Energy-Efficient Channel Switching in Cognitive Radio Networks: A Reinforcement Learning Approach

Haichuan Ding, Xuanheng Li, *Member, IEEE*, Ying Ma, and Yuguang Fang, *Fellow, IEEE*

***Abstract*—In this paper, we investigate energy-efficient channel switching for secondary users (SUs) in cognitive radio networks. Unlike existing schemes where SUs adopt the same channel switching strategies regardless of which channel they currently stay at, our scheme allows SUs to adapt their channel switching strategies to the primary users' (PUs') behaviors on the current channels and apply different channel switching strategies on different channels. Considering the unknown PUs' behaviors, we formulate a reinforcement learning problem which allows SUs to learn channel switching schemes by interacting with the environment. Through simulations, we demonstrate the effectiveness of the learned channel switching scheme.**

***Index Terms*—Channel switching, cognitive radio networks, reinforcement learning**

I. Introduction

To guarantee the performance of primary users (PUs) in cognitive radio networks, secondary users (SUs) need to vacate the channels once PUs reclaim them. In this case, SUs can switch to another vacant channel to continue data transmissions, which is known as the channel switching process. With channel switching, SUs can more efficiently use under-utilized spectrum resources to support desired services, such as video streaming [1]–[3]. Unfortunately, channel switching is not always preferred due to high energy consumption during the switching process, particularly for energy-constrained devices [4]–[7].

In [6], Agarwal et al. compare the energy efficiency of a multi-channel dynamic spectrum access (MC-DSA) scheme and a single channel dynamic spectrum access (SC-DSA) scheme. In the MC-DSA scheme, when PUs reclaim the channels, SUs immediately switch to another channel and continue transmissions. In contrast, the SC-DSA scheme allows SUs to exploit the delay tolerance of the data to stop transmissions and wait on the current channels for spectrum access opportunities. According to their analysis, the MC-DSA scheme is not always more energy-efficient than the SC-DSA scheme. In other words, once the current channel is sensed busy, it might be more energy-efficient for SUs to wait on the current channel if their data is delay-tolerant. Based on this observation, Wang et al. design an energy-efficient spectrum sensing and channel switching scheme by jointly considering imperfect spectrum sensing, the throughput and delay requirements of secondary transmissions [4]. Similarly, Wu et al. consider a cognitive radio network where SUs proactively search for target channel to switch to and develop a channel switching scheme to minimize the energy consumption of transmitting a piece of delay-constrained data [5]. Despite their inherent energy-efficient design, these schemes might not be effective since they ignore heterogeneous PUs' behaviors on different channels and adopt the same channel switching strategy regardless which channel SUs currently stay on. If PUs reclaim the current channel only for a very short period of time, SUs can wait on this channel so that the energy consumed due to channel switching can be saved and the data can still be timely transmitted. In contrast, if the current channel remains occupied for a long time, waiting on the same channel might not be energy-efficient since later SUs still need to switch away from the current channel to meet the delay constraint. Thus, a good channel switching scheme should consider the heterogeneous PUs' behaviors on different channels and make channel switching decisions based on the PUs' behaviors on the current channel. Noticing that SUs might not always know PUs' behaviors in advance, in this paper, we formulate a reinforcement learning problem which allows SUs to learn to make channel switching decisions through its past interactions with the environment. Based on the formulated reinforcement learning problem, we obtain a current channel aware (CCA) channel switching scheme via Q-learning. Through extensive simulations, we demonstrate the effectiveness of the learned CCA scheme and show its superiority over existing schemes.

II. Network Model

We consider a time-slotted system where M channels are owned by PUs and a secondary link intends to access these channels for data transmissions when they are not occupied by PUs. Two SUs of the secondary link are d meters away, and the transmitter, denoted as SUt, attempts to deliver N data packets of length L bits to the receiver, denoted as SUr, in T slots. In each time slot, SUs can communicate on one of the M

This work was partially supported by the U.S. National Science Foundation under grants IIS-1722791. The work of X. Li was partially supported by the National Natural Science Foundation of China under Grant 61801080, the Fundamental Research Funds for the Central Universities under Grant DUT20RC(4)007 and the Doctoral Research Initiation Fund of Liaoning Province under Grant 2019-BS-049. (Corresponding author: Xuanheng Li)

Haichuan Ding is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA (email: dhcbit@gmail.com).

Ying Ma and Yuguang Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA (email: mayingbit2011@gmail.com, fang@ece.ufl.edu).

