[t]}{\sum_{t=1}^{T}P[t]} \tag{5}$$

$$\text{s.t.}: \quad (1) \text{ and } (2) \tag{6}$$

$$P[t]\in\mathcal{P}, R[t]\in\mathcal{R}, \sigma[t]\in\{0,1\}. \tag{7}$$

This stochastic optimization problem is hard to solve, especially due to unavailability of perfect knowledge regarding CSIT and environment dynamics. We propose to apply the deep reinforcement learning technique to fulfil the task.

3 MDP Formulation

To solve the energy-efficient transmission design problem for the considered delay-ware V2V communications, we first define a finite episodic MDP that reflects the optimization problem (5). Afterwards, a DQN algorithm is adopted to solve the MDP. Following the aforementioned transmission process, each episode represents one transmission block with T time slots. At the beginning of time slot t ($t \in \{1,2,\cdots,T\}$), the agent (the decision-maker, i.e., in our case, the source S) takes an action $\boldsymbol{a}[t]$, based on the environment state $\boldsymbol{s}[t-1]$ observed at the end of the time slot $t-1$. At the end of time slot t, the agent receives a reward $r[t]$ from the environment as a response to its action, and also observes the updated state $\boldsymbol{s}[t]$. The procedure continues until the completion of a block at time slot T. An MDP is specified by five elements: state space, action space, rewards, transition probabilities, and discount factor. We elaborate them as follows.

3.1 State Space

The state space of our MDP is defined as:

$$\mathcal{S} = \left\{\boldsymbol{s} | \boldsymbol{s} = \left[s^{\mathrm{m1}}, s^{\mathrm{m2}}, s^{\mathrm{ch}}, s^{\mathrm{bk}}\right]\right\}, \tag{8}$$

where s^{m1} represents the reliability state of type-1 messages, s^{m2} represents the queue state of type-2 messages, s^{ch} represents the channel state, and s^{bk} is the block state.

In particular, at the end of any time slot $t \in \{1, 2, \cdots, T\}$ (i.e., after taking action $\boldsymbol{a}[t]$), the value s^{m1} reflects how much the reliability requirement (1) has been satisfied and is defined as the total number of type-1 messages that have been successfully delivered so far:

$$s^{\mathrm{m1}}[t] = \sum_{i=1}^{t} \phi[i] = s^{\mathrm{m1}}[t-1] + \phi[t], \tag{9}$$

with initial value $s^{\mathrm{m1}}[0] = 0$.

The value s^{m2} reflects how much the latency requirement (2) has been satisfied and is set as the current queue length:

$$s^{\mathrm{m2}}[t] = Q[t] = \max\{s^{\mathrm{m2}}[t-1] - b[t], 0\} + a[t]. \tag{10}$$

$s^{\mathrm{m2}}[0]$ can be any value in set $\{0, 1, \cdots, Q_0\}$, since there may be data waiting in the queue before a new block starts.

The value s^{ch} reflects the channel quality:

$$s^{\mathrm{ch}}[t] = |h[t-1]|. \tag{11}$$

The initial state $s^{\mathrm{ch}}[0]$ can be chosen arbitrarily from $\mathcal{H}$. Clearly, for transmission decision-making at time slot t, $s^{\mathrm{ch}}[t]$ represents delayed CSIT.

Finally, the value s^{bk} is set as the number of remaining time slots in the block:

$$s^{\mathrm{bk}}[t] = T - t = s^{\mathrm{bk}}[t-1] - 1, \tag{12}$$

with initial value $s^{\mathrm{bk}}[0] = T$. The block state $s^{\mathrm{bk}}[t]$ represents the urgency level of taking actions to guarantee the QoS requirements of the two types of messages while achieving the maximum energy efficiency upon completion of the block.

