Deep Reinforcement Learning for Resource Allocation in V2V Communications

Hao Ye and Geoffrey Ye Li

School of Electrical and Computer Engineering
Georgia Institute of Technology
Email: yehao@gatech.edu and liye@ece.gatech.edu

***Abstract*—In this article, we develop a decentralized resource allocation mechanism for vehicle-to-vehicle (V2V) communications based on deep reinforcement learning. Each V2V link is supported by an autonomous "agent", which makes its decisions to find the optimal sub-band and power level for transmission without requiring or having to wait for global information. Hence, the proposed method is decentralized, with minimum transmission overhead. From the simulation results, each agent can effectively learn how to satisfy the stringent latency constraints on V2V links while minimizing the interference to vehicle-to-infrastructure (V2I) communications.**

***Index Terms*—Deep Reinforcement Learning, V2V Communication, Resource Allocation**

I. Introduction

Vehicle-to-vehicle (V2V) communications have become an important technology for improving transportation services and road safety. The safety imperative makes the requirements for V2V communication links very stringent with ultra low end-to-end latency and high reliability, the end-to-end latency at the millisecond level and the reliability at nearly 100% [1]. The challenge has raised a lot of attention in both academia and industry. The Third Generation Partnership (3GPP) supports V2V services based on device-to-device (D2D) communications [2] as D2D shows superior performance in satisfying the quality-of-service (QoS) requirement in V2V applications.

In D2D communications, an effective resource allocation mechanism is needed to coordinate the mutual interference between the D2D links and the cellular users. A three-step approach has been proposed in [3] to control transmission power and allocate spectrum to maximize the system throughput with constraints on minimum signal-to-interference-plus-noise ratio (SINR) for both the cellular and the D2D links. In V2V communication networks, new challenges are brought about by high mobility vehicles, as high mobility means rapidly changing wireless channels. As a result, traditional methods of resource management for D2D communications with a full channel state information (CSI) assumption can no longer be applied in the V2V networks since it would be hard to track channel variations on such a short timescale.

There have been some interesting works on resource allocation for D2D-based V2V communications. Most of them are centralized, in which the central controller collects information of the network and makes decisions for each vehicle. With the global information of the networks, resource allocation can be formulated as an optimization problem, where the QoS requirements of V2V serve as constraints. However, these problems are usually NP-hard and therefore difficult to solve even with the global information of the networks. As a result, various simplified approaches have been proposed to decompose the problems into multiple steps so that local optimal or sub-optimal solutions can be found. In [4], the reliability and latency requirements of vehicular communications have been converted into optimization constraints, which are computable with only large-scale fading information and a heuristic approach has been developed to solve the resource management optimization problem. In [5], a resource allocation scheme has been designed only based on the slowly varying large-scale fading information and the sum V2I ergodic capacity is optimized with V2V reliability guaranteed.

Nevertheless, centralized control schemes will incur a large transmission overhead to get the global network information, as each vehicle needs to send the local channel state and interference information to the controller. In addition, the resource management of V2V communications should be autonomous so that it can still operate well when the infrastructure is not available. Decentralized resource allocation mechanisms in V2V communications are therefore of great importance. Recently, some decentralized resource allocation mechanisms for V2V communications have been developed. In [8], a distributed approach has been proposed to allocate sub-band to the V2V link by exploiting the position information. The V2V links are first clustered based on the positions of the vehicles and load similarities. The resource blocks (RBs) are then assigned to each group and the assignments are refined through iterative swap within each group rather than in the whole network. The low-complexity algorithm in [6] optimizes outage probabilities for V2V communications based on bipartite matching.

In the previous works, the QoS of V2V links are modelled as the reliability of SINR of V2V links and the latency constraint for V2V links has not been considered thoroughly since it is hard to formulate the latency constraints directly into the optimization problems. In order to address the problems

This work was supported in part by a research gift from Intel Corporation and the National Science Foundation under Grants 1443894 and 1731017.

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

that the traditional methods lack the ability to handle, we apply the multi-agent deep reinforcement learning scheme for resource allocation in V2V communications in this article. Reinforcement learning solves problems where each V2V link, as an agent, learns to make optimal decisions on spectrum and power for transmission based on the interacting with the environment. By optimizing strategies from the experience, the reward, which is a function of the capacity of the V2V and V2I links and the corresponding latency, is maximized in the long run.

