# Deep Reinforcement Learning-based Data Transmission for D2D Communications

Achraf Moussaid, Wael Jaafar, Wessam Ajib and Halima Elbiaze
Department of Computer Science, Université du Québec à Montréal
{jaafar.wael, ajib.wessam, elbiaze.halima}@uqam.ca

***Abstract*—Device-to-Device (D2D) communication has gained interest as a promising technology for next generation wireless networks. D2D communication promotes the use of point-to-point communications between users without going through the base stations. In this paper, we aim at maximizing the sum rate of a D2D network, under the assumption of realistic time-varying channels and D2D interference. Specifically, we formulate channels as Finite-State Markov Channels (FSMC). With realistic FSMC, the complexity of the problem is high. Consequently, we propose the use of a centralized Deep Reinforcement Learning (DRL) transmission scheme for D2D communications, where transmission decisions are taken by one agent that has a global knowledge of the D2D network. We compare the DRL-based scheme with other transmission schemes. The results show that it outperforms other approaches in terms of achieved sum rate.**

***Index Terms*—Device-to-Device (D2D), Deep Reinforcement Learning (DRL), Finite-State-Markov-Chain (FSMC).**

## I. Introduction

As the standardization of 5G requirements by 3GPP is advancing rapidly, many aspects of the Non Stand Alone (NSA) and Stand Alone (SA) 5G network are getting an increased attention. Among them, the Device-to-Device (D2D) communication, defined as the direct communication between users (UEs) without going through the Base Stations (BSs), is seen as a promising technology to improve frequency reuse and throughput, and reduce wireless transmissions delays [1]-[2]. D2D communications can be *Inband* or *Outband* [1]-[3]. *Inband* communications occur within the same spectrum as the cellular network. While *Outband* communications happen on unlicensed spectrum bands. On one hand, *Inband* communications can be *Underlay*, i.e. direct data transmissions between UEs can interfere on/be interfered by communications between other UEs and the BS. On the other hand, *Inband Overlay* transmissions do not interfere on/get interfered by multi-user communications between other UEs and the BS, as dedicated spectral resources are allocated to the D2D communications within the BS service range [1].

Recent research in D2D communication has focused on resource allocation (such as transmit power, frequency allocation) [4]-[5], interference analysis [6], scheduling [7]-[8] and data transmission [9]. Aiming at maximizing the end-to-end throughput between a D2D source and destination, and under interference constraints from the cellular network, the authors of [7] proposed a joint mode selection of user scheduling and rate adaptation policy. The solution considers inter-cell interference (ICI) and requires very limited feedback at the BS about interference between D2D and cellular UEs. Using optimal routing and power allocation algorithms and the sequential activation scheme (SLA), they showed that the achieved throughput by multi-hop D2D outperforms one-hop D2D. In [8], the authors investigated a D2D-assisted wireless caching network, where a D2D UE can obtain data (cached) from its own memory, or from a UE neighbor or from the BS. For the second case (data obtained from a UE neighbor), the authors formulated a joint D2D link scheduling and power allocation problem to maximize the system's throughput. Obtained throughput results outperform algorithms proposed in the state of the art. In [9], the authors studied the scenario of a group of D2D links sharing one dedicated *Inband Overlay* channel using Carrier-Sense Multiple Access (CSMA). The Ideal CSMA Network (ICN) model is used to investigate the D2D links behaviors under spatial reuse conditions. Assuming heterogeneous rate requirements, willingness to pay and selfishness of the D2D links, the latter aim at maximizing their target rates and own payoffs. To do so, the authors propose a Stackelberg game where the BS acts as the game's leader to regulate D2D link transmissions by updating the service price in order to maximize the sum throughput.

Even though work in [7]-[9] propose efficient scheduling for D2D wireless networks, they assume an invariant channel model (such as the path-loss model), that does not reflect a real wireless environment. Here, we assume that channels are realistically time-varying. The latter are modeled as Finite-State Markov Channels (FSMC), to characterize the correlation of fading processes [10]. The use of FSMC models may imply significant improvement over memoryless channel models. However, they generate a high complexity for the problem. Hence, we propose a DRL-based approach to tackle D2D communications coordination. DRL consists on the use of Deep-$Q$-Network (DQN) to approximate the value-action function in order to improve action selection, system control and D2D network updating [11].

In the proposed DRL-based scheme, we define for the agent: the state space (D2D channel states and potential D2D links), the action space (which D2D communications to allow simultaneously) and the reward function (achieved data rate). DRL is used to obtain the optimal D2D transmissions selection policy, that maximizes the achieved sum rate over time, with respect to some constraints.