The size of the state space can be extremely large, since s^{m2} is unbounded. We notice that if any $s^{\mathrm{m2}}[t] > Q_0$, the latency constraint (2) is violated and overflow occurs. Carrying on transmissions in the block does not help and thus such a situation should be avoided. This is also the case when $\frac{1}{T}\left(s^{\mathrm{m1}}[t]+s^{\mathrm{bk}}[t]\right)<\phi_0$, since the reliability constraint (1) cannot be satisfied regardless of the remaining actions. Therefore, to reduce the size of state space and enable efficient training, we define an extra "abnormal terminal state" $\boldsymbol{s}^+$. At the end of any time slot t, if either $s^{\mathrm{m2}}[t]>Q_0$ or $\frac{1}{T}\left(s^{\mathrm{m1}}[t]+s^{\mathrm{bk}}[t]\right)<\phi_0$ occurs, the state enters $\boldsymbol{s}^+$ and the episode ends. By this means, the size of the state space is limited to $\frac{1}{2}\left(T-T\cdot\phi_0+1\right)\times\left(T+T\cdot\phi_0+1\right)\times|H|\times\left(Q_0+1\right)$. In fact, it is possible to further decrease the state space by (possibly non-linearly) quantizing the ranges of s^{m1} (i.e., $[0,T]$), s^{m2} (i.e., $[0,Q_0]$), s^{bk} (i.e., $[0,T]$) into K_1, K_2, and K_3 sub-regions, respectively. Tuning the parameters K_1, K_2, K_3 leads to trade-off between performance and training complexity.

3.2 Action Space

The action space of our MDP is defined as

$$\mathcal{A}=\left\{\boldsymbol{a}|\boldsymbol{a}=\left[a^{\mathrm{pw}},a^{\mathrm{rt}},a^{\mathrm{m1}},a^{\mathrm{m2}}\right]\right\}. \tag{13}$$

The values a^{pw} and a^{rt} represent the power level and data rate the agent chooses from $\mathcal{P}$ and $\mathcal{R}$ respectively to send signal, i.e., at the beginning of time slot t the source selects $a^{\mathrm{pw}}[t]=P[t]\in\mathcal{P}$, and $a^{\mathrm{rt}}[t]=R[t]\in\mathcal{R}$.

a^{m1} and a^{m2} are binary indicators that reflect the encoding action. Specifically, when a^{rt} is chosen to be $a^{\mathrm{rt}}[t]>r$, setting $a^{\mathrm{m1}}[t]=1$ and $a^{\mathrm{m2}}[t]=1$ means that both type-1 (with rate r) and type-2 (with rate $a^{\mathrm{rt}}[t]-r$) messages are sent (equivalent to $\sigma[t]=1$). If only type-2 messages are encoded (with rate $a^{\mathrm{rt}}[t]$), one has $a^{\mathrm{m1}}[t]=0$ and $a^{\mathrm{m2}}[t]=1$ (equivalent to $\sigma[t]=0$). In addition, if $a^{\mathrm{rt}}[t]=r$, the source either transmits the type-1 message (indicated by $a^{\mathrm{m1}}[t]=1$ and $a^{\mathrm{m2}}[t]=0$, equivalent to $\sigma[t]=1$) or the type-2 message (indicated by $a^{\mathrm{m1}}[t]=0$ and $a^{\mathrm{m2}}[t]=1$, equivalent to $\sigma[t]=0$). Finally, if $a^{\mathrm{rt}}[t]<r$, only the type-2 message can be encoded and one has $a^{\mathrm{m1}}[t]=0$ and $a^{\mathrm{m2}}[t]=1$ (equivalent to $\sigma[t]=0$). Therefore, the size of action space is less than $2MN$.

3.3 Rewards

The rewards reflect how good the chosen actions are in terms of achieving the optimization objective (5) while guaranteeing the constraints in (6). It is the basis for the agent to learn a good policy. To this end, we form the reward $\mathfrak{R}$ by four parts: energy efficiency reward $\mathfrak{R}^{\mathrm{ee}}$, reliability reward of type-1 messages $\mathfrak{R}^{\mathrm{m1}}$, latency reward of type-2 messages $\mathfrak{R}^{\mathrm{m2}}$, and penalty for entering the abnormal termal state $\mathfrak{R}^{\mathrm{ab}}$.