Recently, deep learning has made great success in computer vision [9], speech recognition [10], and wireless communications [11]. With the help of deep learning techniques, reinforcement learning has shown impressive improvement in many applications, such as playing videos games [7] and Go games [12]. Deep reinforcement learning has also been applied to solve resource management problems. A deep reinforcement learning based approach has been proposed in [13] to address the problem of job scheduling with multiple resource demands in the computing clusters, where the objective is to minimize the average job slowdown and the reward function is based on the reciprocal duration of the job.

In our system, deep reinforcement learning is used to find the mapping between the local observations of each vehicle, including local channel state information and interference levels, and the resource allocation solution. Each V2V link is considered as an agent and the spectrum and transmission power are selected based on the observations of instantaneous channel conditions and exchanged information shared from the neighbors at each time slot. In general, the agents will automatically balance between minimizing the interference of V2V links to the V2I networks and meeting the requirements for the stringent latency constraints imposed on V2V link.

The main contribution of this article is using multi-agent deep reinforcement learning to develop a decentralized resource allocation mechanism for V2V communications, where the constraints on latency can be directly addressed. Based on the simulation results, deep reinforcement learning can learn to share the channel with V2I and other V2V links and generate the least interference to the V2I channels.

II. System Model

In this section, we will introduce the model of vehicle communication networks. As shown in Fig. 1, the vehicular networks consists of $\mathcal{M} = \{1, 2, ..., M\}$ cellular users (CUEs) demanding V2I links, which are orthogonally allocated spectrum bands and with high capacity communication links, and $\mathcal{K} = \{1, 2, ..., K\}$ pairs of D2D users (DUEs), which need V2V links to share information for traffic safety. In order to improve the spectrum utilization efficiency, orthogonally allocated uplink spectrum for V2I links is reused by the V2V links since uplink resources are less intensively used and the interference at the base station (BS) is more controllable.

The interference to the V2I links consists of two parts: the background noise and the signal from the V2V links sharing

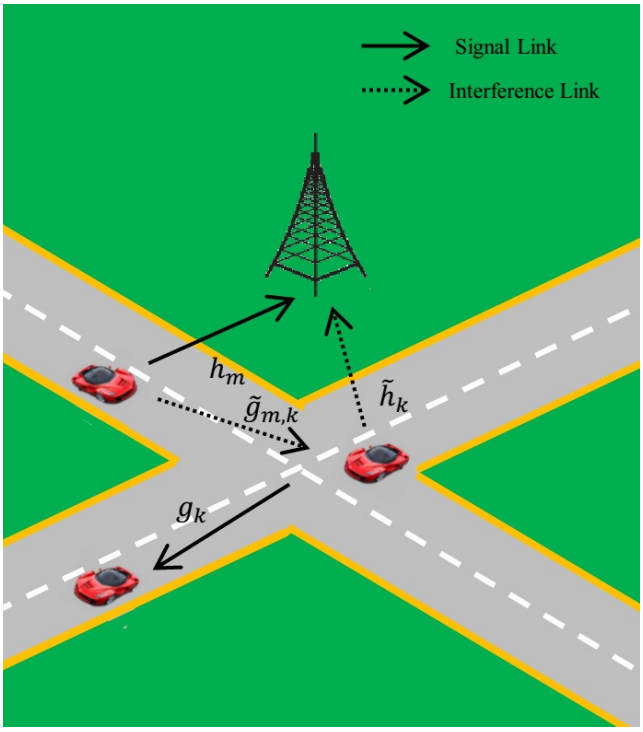

Fig. 1. An illustrative structure of vehicular communication networks.

the same sub-band. The SINR for the mth CUE will be

$$\gamma_m^c = \frac{P_m^c h_m}{\sigma^2 + \sum_{k\in\mathcal{K}} \rho_{m,k} P_k^d \tilde{h}_k}, \tag{1}$$

where P_m^c and P_k^d are the transmission powers of mth CUE and kth DUE, respectively, σ^2 is the noise power, h_m is the power gain of the channel corresponding to the mth CUE, $\tilde{h}_k$ is the interference power gain of the kth DUE, and $\rho_{m,k}$ is the spectrum allocation indicator with $\rho_{m,k} = 1$ if the kth DUE reuses the spectrum of the mth CUE and $\rho_{m,k} = 0$ otherwise. Hence the capacity of the mth CUE can be expressed as

$$C_m^c = W \cdot \log(1 + \gamma_m), \tag{2}$$

where W is the bandwidth.