The remaining of the paper is organized as follows. Section II presents the system model. Section III formulates the D2D

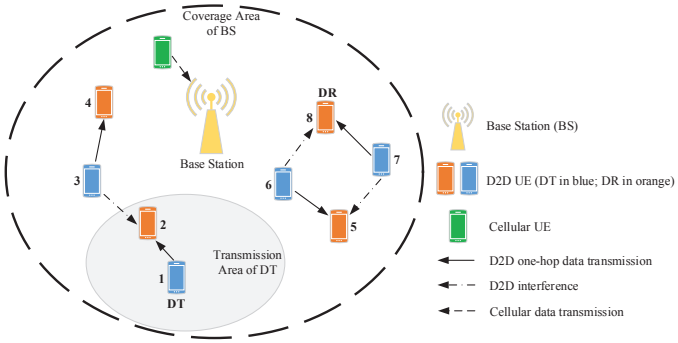


Fig. 1. System model with four concurrent D2D transmissions (solid arrows) and mutual interference (dashed arrows)

transmission network problem for the deep reinforcement learning approach. Section IV illustrates and discusses the simulation results. Finally, a conclusion closes the paper in section V.

## II. System Model

Similarly to [8], we assume a cellular network with one BS and $K$ D2D UEs, randomly spread within the coverage area of the BS, as presented in Fig. 1. D2D transmissions, as shown in Fig. 1 with solid arrows, are motivated by the fact that a D2D receiver (denoted $DR$) requests data stored (or cached) within a D2D transmitter (denoted $DT$). Any UE can be a $DT$ for another UE if *two conditions* are satisfied:

1) The two UEs are close to each other, i.e. the $DR$ is within a transmission range, defined by a distance $d_0$ where $d_0$ is the coverage radius of a $DT$ defined according to the Signal-to-Noise-Ratio (SNR) association rule, that means the received SNR at $DR$, in the absence of any interference, shall be above a fixed threshold $\gamma$ [12].
2) $DT$ has the data requested by $DR$. It should be noted that data requested by a $DR$ can be available at many $DTs$ at the same time. For simplicity, we assume that only one $DT$ can transmit the requested data.

Moreover, we assume that D2D transmissions share one dedicated channel in an *Inband Overlay* network. That means, no interference between D2D UEs and cellular UEs is occurring. However, interference within the D2D network is unavoidable. D2D UEs work on half-duplex mode, i.e. no simultaneous transmission and reception are possible at one UE. And, all transmissions are unicast, that means each transmission by a $DT$ has only one destination $DR$ [8].

At a given time $t$, and from the $K$ D2D UEs, we assume that $K_{DR}$ UEs form the set $\mathcal{S}_{DR}$ request data from neighboring UEs. If, for a $DR_i$ in $\mathcal{S}_{DR}$ ($DR_i$ is the $i^{th}$ $DR$ in set $\mathcal{S}_{DR}$, requesting data), a $DT_j$ in set $\mathcal{S}_{DT}$, of cardinality $K_{DT}$, respects the previous conditions 1 and 2 (i.e. $DT_j$ is the $j^{th}$ $DT$ in set $\mathcal{S}_{DT}$ that has $DR_i$'s data and satisfies $d(DT_j, DR_i) \leq d_0$, where $d(.,.)$ is the Euclidean distance function), then $l_{DT_j,DR_i}$ is a potential D2D link. It is clear that the effective set of $DR$s to be potentially served, denoted $\mathcal{S}'_{DR}$, is $\mathcal{S}'_{DR} \subset \mathcal{S}_{DR}$, since some $DR$s requests cannot find $DT$s satisfying both conditions 1 and 2. Hence, $K'_{DR} = |\mathcal{S}'_{DR}| \leq K_{DR} = |\mathcal{S}_{DR}|$, where $|.|$ is the cardinality, and the set $\mathcal{L}_{DT} = \{l_{DT_j,DR_i}, \forall DT_j \in \mathcal{S}_{DT}; \forall DR_i \in \mathcal{S}'_{DR}\}$ is the set of all potential D2D links.

For clarity of presentation, we define the following notations:

1) All $K$ D2D UEs are identified by a number $i$, $\forall i = 1, \ldots, K$ (See Fig. 1).
2) Let $g_{ij}(t)$ be the channel gain at time $t$ of D2D channel $i-j$. Then, a channel gain matrix $\mathbf{G}_{DT,DR}$ can be defined as:

$$\mathbf{G}_{DT,DR} = \begin{bmatrix} g_{11}(t) & \ldots & g_{1K}(t) \\ \vdots & \ddots & \vdots \\ g_{K1}(t) & \ldots & g_{KK}(t) \end{bmatrix} = [\mathbf{g}_1(t) \ldots \mathbf{g}_K(t)], \tag{1}$$

where

$$g_{ij} = \begin{cases} g_{ii}(t) = 0 & \forall i = 1, \ldots, K, \\ g_{ij}(t) = |h_{ij}(t)|^2 d_{ij}^{-\alpha} > 0 & \forall i, \forall j \neq i \end{cases}$$