Specifically, at the end of time slot t ($t\in\{1,2,\cdots,T\}$), the energy efficiency reward of taking action $\boldsymbol{a}[t]$ in state $\boldsymbol{s}[t-1]$ is defined as the incremental energy efficiency, i.e.,

$$\mathfrak{R}^{\mathrm{ee}}[t]=\frac{\sum_{i=1}^{t}\psi[i]a^{\mathrm{rt}}[i]}{\sum_{i=1}^{t}a^{\mathrm{pw}}[i]}-\mathfrak{R}^{\mathrm{ee}}[t-1], \tag{14}$$

where the initial value $\mathfrak{R}^{\text{ee}}[0] = 0$ and if for any t, $\sum_{i=1}^{t} a^{\text{pw}}[i] = 0$, we set $\mathfrak{R}^{\text{ee}}[t] = 0$. By this means, at the end of the transmission block, the accumulative reward is $\sum_{t=1}^{T} \mathfrak{R}^{\text{ee}}[t] = \frac{\sum_{i=1}^{T} \psi[i] a^{\text{rt}}[i]}{\sum_{i=1}^{T} a^{\text{pw}}[i]} = \frac{\sum_{i=1}^{T} \psi[i] R[i]}{\sum_{i=1}^{T} P[i]}$, which is the achievable energy efficiency. Maximizing $\sum_{t=1}^{T} \mathfrak{R}^{\text{ee}}[t]$ is the same as maximizing the original objective function (5).

The reliability reward of type-1 messages is defined as the incremental successful transmission ratio of the messages, i.e.,

$$\mathfrak{R}^{\text{m1}}[t] = \frac{\sum_{i=1}^{t} \psi[i] a^{\text{m1}}[i]}{T} - \mathfrak{R}^{\text{m1}}[t-1], \tag{15}$$

in which the initial value $\mathfrak{R}^{\text{m1}}[0] = 0$. The accumulative reward after T time slots is $\sum_{t=1}^{T} \mathfrak{R}^{\text{m1}}[t] = \frac{\sum_{i=1}^{T} \psi[i] a^{\text{m1}}[i]}{T} = \frac{\sum_{i=1}^{T} \phi[i]}{T}$. Maximizing this value is the same as maximizing the actual ratio of delivering type-1 messages defined in (1).

The latency reward of type-2 messages is defined as the reduction of the queue length (as a fraction of Q_0) resulted from taking action $\boldsymbol{a}[t]$:

$$\mathfrak{R}^{\text{m2}}[t] = -\left(\frac{s^{\text{m2}}[t]}{Q_0} - \frac{s^{\text{m2}}[t-1]}{Q_0} \right). \tag{16}$$

Maximizing the accumulative reward at the end of the block $\sum_{t=1}^{T} \mathfrak{R}^{\text{m2}}[t] = -\frac{Q[T]}{Q_0}$ minimizes the final queue length.

Finally, a penalty is imposed if at the end of any time slot t the system enters the abnormal terminal state, since the transmission requirements are violated. This penalty is set to

$$\mathfrak{R}^{\text{ab}}[t] = \begin{cases} -\Gamma & \text{for } \boldsymbol{s}^{+} \\ 0 & \text{otherwise} \end{cases}, \tag{17}$$

where Γ is a large positive number.

The total reward of our MDP at time slot t is set to

$$\mathfrak{R}[t] = \gamma_1 \mathfrak{R}^{\text{ee}}[t] + \gamma_2 \mathfrak{R}^{\text{m1}}[t] + \gamma_3 \mathfrak{R}^{\text{m2}}[t] + \mathfrak{R}^{\text{ab}}[t], \tag{18}$$

where γ_1, γ_2, and γ_3 are weighting parameters with $\gamma_1 + \gamma_2 + \gamma_3 = 1$. We can tune these parameters to specify our preference in maximizing the achievable energy efficiency, maximizing successful transmission ratio of the type-1 messages, or minimizing the queue length at the end of each transmission block. The special case of setting $\gamma_2 = \gamma_3 = 0$ leads to the original optimization problem (5).

3.4 Transition Probabilities and Discount Factor

The probability that state $\boldsymbol{s}$ enters state $\boldsymbol{s}'$, providing reward $\mathfrak{R}$, after the agent takes action $\boldsymbol{a}$, is:

$$p^{\boldsymbol{a}}_{\boldsymbol{s},\boldsymbol{s}',\mathfrak{R}} = \Pr\left\{ \boldsymbol{s}', \mathfrak{R} \middle| \boldsymbol{s}, \boldsymbol{a} \right\}, \quad \forall \boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}. \tag{19}$$

Certainly such transition probabilities are unknown. since the channel transmission probabilities are not available. The MDP has to be solved by a model-free algorithm.

Since our MDP is episodic and we consider the rewards attained in different time slots to be equally important, the discount factor is set to be 1. The total return is thus

$$G = \sum_{t=1}^{T} \mathfrak{R}[t]. \tag{20}$$

Solving the MDP is to find the strategy that maximizes G.