Similarly, V2V links may share the same spectrum thus the SINR of the kth DUE can be expressed as

$$\gamma_k^d = \frac{P_k^d \cdot g_k}{\sigma^2 + G_c + G_d}, \tag{3}$$

with

$$G_c = \sum_{m\in\mathcal{M}} \rho_{m,k} P_m^c \tilde{g}_{m,k}, \tag{4}$$

and

$$G_d = \sum_{m\in\mathcal{M}} \sum_{k'\in\mathcal{K}k\neq k'} \rho_{m,k}\rho_{m,k'} P_{k'}^d \tilde{g}_{k',k}^d, \tag{5}$$

where g_k is the power gain of kth DUE, $\tilde{g}_{k,m}$ is the interference power gain of the mth CUE, and $\tilde{g}_{k',k}^d$ is the interference

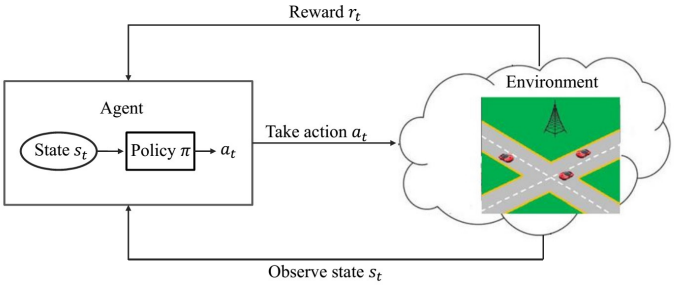

Fig. 2. Deep reinforcement learning for V2V communications

power gain of the k'th DUE. The capacity of the kth DUE can be expressed as

$$C_k^d = W \cdot \log(1 + \gamma_k^d). \tag{6}$$

Due to the essential role of V2V communications in vehicle security protection, there are stringent latency and reliability requirements for V2V links while the data rate is not of great importance. Traditionally, these constraints are handled by converting into the outage probabilities [4], [5]. In our method, the latency and reliability constraints are considered in the reward function directly, where a negative reward is given when the constraints are violated. In contrast to V2V safety communications, the latency requirement is less strict for the traditional cellular traffic. Therefore, traditional resource allocation focuses on maximizing the throughput under certain fairness considerations. In our system, the maximization of the V2I sum rate will be also reflected in the reward function in our method.

In the decentralized resource management scenario, the V2V links will select the RB based on local observations and the BS is assumed to have no information on the V2V links. Therefore, the resource allocation procedures of the V2I network should be independent of the resource management of V2V links. After the resource allocation procedures of V2I links, the main goal of the proposed autonomous scheme is to ensure that the latency constraints for each V2V link can be met while the interference of the V2V links to the V2I links should be minimized.

III. Deep Reinforcement Learning for Resource Allocation

In this section, the framework on deep reinforcement learning for resource allocation in V2V communications is introduced. The key parts in reinforcement learning framework are presented in detail and algorithms to train the deep Q-networks are shown as the proposed solution.

A. Reinforcement Learning

The structure of reinforcement learning for resource allocation in V2V communications is shown in Fig. 2, where an agent, corresponding to a V2V link, interacts with the environment. In this scenario, the environment is considered to be everything outside the V2V link. It should be noted that the behaviors of other V2V links cannot be controlled in the decentralized settings. As a result, their actions, such as selected spectrum, transmission power, etc., are treated as a part of the environment.