$\mathbf{g}_k(t)$ is the $k^{th}$ column of matrix $\mathbf{G}_{DT,DR}$ emphasizing the channel states between UE $k$ and the other UEs, $h_{ij}(t)$ is the short-scale channel coefficient of channel $i-j$ at time $t$ and $\alpha$ is the path-loss exponent. Channels $h_{ij}(t)$ are independently and identically distributed (i.i.d.) complex Gaussian random variables with zero mean and unit variance. The channel gain $g_{ij}$ will be modeled as a FSMC as detailed in Section III.
3) Let $l_{ij}(t)$ be the D2D directional link between D2D UEs $i$ and $j$, at time $t$ (i.e. transmission is from $i$ to $j$). Then, we define the overall D2D links matrix as $\mathbf{L}_{DT,DR}$:

$$\begin{aligned} \mathbf{L}_{DT,DR} &= \begin{bmatrix} l_{11}(t) & \ldots & l_{1K}(t) \\ \vdots & \ddots & \vdots \\ l_{K1}(t) & \ldots & l_{KK}(t) \end{bmatrix} = [\mathbf{c}_1 \ldots \mathbf{c}_K] \\ &= \begin{bmatrix} \mathbf{l}_1 \\ \vdots \\ \mathbf{l}_K \end{bmatrix}, \end{aligned} \tag{2}$$

where

$$l_{ij} = \begin{cases} l_{ii}(t) = 0 & \forall i = 1, \ldots, K, \\ l_{ij}(t) = 0 & \text{if } l_{ij} \text{ is not a potential D2D link} \\ l_{ij}(t) = 1 & \text{if } l_{ij} \text{ is a potential D2D link,} \end{cases}$$

$\mathbf{c}_k$ is the $k^{th}$ column of $\mathbf{L}_{DT,DR}$ illustrating the $DTs$ that can serve the $DR$ $k$, $\forall k = 1, \ldots, K$. While, $\mathbf{l}_k$ is the $k^{th}$ row of $\mathbf{L}_{DT,DR}$ that expresses the $DRs$ which can be served by $DT$ $k$, $\forall k = 1, \ldots, K$.

For the link $l_{ij}$ to be successful, the Signal-to-Interference-plus-Noise-Ratio (SINR) measured at $DR$ $j$ (at time $t$), when $DT$ $i$ is the selected transmitter to $j$, shall be equal or greater than a minimum SINR threshold $\gamma_j$, as expressed below:

$$\text{SINR}_j = \frac{\gamma_{ij}}{\sum\limits_{k \neq i} \gamma_{kj} + 1} \geq \gamma_j, j \in \mathcal{S}'_{DR} \quad (3)$$

where $\gamma_{ij} = \frac{P_i}{N_0} g_{ij}$, $\gamma_{kj} = \frac{P_k}{N_0} g_{kj}$, $P_i$ (resp. $P_k$) is the transmit power of transmitter $i$ (resp. transmitter $k \neq i$ to other destinations than $j$), $g_{ij}$ is the gain of the channel $i - j$ (at time $t$)[1], and $N_0$ is the power of the *Additive White Gaussian Noise* (AWGN) at $DR$ $j$.

## III. Problem Formulation

In this section, we formulate at first the D2D channels as FSMCs. Then, we show how to formulate the achieved sum rate maximization problem for the D2D network as a centralized deep-$Q$-learning process, in order to determine the optimal policy for link activation.

### A. Time-Varying Channels

D2D channels are modelled as FSMCs, which is an efficient way to capture the fading nature of wireless channels [10], [13]. The SINR, as defined in (3) provides information about the quality of the wireless channel. Directly modelling $\text{SINR}_j$ as a Markov random variable is not appropriate. Alternatively, due to the relation of SINR to the channel coefficient, the short-scale channel gain $|h_{ij}(t)|^2$, at time $t$, can be modeled as a Markov random variable. Then, $|h_{ij}(t)|^2$ can be partitioned and quantified into $H$ levels, where each level corresponds to a state of the Markov channel [13]. As a consequences, for a given link $i - j$, the channel gain $g_{ij}(t)$ can be partitioned and quantified into $H$ levels. We assume that the channel states vary from one state to another when a time slot $t \in \{0, 1, \ldots, T-1\}$ elapses, where $T >> 1$.

### B. Problem Formulation for the DRL Approach

As seen in Section II, $\mathcal{L}_{DT}$ is the set of potential D2D links that can be activated. It is not likely that all links can be activated at the same time. In fact, it is probable that two links share the same UEs (ex: one $DT$ that has data to more than one $DR$), or a UE is a $DR$ for a link and a $DT$ for another link). Consequently, both links cannot be activated simultaneously. Moreover, due to interference, some potential links' SINRs, as defined in (3), can be below the threshold $\gamma_j$, $\forall j \in \mathcal{S}'_{DR}$.