4 Deep Reinforcement Learning Algorithm

We apply DQN, which is a model-free reinforcement learning method that combines the Q-learning algorithm with DNNs, to solve the finite MDP presented in the above section.

Q-learning is a tabular off-policy reinforcement learning algorithm that establishes a Q-table to infer the optimal policy, i.e., the optimal mapping from states to the probability distribution over actions. Each element in the Q-table $\tilde{Q}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ represents the value of an action-value function $q(\boldsymbol{s}_t, \boldsymbol{a}_t)$, the expected return of taking action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$, and then following policy π:

$$q\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) = E_{\pi}\left[G_t = \sum_{i=t+1}^{T} \mathfrak{R}_i \middle| \boldsymbol{s}_t, \boldsymbol{a}_t \right]. \tag{21}$$

During the training process, for each sampled training episode, Q-learning keeps updating the Q-table according to

$$\tilde{Q}\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) \leftarrow \tilde{Q}\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) + \alpha\big(\mathfrak{R}_t + \gamma \max_{\boldsymbol{a} \in \mathcal{A}} \tilde{Q}\left(\boldsymbol{s}_{t+1}, \boldsymbol{a}\right) - \tilde{Q}\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right)\big), \tag{22}$$

in which action $\boldsymbol{a}_t$ is sampled from an ϵ-greedy policy according to the Q-table, $\mathfrak{R}_t$ and $\boldsymbol{s}_{t+1}$ are respectively the observed reward and new state, γ is discount factor, and $\alpha \in [0, 1]$ represents learning rate. Upon convergence, the optimal policy can then be derived from the Q-table.

The issue with conventional Q-learning is that when the state space and/or action space are large, the storage and update of Q-table demand large memory space and long convergence time in the training process [9]. DQN targets this problem by applying DNNs to approximate the action-value functions. However, this may further cause convergence and stability problems [12]. First, small updates to the action-value function $q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ may change the on-going policy significantly, which may mislead the agent to prefer a set of actions with correlated data. To handle this issue, reference [13] proposes the target network technique, which implements two networks with the same architecture but different network weight updating frequencies.

In addition, the input data of the DNNs are often highly correlated due to the ϵ-greedy sampling policy. The environment transfers from each state to a

Table 1. Simulation parameters

Parameter	Value
Type-1 message reliability requirement ϕ_0	$\{0.6, 0.7, 0.8\}$
Type-2 message queue length limit Q_0	$\{10, 8\}$
Transmit power levels $\mathcal{P}$	$\{0, 1, 2, 4, 8, 16, 32\}$
Transmit rates $\mathcal{R}$	$\{0, 1, 2, 3\}$
Abnormal terminal state penalty $-\varGamma$	-10
Initial states of the two messages $(s^{\mathrm{m1}}[0], s^{\mathrm{m2}}[0])$	$(0, 0)$

certain set of next states with high probability. Hence gradient decent may be conducted frequently based on similar and correlated inputs. Such a fact can cause strictly sub-optimal or unstable training results. Experience replay [12] stores the agent's experiences at each time slot in a data-set. In each episode, a batch containing fixed pieces of experience is selected randomly. The correlation between experience is broken.

The complete algorithm that we apply to solve our MDP is constructed following that in [12]. In the next section, we implement it, using Tensorflow [?], on an example problem. Our experiments apply DNNs with 5 layers. The input and output layers consist of 4 and 35 neurons respectively, taking the state at each time slot and the corresponding action-value of each available action as inputs and outputs. Each of the three hidden layers has 20 neurons. The learning rate is chosen to be $\alpha = 0.005$ and the mini-batch size is 300. The optimization method is Adam [?].