At each time t, the V2V link, as the agent, observes a state, s_t, from the state space, $\mathcal{S}$, and accordingly takes an action, a_t, from the action space, $\mathcal{A}$, selecting sub-band and transmission power based on the policy, π. The decision policy, π, is determined by a Q-function, $Q(s_t, a_t, \theta)$, where θ is the parameter of the Q-function and can be obtained by deep learning. Following the action, the state of the environment transits to a new state s_{t+1} and the agent receives a reward, r_t, determined by the capacities of the V2I and V2V links and the latency constraints of the corresponding V2V link. In our system, the state observed by each V2V link for characterizing the environment consists of several parts: the instant channel information of the corresponding V2V link, g_t, the previous interference power to the link, I_{t-1}, the channel information of the V2I link, e.g., from the V2V transmitter to the BS, h_t, the selected of sub-channel of neighbors in the previous time slot, N_{t-1}, the remaining load of the DUE to transmit, L_t , and the remaining time to meet the latency constraints U_t. Hence, the state can be expressed as $s_t = [g_t, I_{t-1}, h_t, N_{t-1}, L_t, U_t]$. The instant channel information and the interference received reveal the quality of each sub-band channel. The distribution of neighbors' selection relates to the interference to the other DUE users. The remaining amount of message to transmit and the remaining time contain information for selecting suitable power level.

At each time, the agent takes an action $a_t \in \mathcal{A}$, which includes selecting a sub-channel and a power level for transmission, according to the current state, $s_t \in \mathcal{S}$, based on the decision policy π. The transmission power is discretized into 3 levels, which leads to a $3 \times N_{RB}$ as the dimension of the action space if there are N_{RB} resource blocks.

The objective of V2V resource allocation is to minimize the interference to the V2I links with the latency constraints for V2V links guaranteed. In order to reach this objective, the frequency band and transmission power level selected by each V2V link should have small interference to all V2I links as well as other V2V links and it also needs to provide enough resources to meet the requirement of latency constraints. Therefore, the reward function is combined with three parts, the capacity of V2I links, the capacity of V2V links, and the latency condition. The sum capacities of V2I and V2V links are used to measure the interference to the V2I and other V2V links, respectively. The latency condition is represented as a penalty, which increases linearly as the remaining time U_t decreases. Therefore, the reward function can be expressed as,

$$r_t = \lambda_c \sum_{m \in \mathcal{M}} C_m^c + \lambda_d \sum_{k \in \mathcal{K}} C_k^d - \lambda_p (T_0 - U_t), \tag{7}$$

where T_0 is the constraint time and λ_c, λ_d, and λ_p are weights of the three parts, respectively.

The state transition and reward are stochastic and follow the Markov decision process (MDP), where the state transition probabilities and rewards depend only on the state of the environment and the action taken by the agent. The transition from s_t to s_{t+1} with reward r_t when action a_t is taken can be characterized by the conditional transition probability, $p(s_{t+1}, r_t|s_t, a_t)$. It should be noted that the agent can only control its own actions and has no prior knowledge on the transition probability matrix $\mathbf{P} = \{p(s_{t+1}, r_t|s_t, a_t)\}$, which is determined by the environment. In our problem, the transition on the channels, the interference, and the remaining messages to transmit are generated by the simulator of the wireless environment. The goal of reinforcement learning is to maximize the return defined as the expected cumulative discounted rewards,

$$G_t = \mathbb{E}[\sum_{n=0}^{\infty} \beta^n r_{t+n}], \tag{8}$$

where β is the discount factor.

B. Q-Learning

The agent takes actions based on a policy, π, which is a mapping from the state space, $\mathcal{S}$, to the action space, $\mathcal{A}$, and can be expressed as $\pi : s_t \in \mathcal{S} \rightarrow a_t \in \mathcal{A}$. As indicated before, the action, $a_t \in \mathcal{A}$, corresponds to how to select power level and spectrum for V2V links given a state s_t described above in our problem.

We use Q-learning to get an optimal policy for resource allocation in V2V communications to maximize the long-term expected accumulated discounted rewards [14]. The Q-value for a given state-action pair (s_t, a_t), $Q(s_t, a_t)$, of policy π is defined as the expected accumulated discounted rewards when taking an action $a_t \in \mathcal{A}$ and following policy π thereafter. Once Q-values, $Q(s_t, a_t)$, are given, an improved policy, π, can be easily constructed by taking the action,

$$a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a). \tag{9}$$

That is, the action is taken to maximize the long-term accumulated rewards.