In order to check if a $DT$ $k$ can serve more than one $DR$, from (2) we calculate $\mathbf{l}_k \mathbf{l}_k^t$, where $(.)^t$ is the transpose operator. If the result is $> 1$, that means $k$ can serve two or more $DRs$. To check if a UE $k$ is a $DT$ for a link and a $DR$ for another one, we calculate $(\mathbf{l}_k + \mathbf{c}_k^t) \times (\mathbf{l}_k^t + \mathbf{c}_k)$. If the result is $\geq 2$, then $k$ is shared by at least 2 links.

Taking into account the previous constraints, we want to make an action at each time $t$ to decide which potential links activated allow the maximization of the achieved sum rate, with respect to constraint (3). We assume here that a central agent[2] is responsible for acquiring the D2D channel states and the $\mathcal{L}_{DT}$ states. Next, it assembles and translates all these information into a *System State*. The latter is fed into the Deep-$Q$-Network, which feeds back the optimal action policy $\text{argmax}_\pi Q^*(x, a)$ for the current time $t$, where $Q(.,.)$ is the state-action function, $x$ is the system state, $a$ is the action performed, and $\pi$ is the policy [13]. After obtaining the action, the central agent will send a bit to UEs in $\mathcal{S}_{DT} \cup \mathcal{S}'_{DR}$ to inform them to be active or not. Then, after the decided action is taken, the system will transfer to a new state and the rewards will be obtained according to the reward function. We refer the reader to [11]-[13] for further details on the functioning of the deep reinforcement learning framework.

The following subsections identify the states, actions and rewards for the deep-$Q$-learning model.

*1) System State:* At time $t$, the system state is determined by the states of the UEs' links and their channels. It is defined by:

$$\mathbf{X}(t) = \{\mathbf{G}_{DT,DR}, \mathbf{L}_{DT,DR}, \}, \quad (4)$$

The UEs' states and the links are directly related to the random communication requests initiated by a part of the D2D users, towards a part of the remaining D2D users, at each instant $t$. The number of possible system states can vary significantly. Hence, it is difficult for conventional approaches to handle this problem. Consequently, deep-$Q$-network is an adequate candidate to successfully learn from high-dimensional inputs [13]-[14].

*2) System Action:* The central agent has to decide at time $t$ which links can be simultaneously activated, in order to maximize the system's achieved sum rate with respect to (3). The global action of the agent can be given by:

$$\begin{aligned} \mathbf{A}(t) &= \begin{bmatrix} a_{11}(t) & \ldots & a_{1K}(t) \\ \vdots & \ddots & \vdots \\ a_{K1}(t) & \ldots & a_{KK}(t) \end{bmatrix} = [\mathbf{c'}_1 \ldots \mathbf{c'}_K] \\ &= \begin{bmatrix} \mathbf{l'}_1 \\ \vdots \\ \mathbf{l'}_K \end{bmatrix}, \end{aligned} \quad (5)$$

where

$$a_{ij} = \begin{cases} a_{ii}(t) = 0 & \forall i = 1, \ldots, K, \\ a_{ij}(t) = 0 & \text{if } i \neq j \text{ and } l_{ij} \text{ is not activated} \\ a_{ij}(t) = K & \text{if } i \neq j \text{ and } l_{ij} \text{ is activated,} \end{cases},$$

and $\mathbf{c}'_k$ (resp. $\mathbf{l}'_k$) is the $k^{th}$ column (resp. $k^{th}$ row) of $\mathbf{A}(t)$.

It is to be noted that the actions set is built from $\mathbf{L}_{DT,DR}$, with respect to the following constraints, $\forall k = 1, \ldots, K$:

$$e_k = \mathbf{l}'_k {\mathbf{l}'_k}^t \leq K^2 \quad (6)$$

$$e'_k = (\mathbf{l}'_k + {\mathbf{c}'_k}^t) \times ({\mathbf{l}'_k}^t + \mathbf{c}'_k) \leq K^2 \quad (7)$$

To do so, we propose a greedy method, as presented in Algorithm 1. The obtained action matrix $\mathbf{A}(t)$ guarantees that conditions (6)-(7) are respected. It is to be noted that if (3) is not satisfied for all $DR$s, then the agent has to modify its action (built from Algorithm 1) by removing gradually links from the selected action matrix $\mathbf{A}(t)$, until satisfaction of (3)

[1]The channel $i - j$ is assumed to be symmetrical and that $g_{ij} = g_{ji}$

[2]An agent is the term used in machine learning to designate a central entity that takes actions on the network.