5 Performance Evaluation

We use computer simulation experiments to demonstrate the effectiveness of our method (termed DQN scheme). In the simulations, the block size is set to $T = 10$ slots. Both bandwidth and noise power are normalized to be $B = 1$ and $N_0 = 1$. The two types of messages respectively have fixed and Poisson distributed rates with $r = 1$ and $E[a[t]] = 0.6$ bit/slot. The channel gains are assumed to be $\mathcal{H} = \{0.359, 0.644, 0.866, 1.073, 1.286, 1.525, 1.830, 2.355\}$, approximately corresponding to the $\frac{1}{16}$, $\frac{3}{16}$, $\cdots$, $\frac{15}{16}$ percentiles of a standard Rayleigh distribution respectively. The channel transition probabilities are set to $p_{i,i} = 0.5\ \forall i \in \{1, \cdots, 8\}$, $p_{i,i\pm1} = 0.2\ \forall i \in \{2, \cdots, 7\}$, $p_{i,i\pm2} = 0.05\ \forall i \in \{3, \cdots, 6\}$, $p_{1,2} = p_{8,7} = 0.4$, $p_{1,3} = p_{8,6} = p_{2,4} = p_{7,5} = 0.1$. The remaining parameters described in Sects. 3 and 4 are displayed in Table 1. The size of the state space of our MDP is 3960 and for each state there are maximally 35 possible actions to choose. Enumerating all combinations of actions to find the optimal policy is very computationally expensive.

We compare our DQN scheme with four baseline methods. The first follows the scheme proposed in [4] and is termed Lyapunov optimization (LO) scheme.

Accurate instantaneous CSI at the source is assumed to be available. The Lyapunov optimization theorem is applied to transform the reliability and latency requirements (1) and (2) into a penalty term to the objective function. This allows the sequential optimization problem to be solved greedily in each time slot. The method is essentially for continuous power and rate allocation problems with large block length. To apply it in our problem, the parameters are first carefully selected to make sure that the penalty term is sufficiently large. The demanded reliability level of type-1 messages and queue length of type-2 messages within the limited block length are hence guaranteed. The power and rate pair in $\mathcal{P}$ and $\mathcal{R}$ that is closest to its solution is chosen to conduct the transmission.

The other three approaches do not assume accurate CSIT and carry out their transmissions in heuristic fashions. First, an FR (fixed-rate) scheme always transmits with rate $R[t] = 2$ using the smallest power level according to the delayed channel knowledge. If the reliability requirement (1) is not yet satisfied, both type-1 and type-2 messages are transmitted (each with rate 1 bit/slot). Otherwise, only type-2 messages are sent with rate 2 bit/slot. Although the transmission rate is chosen to be larger than the expected sum delivery rate of the two messages, 1.6 bit/slot, due to the lack of accurate CSIT, transmission errors may occur. This limits the transmission rate of the type-2 message to be small and thus may cause increasing queue length and then potential overflow.

The EQ (empty-queue) scheme targets addressing the above issue by always trying to empty the source queue. Specifically, if the reliability requirement (1) is not satisfied, both type-1 and type-2 messages are transmitted, with rates 1 bit/slot and $\min\{Q[t-1], 2\}$ bit/slot respectively, using the smallest power level derived according to the available channel knowledge. Otherwise, after sufficient type-1 messages are delivered successfully, only type-2 messages are sent with rate $\min\{Q[t-1], 3\}$ bit/slot. Compared with the FR scheme, the EQ scheme intends to reduce the possibility of overflow. But it may send messages with a high data rate and easily lead to unsuccessful transmission (due to channel outage). Further, the method does not naturally guarantee the requirements (1) and (2) to be satisfied, either.

To ensure the performance requirements of the two messages without accurate CSIT, a WS (worst-case scenario) scheme works similarly to the EQ scheme, but chooses its power level according to the worst-case scenario (the smallest channel gain that has non-zero transfer probability from the observed value). For example, if the delayed channel gain is 1.286, the power is chosen by assuming that the true value of the unknown channel gain is the worst case 0.866. Clearly, the scheme always satisfactorily delivers the two types of messages, at the cost of potentially a low achievable energy efficiency. For fair comparison, we generate 10^5 *test blocks* of channel gains and arrival rates of the type-2 messages. The average achievable energy efficiency and the number of successful transmission blocks of the five schemes are compared.