Xuanheng Li is with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116023, China (email: xhli@dlut.edu.cn).

channels if it is sensed idle. Following [5], [8], the availability of each channel, which remains unchanged during each time slot, is modeled as a two-state Markov chain. Specifically, when the mth channel is occupied in the current time slot, it will become idle in the next time slot with probability p_m. When the mth channel is idle in the current time slot, it will be occupied in the next time slot with probability q_m. At the beginning of each slot, SUt conducts wideband spectrum sensing to determine the availability of these M channels, which consumes E_s energy. If the current channel is idle, SUt will transmit on this channel during the corresponding time slot. Otherwise, SUt can choose to wait on the current channel without transmitting or switch to another idle channel to continue transmitting. The energy consumed during each channel switching is E_{sw}. From [9], [10], when transmitting, the total power consumption of SUt is $P/\eta{+}P_c$, where P is the transmit power of SUt, η is the efficiency of the RF chain, P_c is the circuit power consumption during transmissions. The power consumption of SUt is 0 when it waits on current channel without transmitting.

Since p_m's and q_m's might not be known to SUt, it can hardly predict future channel availability and make judicious spectrum switching decisions accordingly. To address this challenge, we will formulate a reinforcement learning problems in the next section, which allows SUt to make reasonable channel switching decisions by learning from its past interactions with the environment.

III. Reinforcement Learning for Channel Switching

When the current channel is sensed busy, SUt makes channel switching decisions based on the availability of other channels, the amount of data to be transmitted, and which channel it currently stays at. We collect these information in a vector s_t as

$$s_t = \left(\varepsilon_t, \mathrm{N}_t, \iota_t, t\right), \tag{1}$$

where t is the index of the current time slot, $\varepsilon_t \in \{0,1\}^{1\times M}$ represents the availability of the M channels at the tth slot, and the mth channel is idle in the tth slot when $\varepsilon_t[m] = 1$. N_t is the number of packets to be transmitted at the beginning of the tth slot, and ι_t is the index of the channel where SUt stays at the beginning of the tth slot. s_t summarizes all the information which SUt needs to make channel switching decisions and will be called state in the following analysis.

Based on s_t, SUt can choose its action a_t in the corresponding slot. The actions available at the tth slot are collected in a set $\mathcal{A}_t \subset \{-1, 0, 1, \cdots, M\}$. When $a_t = -1$, SUt waits on the current channel without transmitting. When $a_t = 0$, SUt transmits on the current channel. When $a_t = m$, $m \notin \{-1, 0, \iota_t\}$, SUt switches to the mth channel and continues transmission. $\mathcal{A}_t$ is closely related to s_t. For example, $0 \in \mathcal{A}_t$ only when $\varepsilon_t[\iota_t] = 1$, and $1 \in \mathcal{A}_t$ only when $\varepsilon_t[1] = 1$. According to [11], the mapping from s_t to a_t is called policy and is denoted as π, i.e., $\pi(s_t) = a_t$.

After applying a_t, SUt finds itself in a new state s_{t+1} at the beginning of the $(t+1)$th slot and receives a reward r_{t+1}. To facilitate the learning of an energy-efficient channel switching scheme, we use the energy consumption of SUt in each slot as the reward signal, namely,

$$r_{t+1} = \begin{cases} -e_t & s_{t+1} \notin \mathcal{S}_{\mathcal{T}} \\ -e_t & s_{t+1} \in \mathcal{S}_{\mathcal{T}}, \mathrm{N}_t = 0 \\ -e_t - \phi & s_{t+1} \in \mathcal{S}_{\mathcal{T}}, t = T, \mathrm{N}_t > 0 \end{cases}, \tag{2}$$

where e_t is the energy consumption of SUt in the tth slot, and $\mathcal{S}_{\mathcal{T}}$ is the set of terminal state. The terminal states are states where the data transmission process terminates, which happens either when the delay constraint is achieved or when all the packets are delivered. In other words, $\mathcal{S}_{\mathcal{T}}$ contains states with either $t = T$ or $\mathrm{N}_t = 0$. If the process terminates at a state with $t = T, \mathrm{N}_t > 0$, a penalty of ϕ as shown in (2) will be incurred since the delay constraint is violated. In contrast, if the process terminates at a state with $\mathrm{N}_t = 0$, the data packets are successfully delivered and no penalty will be incurred. This is how the delay constraint T is considered in our formulation given the state s_t defined in (1).