The optimal policy with Q-values Q^* can be found without any knowledge of the system dynamics based on the following update equation,

$$\begin{aligned} Q_{new}(s_t, a_t) =& Q_{old}(s_t, a_t) + \alpha[r_{t+1} + \\ & \beta \max_{s \in \mathcal{S}} Q_{old}(s, a_t) - Q_{old}(s_t, a_t)], \end{aligned} \tag{10}$$

It has been shown that in the Markov decision process (MDP) case, the Q-values will converge with probability 1 to the optimal Q^* if each action in the action space is executed under each state for an infinite number of times on an infinite run and the learning rate α decays appropriately. The optimal policy, π^*, can be found once the optimal Q-value, Q^* , is determined.

Once the optimal policy is found through training, it can be used to select spectrum band and transmission power level for V2V links to maximize overall capacity and ensure the latency constraints of V2V links.

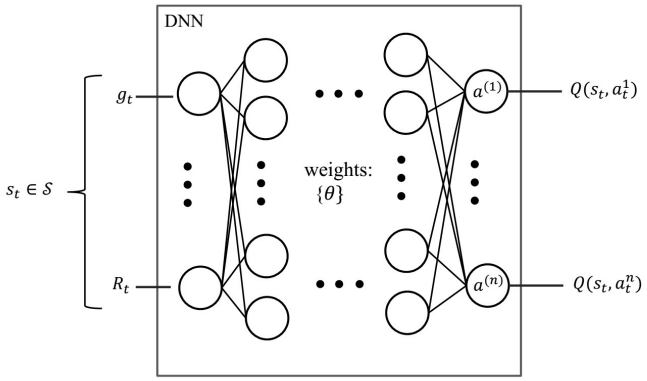

Fig. 3. Structure of Deep Q-networks

C. Deep Q Networks

Q-learning works well if the state and action spaces of the problem are small and a look-up table can be used to accomplish the update rule. However, it is impossible when the state-action space becomes very large. In this situation, many states may be rarely visited, thus the corresponding Q-values are seldom updated, leading to a much longer time to converge. Deep Q-network combines Q-learning with deep learning. The Q-function is approximated by a deep neural network as shown in Fig. 3. The basic idea behind deep Q-network is the use of a deep neural network (DNN) function approximator with weights $\{\theta\}$ as a Q-network [14]. Once $\{\theta\}$ is determined, Q-values, $Q(s_t, a_t)$, will be the outputs of the DNN in Fig. 3. DNN can address sophisticate mappings between the channel information and the desired output based on a large amount of training data, which will be used to determine Q-values.

The Q-network updates its weights, θ, at each iteration to minimize the following loss function derived from the same Q-network with old weights on a data set D,

$$Loss(\theta) = \sum_{(s_t, a_t) \in D} (y - Q(s_t, a_t, \theta))^2, \tag{11}$$

where

$$y = r_t + \max_{a \in \mathcal{A}} Q_{old}(s_t, a, \theta), \tag{12}$$

where r_t is the corresponding reward.

D. Training and Testing Algorithms

Like most machine learning algorithms, ours consists of two stages in our system, the training stage and the testing stage. The training and test data are generated from an environment simulator and the agents. Each sample includes s_t, s_{t+1}, a_t, and r_t. Our simulator consists of DUEs and CUEs and their channels, where the vehicles are randomly dropped and the channels for CUEs and DUEs are generated based on the positions of the vehicles. With the selected spectrum and power of V2V links, the simulator can provide s_{t+1} and r_t

to the agents. In the training stage, we follow the deep Q-learning with experience replay [14], where the generated data are saved in a storage called *memory*. As shown in Algorithm 1, the mini-batch data used for updating the Q-network is sampled from the *memory* in each iteration. In this way, the temporal correlation of data can be suppressed. The policy used in each V2V link for selecting spectrum and power is random at the beginning and is gradually improved with the updated Q-networks. As shown in Algorithm 2, in the test stage, the actions in V2V links are chosen with the maximum Q-value given by the trained Q-networks, based on which the evaluation is obtained.

Since each V2V link is considered as an agent, the actions of other V2V links are unknown if they update their actions simultaneously and independently. As a result, the states that each agent observes cannot fully characterize the whole environment. To mitigate this problem, the V2V links are set to be updated asynchronously. At each time slot, only one or a small proportional of V2V links will update their selections of actions. In this way, for each agent, the environmental changes due to other agents' actions can be observed.