**Algorithm 1** Greedy Action Construction

Create an action matrix $\mathbf{A}(t) = \text{zeros}(K, K)$.
Set $k = 1$.
Set $\mathbf{L}^{(k)}_{DT,DR} = \mathbf{L}_{DT,DR}$.
**while** $\mathbf{L}^{(k)}_{DT,DR} \neq [\,]$ **do**
    From the potential D2D links set $\mathbf{L}_{DT,DR}$,
    randomly select a first link $l_{ij}$ to activate.
    For the selected link $l_{ij}$, set in $\mathbf{A}(t)$: $\mathbf{c}'_i = [0 \dots 0]^t$,
    $\mathbf{c}'_j = [0 \dots 0 \quad K \quad 0 \dots 0]^t$ (where $a_{ij}(t) = K$
    is the $i^{th}$ element in column $\mathbf{c}'_j$), $\mathbf{l}'_j = [0 \dots 0]$ and
    $\mathbf{l}'_i = [0 \dots 0 \quad K \quad 0 \dots 0]$ (where $a_{ij}(t) = K$ is the
    $j^{th}$ element in row $\mathbf{l}'_i$).
    Set $k = k + 1$.
    Create a new potential D2D links set $\mathbf{L}^{(k)}_{DT,DR}$ by
    removing all links in $\mathbf{L}_{DT,DR}$ having the $i^{th}$ or $j^{th}$
    UEs as participants.
**end while**

for the remaining $DR$s, or until no link is left[3]. That means, to un-select link $l_{ij}$, $a_{ij}(t)$ is set to 0 (previously equal to $K$) in $\mathbf{A}(t)$. The proposal of this action construction method aims at minimizing the action space size for a faster convergence of the DRL to the optimal policy, while maximizing the number of D2D links to be activated.

*3) System Reward Function:* The reward function is the objective function of the system. We have chosen to maximize the sum achieved data rate of the D2D network as the objective, with respect to constraint (3). In our system model, that means maximizing the number of activated links with best SINRs above the thresholds of the $DR$s. Unlike previous work in the literature, we consider also the penalties caused by the non-satisfaction of $DRs$ requests that have found at least one $DT$ with their data, but were not activated. The reward function at the $k^{th}$ $DR$, $\forall k \in \mathcal{S}'_{DR}$ can be expressed by:

$$r_k(t) = \begin{cases} \alpha_k \log_2(1+\gamma_k) & \text{if } \sum_{i=1}^{K} (a_{ik} - l_{ik}) < 0, \\ \log_2(1 + \text{SINR}_k) & \text{if } \sum_{i=1}^{K} (a_{ik} - l_{ik}) \geq 0 \end{cases} \quad (8)$$

where the term $\sum_{i=1}^{K} (a_{ik} - l_{ik})$ is the sum of the $k^{th}$ column elements of matrix $\mathbf{A}(t) - \mathbf{L}_{DT,DR}$ and it captures the differences between the activated and inactivated potential D2D links for the $k^{th}$ $DR$. $\alpha_k log_2(1+\gamma_k)$ is the penalty given to the agent when a $DR$'s request is not satisfied despite the presence of a neighboring $DT$ with the requested data, and $\alpha_k$ is an arbitrary penalty value $-1 \leq \alpha_k \leq 0$.

The immediate system reward $r(t)$ is the sum of all the rewards i.e. $r(t) = \sum_{k \in \mathcal{S}'_{DR}} r_k(t)$. The central agent receives reward $r(t)$ in state $\mathbf{X}(t)$, when action $\mathbf{A}(t)$ is performed in time $t$. Using the deep-$Q$-network aims at determining the selection policy that maximizes the discounted cumulative reward $R$ during the communication period $T$. It is given by:

$$R = \max_\pi \mathbb{E}\left[\sum_{t=0}^{T-1} \varepsilon^t r(t)\right], \quad (9)$$

where $\pi$ is the selected policy (action), $\mathbb{E}(.)$ is the expectation function, and $0 < \varepsilon \leq 1$ is the discount factor of the future rewards [13].

Let $D(t) = \{e(0), \dots, e(t-1)\}$ be the replay memory of the DRL framework, where $e(t') = \{\mathbf{X}(t'), \mathbf{A}(t'), r(t'), \mathbf{X}(t'+1)\}$ is the experience at time $t'$. We define $Q(x, a; \theta)$ as the neural network function approximate of state-action function $Q(x, a)$. Hence, the optimal action-value function $Q^*(x, a) \approx Q(x, a; \theta)$, where parameters $\theta$ are the weights of the neural network. Finally, we define the target value of the DRL framework as $y = r + \varepsilon \max_{a'} Q(x', a'; \theta_i^-)$, where $r$ is sum data rate, $x'$ is the next state, $a'$ is the action taken on the next state $x'$ and $\theta_i^-$ are the weights of the neural network used to compute $y$ at iteration $i$, and are updated with the Q network weights $\theta_i$ every time step. Hence, we define $\hat{Q}(x, a; \theta_i^-)$ as the updated action-value function of $y$ after each time step, i.e. $y = r + \varepsilon \max_{a'} \hat{Q}(x', a'; \theta_i^-)$.

Similarly to [12]-[14], the Training Algorithm of the DRL-based framework is given by Algorithm 2 below.

## IV. Simulation Results

In this section, we present the simulation parameters and results of the proposed DRL-based approach. Implementation is realized using Tensorflow 1.5 [15] on a Linux server.