From Section II, when $a_t = 0$, the energy consumption is due to spectrum sensing. When $a_t > 0$, channel switching will incur additional energy consumption besides spectrum sensing. Thus, e_t can be expressed as

$$e_t = \begin{cases} E_s & a_t = -1 \\ E_s + E_{tx} & a_t = 0 \\ E_s + E_{sw} + E_{tx} & a_t > 0 \end{cases}, \tag{3}$$

where E_s is the amount of energy spent on spectrum sensing, E_{tx} is the energy consumption due to data transmissions. Clearly, E_{tx} is closely related to the number of packet delivered in the tth slot. According to [12], the capacity of the secondary link is[1]

$$C = B \log_2\left(1 + \gamma P d^{-k} / \sigma^2\right), \tag{4}$$

where B is the bandwidth of the current channel, γ is an antenna related constant, k is the path loss exponent, and σ^2 is the power of the noise. Then, the number of packets delivered in the tth slot can be derived as

$$n_t = \min\left\{ \left\lfloor \frac{C\left(\tau - \tau_s\right)}{L} \right\rfloor, \mathrm{N}_t \right\}, \tag{5}$$

where τ is the length of slot and τ_s is the time spent on spectrum sensing. $\lfloor . \rfloor$ is the floor function and the min operator is used to ensure that the number of packets transmitted during the tth slot is no more than that to be transmitted at the beginning of the corresponding slot. Based on the discussions in Section II, E_{tx} can be derived as

$$E_{tx} = \frac{\left(P/\eta + P_c\right) L n_t}{C}. \tag{6}$$

Then, the reward received by SUt during each time slot can be derived based on (2)–(6). To facilitate energy-efficient channel switching, we are interested in a policy, π^*, which

[1] To highlight the impact of PUs' behaviors on SUs' channel switching strategies, we do not consider channel fading in our model. However, wireless channel fading can be easily considered in our model by either introducing another element to the state s_t or incorporating it into ε_t by viewing the channel fading as a part of PUs' behaviors.

maximizes $\mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$, the expected accumulated reward received during the data delivery process. Note that the policy π^* gives us the optimal channel switching scheme. Due to the lack of knowledge about p_m's and q_m's, the state transition probabilities are not known to SUt. In this case, SUt cannot obtain the optimal channel switching scheme, i.e., the policy π^*, directly by formulating and solving a Markov decision process. As a result, Q-learning, a model-free reinforcement learning scheme, is adopted, which allows SUt to find the optimal channel switching scheme through its previous actions, observed states, and received rewards. Specifically, SUt intends to learn a function $Q(s_t, a_t)$ which is an approximation of the optimal action-value function $q_*(s_t, a_t)$. $q_*(s_t, a_t)$ is the maximum expected accumulated reward starting from state s_t and taking action a_t [11]. With $q_*(s_t, a_t)$, the optimal channel switching scheme, i.e., the policy π^*, can be obtained as $\pi^*(s_t) = \arg\max_{a_t \in A_t} q_*(s_t, a_t)$. The learning process breaks naturally into episodes. SUt attempts to transmit N packets during each episode, and an episode ends when those packets are delivered or the delay constraint is achieved.

Each episode starts with a state where $\mathrm{N}_t = N$ and $t = 1$. For each slot within this episode, SUt selects an action based on s_t and the ϵ-greedy policy and obtains a reward r_{t+1} [11]. With r_{t+1}, SUt updates $Q(s_t, a_t)$ as

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Bigg(r_{t+1} + \max_{a' \in \mathcal{A}_{t+1}} Q(s_{t+1}, a') - Q(s_t, a_t)\Bigg), \tag{7}$$

where α is a step-size parameter, and $\mathcal{A}_{t+1}$ is the set of available actions in the $(t+1)$th slot. With the learned function $Q(s_t, a_t)$, SUt can obtain an energy-efficient channel switching scheme as $\pi(s_t) = \arg\max_{a_t \in \mathcal{A}_t} Q(s_t, a_t)$. This implies that, at state s_t, SUt always selects the action which is expected to minimize energy consumption. In other words, when SUt decides to switch to another channel, the available channels will not be eqully likely selected if PUs' behaviors on these channels are different.