Algorithm 1 Training Stage Procedure

1: **procedure** TRAINING
2: **Input**: Q-network structure, environment simulator.
3: **Output**: Q-network
4: **Start:**
Random initialize the policy π
Initialize the model
Start environment simulator, generate vehicles, V2V links, V2I links.
5: **Loop**:
Random sample V2V links in the system.
Generate a set of data based on policy π from the environment simulator.
Save the data item {state, reward, action, post-state} into memory.
Sample a mini-batch of data from the memory.
Train the deep Q-network using the mini-batch data.
Update the policy π: chose the action with maximum Q-value.
6: **End Loop**
7: **Return**: Return the deep Q-network

IV. SIMULATIONS

In this section, we present simulation results to demonstrate the performance of the proposed method. We consider a single cell outdoor system with the carrier frequency of 2 GHz. We follow the simulation setup for the Manhattan case detailed in 3GPP TR 36.885 [2], where there are 9 blocks in all and with both line-of-sight (LOS) and non-line-of-sight (NLOS) channels. The vehicles are dropped in the lane randomly according to the spatial Poisson process and each plans to communicate with the 3 nearby vehicles. Hence the number of V2V links, K, is 3 times of the number of vehicles. Our

Algorithm 2 Test Stage Procedure

1: **procedure** TESTING
2: **Input**: Q-network, environment simulator.
3: **Output**: Evaluation results
4: **Start:** Load the Q-network model
Start environment simulator, generate vehicles, V2V links, V2I links.
5: **Loop**:
Select a V2V link in the system.
Select the action by choosing the action with the largest Q-value.
Update the environment simulator based on the actions selected.
Update the evaluation results, i.e., the average of V2I capacity and the probability of successful DUEs.
6: **End Loop**
7: **Return**: Evaluation results

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Carrier frequency	2 GHz
Bandwidth	10 MHz
BS antenna height	25m
BS antenna gain	8dBi
BS receiver noise figure	5dB
Vehicle antenna height	1.5m
Vehicle antenna gain	3dBi
Vehicle receiver noise figure	9dB
Vehicle speed	36 km/h
Number of lanes	3 in each direction (12 in total)
Latency constraints for V2V links T_0	100 ms
V2V transmit power level list	[23, 10, 5] dBm
Noise power σ^2	-114 dBm
[λ_c,λ_d,λ_p]	[0.1, 0.9, 1]

deep Q-network is a five-layer fully connected neural network with three hidden layers. The numbers of neurons in the three hidden layers are 500, 250 and 120, respectively. The activation function of Relu is used, which is defined as

$$f_r(x) = \max(0, x). \tag{13}$$

The learning rate is 0.01 at the beginning and decreases exponentially. We also utilize ϵ-greedy policy to balance the exploration and exploitation [14] and adaptive moment estimation method (Adam) for training [15]. The detail parameters can be found in Table 1.

The proposed method is compared with other two methods. The first is a random resource allocation method. At each time, the agent randomly chooses a sub-band for transmission. The other method is developed in [8], where vehicles are first grouped by the similarities and then the sub-bands are allocated and adjusted iteratively to the V2V links in each group.

A. V2I Capacity

Fig. 4 shows the summation of V2I rate versus the number of vehicles. From the figure, the proposed method has a much

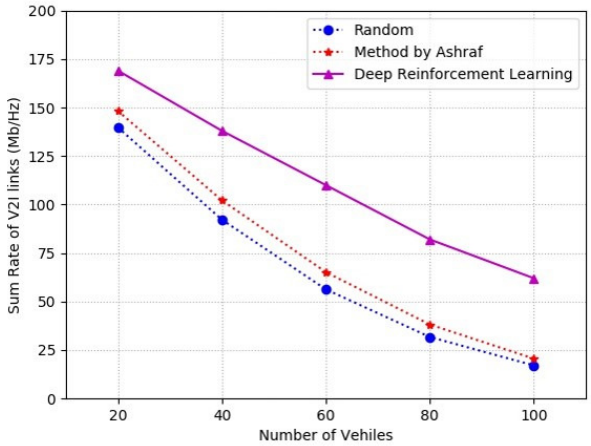

Fig. 4. Mean rate versus the number of vehicles.