We choose the weighting parameters in (18) as $\gamma_1 = 1$ and $\gamma_2 = \gamma_3 = 0$, i.e., our DQN scheme aims to solve the original energy efficiency maximization

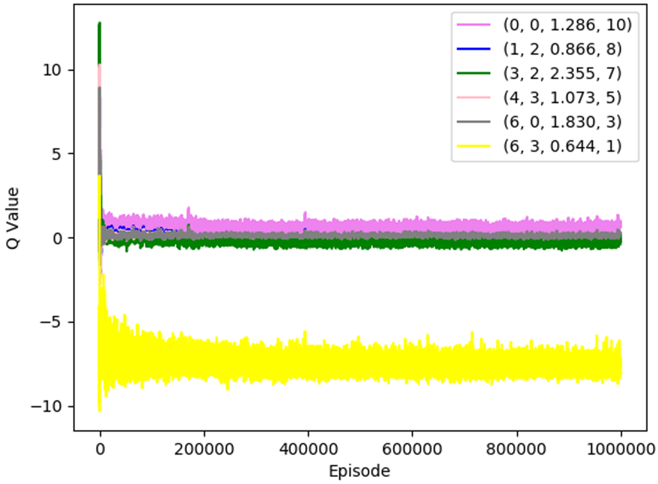

Fig. 2. Learning curve for six example states.

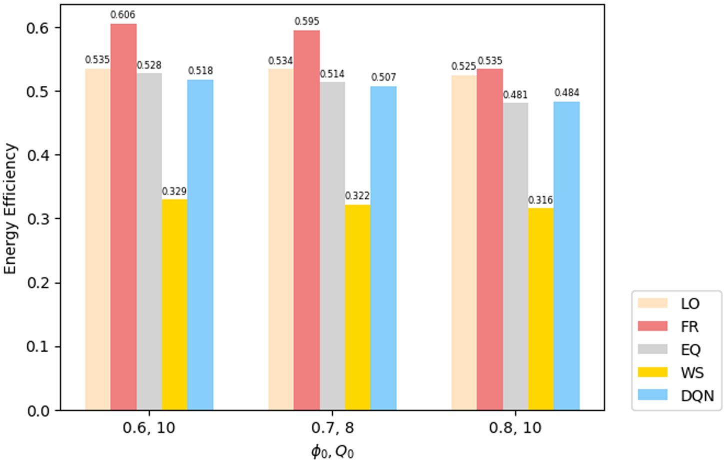

Fig. 3. Energy efficiency comparisons.

problem (5). The training of our algorithm is conducted using 10^6 episodes (blocks). We randomly select 6 out of the 3960 possible states. For each update of the DNN weights, these 6 states are input to the network and each generates 35 outputs (i.e., approximated action-value functions). The average, over all 35 actions, of the outputs associated with these states are displayed in Fig. 2. From the figure it is seen that our algorithm successfully converges. In what follows, we discuss the results obtained using the 10^5 test blocks.

Figure 3 illustrates the average achievable energy efficiency of the five schemes with three different sets of QoS requirements: (ϕ_0, Q_0) =

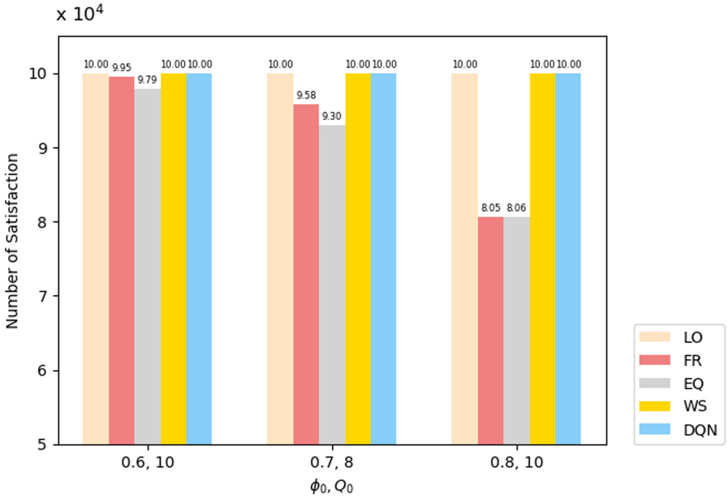

Fig. 4. Satisfaction of transmission constraints.

$(0.6, 10), (0.7, 8), (0.8, 10)$. If for any test block, a transmission scheme results in violation of the requirements (1) and (2), the energy efficiency attained by this scheme in such a failed block is treated as zero (and hence reduces the overall average test energy efficiency). The numbers of successful transmission blocks (i.e., the constraints (1) and (2) are both satisfied) are displayed in Fig. 4. It can be seen from Fig. 3 that the proposed DQN scheme obtains much better performance than the WS scheme, due to its ability of learning the environment changing dynamics and then finding a good solution for the sequential decision-making problem (5). There is no need to always prepare for the worst-case scenario and thus the energy efficiency can be significantly improved. The performance of the DQN scheme is comparable to the LO scheme, even though only delayed channel knowledge is available for making transmission decisions.