In our simulation, we consider a similar network to the one presented in Fig.1, where the number of D2D users is $K = 10$, randomly spread and are stationary around a BS. We assume that all D2D devices are equipped with one antenna, and that the channels between nodes are FSMC. Based on the definition of FSMC, the short-scale channel gains $|h_{ij}|^2$ $(\forall i, j = , 1 \dots, K)$ are arbitrarily quantified and partitioned into 10 levels values as $10^{-6}$, 0.1, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4 [13]. For simplicity, we assume also that the channel state transition probability is the same for all channels. The matrix of channel state transition, used for the simulation, is given at the end of the paper [13]. The channel state transition probability is changed for simulations in Fig. 6.

Without loss of generality, we assume that transmit powers of $DT$s are the same and are equal to $P$ (different transmit powers can be assumed, but increases further the complexity of the problem [8].). We assume also that the following parameters are fixed in the simulations (unless otherwise specified): SNR= $\frac{P}{N_0} = 20\ dB$, SINR thresholds are fixed to the same value $\gamma = 1$, path-loss exponent $\alpha = 4$, the rate penalty factor $\alpha_k = -0.3$ and the period of time for each episode is $T = 150$ time slots. Finally, the parameters used for the DRL agent are detailed in Table I at the end of the paper.

In Fig. 2, we illustrate the sum rate performances (expressed in bps/Hz) of the following three techniques, as a function

[3]i.e. all SINRs are below their associated thresholds, and in this case, doing nothing is the best option.