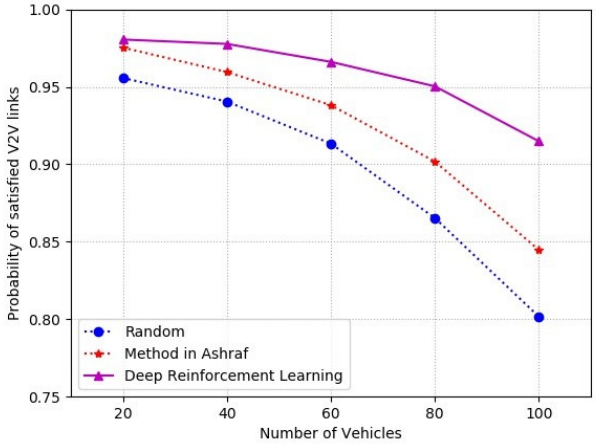

Fig. 5. Probability of satisfied V2V links versus the number vehicles.

better performance to mitigate the interference of V2V links to the V2I communications. Since the method in [8] maximizes the SINR in V2V links, rather than optimizing the V2I links directly, it has only a slightly better performance than the random method, much worse than the proposed method.

B. V2V Latency

Fig. 5 shows the probability that V2V links satisfy the latency constraint versus the number of vehicles. From the figure, the proposed method has a much larger probability for V2V links to satisfy the latency constraint since it can dynamically adjust the power and sub-band for transmission so that the links likely violating the latency constraint have more resources.

V. Conclusion

In this article, a decentralized resource allocation mechanism is proposed for the V2V communications based on deep reinforcement learning, where each V2V link is regarded as an agent, making its own decisions to find optimal spectrum and power for transmission. Since the proposed method is decentralized, the global information is not required for each agent to make its decisions, the transmission overhead is small. From the simulation results, each agent can learn how to satisfy the V2V constraints while minimizing the interference to V2I communications.

VI. Acknowledgment

We would like to thank Dr. Biing-Hwang Juang for helpful discussions and comments. We would also like to thank Dr. May Wu, Dr. Satish C. Jha, Dr. Kathiravetpillai Sivanesan, Dr. Lu Lu, and Dr. JoonBeom Kim from Intel Corporation for their insightful comments, which have substantially improved the quality of this paper.

References

[1] H. Seo, K. D. Lee, S. Yasukawa, Y. Peng, and P. Sartori. "LTE evolution for vehicle-to-everything services," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 22–28, Jun. 2016.

[2] *3rd Generation Partnership Project: Technical Specification Group Radio Access Network: Study LTE-Based V2X Services: (Release 14)*, Standard 3GPP TR 36.885 V2.0.0, Jun. 2016.

[3] D. Feng, L. Lu, Y. Yuan-Wu, G. Li, S. Li, and G. Feng, "Deviceto-device communications in cellular networks," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 49–55, Apr. 2014.

[4] W. Sun, E. G. Strom, F. Brannstrom, K. C. Sou, and Y. Sui, "Radio resource management for D2D-based V2V communication," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6636–6650, Aug. 2016.

[5] L. Liang, G. Y. Li, and W. Xu, "Resource allocation for D2D-enabled vehicular communications," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 3186–3197, Jul. 2017.

[6] B. Bai, W. Chen, K. B. Letaief, and Z. Cao, "Low complexity outage optimal distributed channel allocation for vehicle-to-vehicle communications," *IEEE J. Sel. Areas Commun*, vol. 29, no. 1, pp.161–172, Jan. 2011.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[8] M. I. Ashraf, M. Bennis, C. Perfecto, and W. Saad, "Dynamic proximity-aware resource allocation in Vehicle-to-Vehicle (V2V) communications, " in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2016, pp. 1–6.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097-1105.

[10] C. Weng, D. Yu, S. Watanabe, and B. H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. ICASSP*, May 2014, pp. 5532–5536.

[11] H. Ye, G. Y. Li, and B.-H. F. Juang. "Power of deep learning for channel estimation and signal detection in OFDM systems" *to appear in IEEE Wireless Commun. Lett.*, 2017.

[12] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershevlvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*,vol. 529, no.7587, pp. 484–489, Jan. 2016.

[13] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. of the 15th ACM Workshop on Hot Topics in Networks.* ACM, Nov. 2016, pp. 50–56.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. H. I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Human-level control through deep reinforcement learning," *Nature* vol. 518, no. 7540, pp. 529–533, Feb. 2015

[15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014