In addition, Fig. 3 shows that the FR scheme and EQ scheme may attain even higher energy efficiency. But one can see from Fig. 4 that both approaches lead to violation of the transmission requirements of the two types of messages. Therefore, their high achievable energy efficiency (derived using only the successful transmission blocks) does not serve as the evidence of their usefulness. Clearly, our DQN scheme provides satisfactory performance for all test blocks. Under different QoS requirements, its achievable energy efficiency also changes. This implies that policies are learned to be adaptive to different environments. This is different from the LO and WS schemes that always apply the same transmission strategy. The above observations clearly show the advantages of the proposed transmission design. Note that in our paper, we consider the energy efficiency defined in (5) as the objective for a delay-aware V2V communication system. One can follow the designing process and extend the system model to other scenarios with different performance metric, e.g., minimized power consumption. As mentioned earlier, changing the choices of γ_1, γ_2, and γ_3 in (18) also provides tradeoff between the overall designing objective and the performance of individual messages.

6 Conclusion

We have investigated applying machine learning to assist in cross-layer transmission design for delay-aware V2V systems. A typical scenario, in which multiple types of messages are transmitted between a vehicle source-destination pair with imperfect transmitter-side channel knowledge, has been considered. We have shown that by transforming an energy-efficiency maximization problem to a finite MDP, we can solve it efficiently through advanced reinforcement learning techniques, with achievable performance much better than several heuristic solutions. The advantages of combining machine learning with wireless communication design has been demonstrated. In this paper, we have focused on discretized state and action spaces. Systems with multiple source-destination pairs and continuous state/action spaces are currently under investigation.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant 61771343, and the Intelligent Connected Vehicle Pilot Demonstration Project under grant 2019B090912002.

References

1. Cao, B., Zhang, L., Li, Y., Feng, D., Cao, W.: Intelligent offloading in multi-access edge computing: a state-of-the-art review and framework. IEEE Commun. Mag. **57**(3), 56–62 (2019)
2. Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K., Hanzo, L.: Machine learning paradigms for next-generation wireless networks. IEEE Wirel. Commun. **24**(2), 98–105 (2016)
3. Zhang, C., Patras, P., Haddadi, H.: Deep learning in mobile and wireless networking: a survey. IEEE Commun. Surv. Tutor. **21**(3), 2224–2287 (2019)
4. Lan, D., Wang, C., Wang, P., Liu, F., Min, G.: Transmission design for energy-efficient vehicular networks with multiple delay-limited applications. In: 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, USA, December 2019
5. Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-learning. In: Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA, February 2016
6. Seo, H., Lee, K., Yasukawa, S., Peng, Y., Sartori, P.: LTE evolution for vehicle-to-everything services. IEEE Commun. Mag. **54**(6), 22–28 (2016)
7. Dar, K., Bakhouya, M., Gaber, J., Wack, M., Pascal, L.: Wireless communication technologies for ITS applications. IEEE Commun. Mag. **48**(5), 156–162 (2010)
8. Crites, R., Barto, A.: Improving elevator performance using reinforcement learning. In: Advances in Neural Information Processing Systems, pp. 1017–1023 (1996)

9. Sutton, R., Barto, A.: Introduction to Reinforcement Learning. MIT Press, Cambridge (2018)
10. Li, S., Xu, L., Zhao, S.: 5G internet of things: a survey. J. Ind. Inf. Integr. **10**, 1–9 (2018)
11. Lillicrap, T., et al.: Continuous control with deep reinforcement learning. CoRR abs/1509.02971 (2015)
12. Mnih, V., et al.: Playing Atari with deep reinforcement learning. NeurlIPS (2013)
13. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
14. Sun, W., Ström, E., Brännström, F., Sou, K., Sui, Y.: Radio resource management for D2D-based V2V communication. IEEE Trans. Veh. Technol. **65**(8), 6636–6650 (2015)
15. Zhan, W., Luo, C., Wang, J., Wang, C., Min, G., Duan, H., Zhu, Q.: Deep reinforcement learning-based offloading scheduling for vehicular edge computing. IEEE Internet Things J. **7**(6), 5449–5465 (2020)