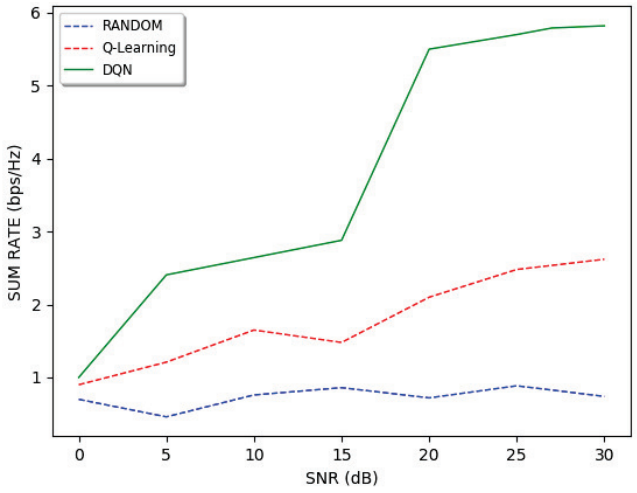


Fig. 2. Average sum rate versus SNR for different schemes

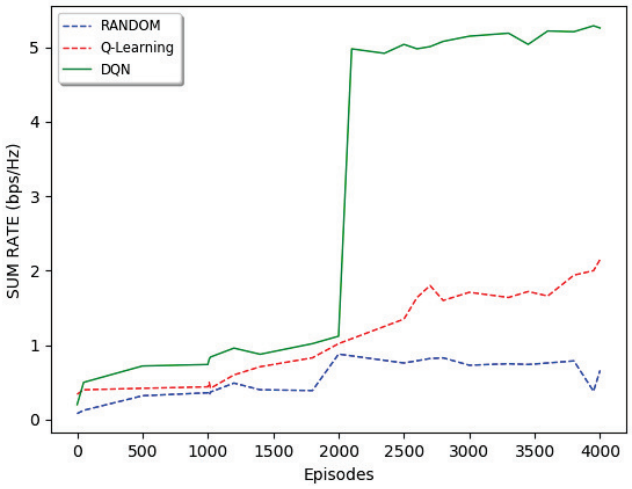


Fig. 3. Average sum rate versus number of episodes for different schemes

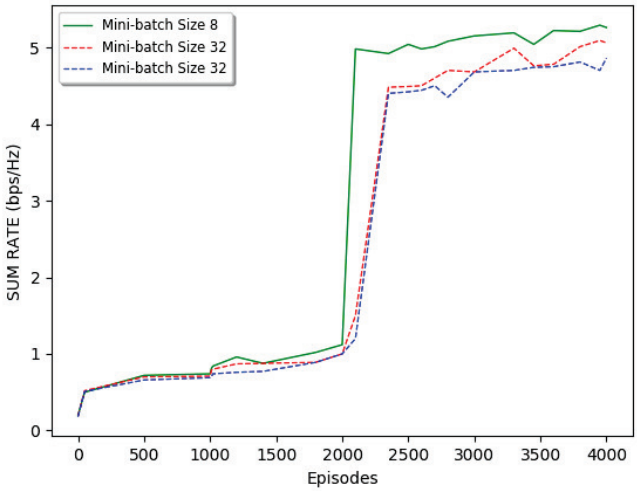


Fig. 4. Average sum rate versus episodes for different mini-batch sizes

**Algorithm 2** DRL Training Algorithm

Initialization.
Initialize Replay Memory $D(t)$ to capacity $N$.
Initialize action-value function $Q$ with random weights $\theta$.
Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$.
**for** each episode **do**
  Initialize the starting state $x$.
  **for** $t = 0, \ldots, T-1$ **do**
    %% $\varepsilon$-greedy policy %%
    Select a random number $p \in [0, 1]$.
    **if** $p < \varepsilon$ **then**
      Select action $a^*(t) = \text{argmax}_a Q(x, a; \theta)$
    **else**
      Select randomly an action $a(t) \neq a^*(t)$
    **end if**
    Execute action $a(t) = \mathbf{A}(t)$ in the system.
    Observe reward $r(t)$ and next state $\mathbf{X}(t+1)$.
    Store experience (or transition) $e(t)$ in $D(t)$.
    %% Experience Replay %%
    Sample random mini-batch of experiences $\{e(j), \forall j \text{ randomly selected}\}$ from $D(t)$.
    **for** $e(j)$ experience in mini-batch **do**
      **if** episode terminates at step $j+1$ **then**
        Set $y_j = r(j)$
      **else**
        Set $y_j = r(j) + \varepsilon\, \hat{Q}(x(j+1), a(j+1); \theta^-)$.
      **end if**
      Perform a gradient descent step on $[y_j - Q(x(j), a(j); \theta)]^2$ with respect to $Q$ network parameters $\theta$.
      Return value of parameters $\theta$ in the deep $Q$ network.
    **end for**
    %% Periodic Update of Target Network %%
    Every step, reset $\hat{Q} = Q$, i.e. $\theta^- = \theta$
  **end for**
**end for**

of SNR: RANDOM, where links are randomly activated at each time $t$; Q-learning, where only reinforcement learning is applied [16]; and DQN, as detailed in Section III. As shown, the proposed DRL-based transmission technique outperforms the other schemes for all SNR values. This is expected since the optimal policy is obtained using the combination of reinforcement learning and deep network. Q-learning presents worse results, but better than RANDOM. The latter realizes very low and unstable performances.

Fig. 3 compares the sum rates of the three previous techniques as a function of the number of learning episodes. As it can be seen, DQN converges the fastest to optimum, compared to the other techniques. Indeed, after $\approx$ 2000 episodes, DQN's neural network weights are optimized and allow best performances achievement. The performances gaps between the techniques agree with the conclusions of the

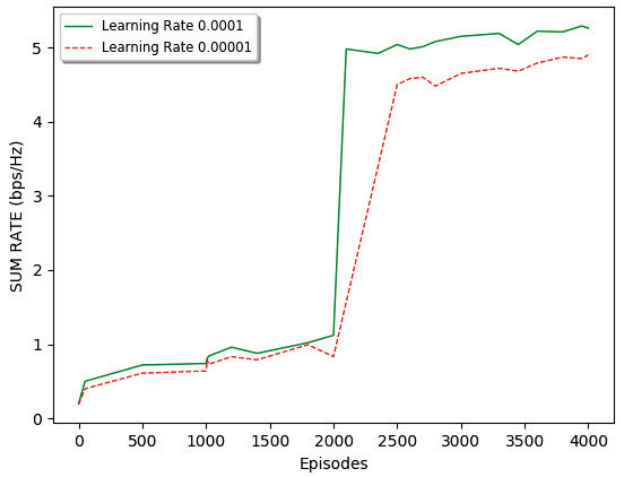


Fig. 5. Average sum rate versus episodes for different learning rates

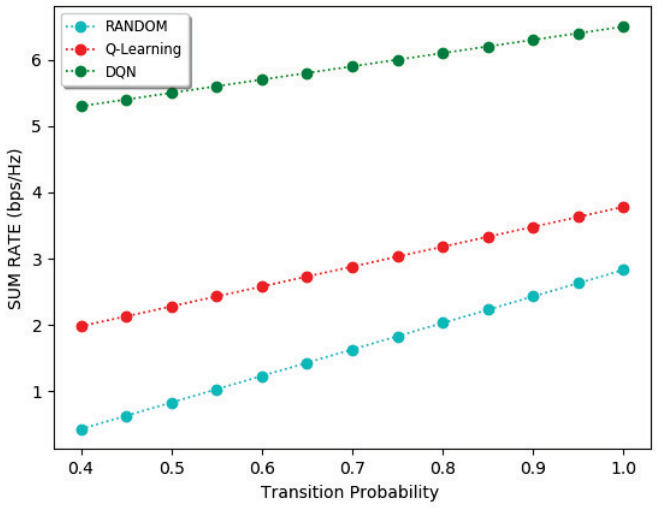


Fig. 6. Average sum rate versus state-transition probability

previous figure.

In Fig. 4, the effect of the mini-batch size is investigated for the DQN technique. The mini-batch size fixes the number of experiences that will be used at each training step. With a small-sized mini-batch (=8), convergence is faster than in the other cases with larger mini-batch sizes. 8 is an adequate size and is the one used along all simulations.

In Fig. 5, we study the impact of the learning rate on the DQN performance. As the number of episodes increases, the sum rate improves until it reaches an approximately stable value. With learning rate 0.0001, convergence is achieved faster than in the case of learning rate 0.00001. Nevertheless, a higher learning rate tends to achieve a local optimum rather than a global one.

Fig. 6 presents the impact of different state-transition probabilities for staying in the same state, on the average sum rate of the network. As defined in matrix $\mathbf{P}_{ch}$, the other transition probabilities are calculated as half-each other successively. As it can be seen, the sum rate increases proportionally with the transition probability for all techniques. This is expected, since with a more static channel, the optimal policy is obtained faster. For all transition probability values, DQN outperforms Q-learning and RANDOM as agreed in Figs. 2-3.

## V. Conclusion

In this paper, we tackle the problem of overlay D2D communications, where realistic time-varying channels and D2D interference are considered. The complexity of the system is very high when channel states are modeled as finite-state Markov channels. Aiming at maximizing the sum rate of the D2D network, we propose a deep-reinforcement-learning approach, based on deep-$Q$-network, to activate D2D transmissions. Simulation results illustrate the advantages of the proposed DRL-based scheme compared to other approaches.

## References


[1] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, $4^{th}$ quarter 2014.

[2] R. I. Ansari, C. Chrysostomou, S. A. Hassan, M. Guizani, S. Mumtaz, J. Rodriguez, and J. J. P. C. Rodrigues, "5g d2d networks: Techniques, challenges, and future prospects," *IEEE Systems J.*, vol. PP, no. 99, pp. 1–15, 2018.

[3] S. Mumtaz, K. M. S. Huq, and J. Rodriguez, "Direct mobile-to-mobile communication: Paradigm for 5g," *IEEE Wireless Commun.*, vol. 21, no. 5, pp. 14–23, Oct. 2014.

[4] A. Abrardo and M. Moretti, "Distributed power allocation for d2d communications underlaying/overlaying ofdma cellular networks," *IEEE Trans. on Wireless Commun.*, vol. 16, no. 3, pp. 1466–1479, March 2017.

[5] S. Dominic and L. Jacob, "Distributed resource allocation for d2d communications underlaying cellular networks in time-varying environment," *IEEE Commun. Letters*, vol. 22, no. 2, pp. 388–391, Feb. 2018.

[6] A. Celik, R. M. Radaydeh, F. S. Al-Qahtani, and M. S. Alouini, "Resource allocation and interference management for d2d-enabled dl/ul decoupled het-nets," *IEEE Access*, vol. 5, pp. 22 735–22 749, 2017.

[7] S. Bulusu, N. B. Mehta, and S. Kalyanasundaram, "Rate adaptation, scheduling, and mode selection in d2d systems with partial channel knowledge," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1053–1065, Feb. 2018.

[8] L. Zhang, M. Xiao, G. Wu, and S. Li, "Efficient scheduling and power allocation for d2d-assisted wireless caching networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2438–2452, Jun. 2016.

[9] J. Lyu, Y. H. Chew, and W. C. Wong, "A stackelberg game model for overlay d2d transmission with heterogeneous rate requirements," *IEEE Trans. Veh. Tech.*, vol. 65, no. 10, pp. 8461–8475, Oct. 2016.

[10] H. S. Wang and N. Moayeri, "Finite-state markov channel - a useful model for radio communication channels," *IEEE Trans. Veh. Tech.*, vol. 44, no. 1, pp. 163–171, Feb. 1995.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. MIT Press, 2017.

[12] W. Cao, G. Feng, S. Qin, and M. Yan, "Cellular offloading in heterogeneous mobile networks with d2d communication assistance," *IEEE Trans. on Veh. Tech.*, vol. 66, no. 5, pp. 4245–4255, May 2017.

[13] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. M. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Tech.*, vol. 66, no. 11, pp. 10 433–10 445, Nov. 2017.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, and al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[15] M. A. et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[16] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 1st ed. MIT Press, 1998.

$$\mathbf{P}_{ch} = \begin{bmatrix} 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 \\ 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 \\ 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 \\ 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 \\ 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 \\ 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 \\ 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 \\ 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 \\ 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 \\ 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 \end{bmatrix}$$

| Parameter | Value | Description |
|---|---|---|
| Mini-batch size | 8 | Number of experiences used for each training step |
| Capacity of the replay memory ($N$) | 2 500 | Sampling mini-batches from most recent experiences in replay memory |
| Pre-training steps | 1 000 | Nbr. random actions before learning (Population of replay memory) |
| Total training steps | 4 000 | Total nbr. of steps to train the $Q$ network |
| Discount Factor ($\varepsilon$) | 0.99 | Discount factor of the $Q$-function |
| Learning rate | 0.0001 | Learning rate of the Optimizer |

Table I: DQN PARAMETERS FOR SIMULATIONS