From above discussions, the computational complexity of the learning algorithm mainly comes from the derivation of the updated function $Q(s_t, a_t)$ based on (7). As a result, we can analyze its computational complexity based on how often (7) is computed and the complexity to compute (7). If the training process takes $\mathcal{I}$ episodes, (7) will be computed $\mathrm{O}(\mathcal{I}T)$ times, where T is the delay constraint and thus the upper bound of how many times (7) is computed per episode. The computation in (7) includes 3 additions/substractions, 1 multiplication, and $|\mathcal{A}_{t+1}| - 1$ comparisons, where $|\mathcal{A}_{t+1}|$ is the cardinality of $\mathcal{A}_{t+1}$. Noticing that each of these operations takes constant time and $|\mathcal{A}_{t+1}|$ is bounded by $M + 2$, the complexity of (7) is $\mathrm{O}(M)$, where M is the number of channels to be considered. Based on above discussions, if the training process takes $\mathcal{I}$ episodes, the complexity of the learning algorithm is $\mathrm{O}(\mathcal{I}TM)$.

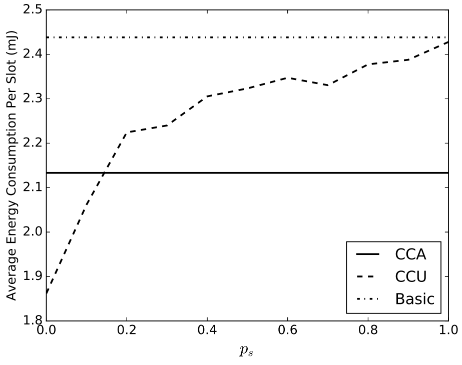

Fig. 1. Energy consumption under different schemes.

IV. Performance evaluation

To evaluate the performance of the learned channel switching scheme, we consider a scenario with $M = 3$ channels which can be opportunistically accessed by SUs for data transmissions. Each channel has a bandwidth of $1MHz$. The parameters related to PUs' behavior are set as $p_1 = 0.9, q_1 = 0.8, p_2 = 0.1, q_2 = 0.5, p_3 = 0.9, q_3 = 0.8$. The distance between the transceivers of the considered secondary link is $d = 100m$. In each episode, SUt attempts to deliver $N = 10$ packets of length $1500Bytes$ in $T = 15$ slots. The length of each slot is $\tau = 10ms$ and the first $\tau_s = 2ms$ of each slot is dedicated to spectrum sensing. SUt adopts a fixed power $P = 18mW$ for data transmissions, and the circuit power consumption during transmissions is $P_c = 30mW$. The parameter η is set to 1. In each slot, SUt will spend $E_s = 0.06mJ$ on spectrum sensing. SUt will spend $E_{sw} = 0.2mJ$ per channel switching [4], [6]. Noticing that the noise power spectral density at the receiver is $-110dBm/Hz$, the power of the noise σ^2 is 1. The signal propagation related parameters are set to $\gamma = 1$ and $k = 3$. During the learning process, we set the step-size parameter α to 0.1 and adopt a ϵ-greedy policy with $\epsilon = 0.1$ for exploration. The following performance evaluation is conducted based on the channel switching policy learned after 10^4 episodes.

In Fig. 1, we present the energy consumption of our CCA channel switching scheme. To evaluate the effectiveness of our scheme, we compare it with the current channel unaware (CCU) channel switching scheme introduced in [4]. Unlike our CCA channel switching scheme, the CCU channel switching scheme does not consider the heterogeneous PUs' behaviors on different channels and always adopts the same channel switching strategies regardless which channel SUt currently stays at. Specifically, in the CCU channel switching scheme, SUt switches to another channel with probability p_s as long as the current channel is sensed busy. The energy consumption of the CCU channel switching scheme under various p_s is shown in Fig. 1. Besides the CCU channel switching scheme, we also present the energy consumption for a basic channel switching scheme where SUt makes channel switching decisions without considering the energy consumption incurred by channel switching. It should be noted that all the results presented in

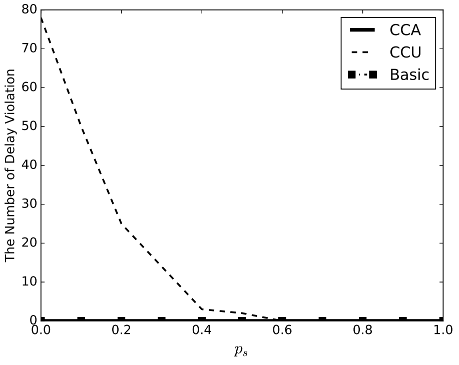

Fig. 2. The number of delay violation under different schemes.

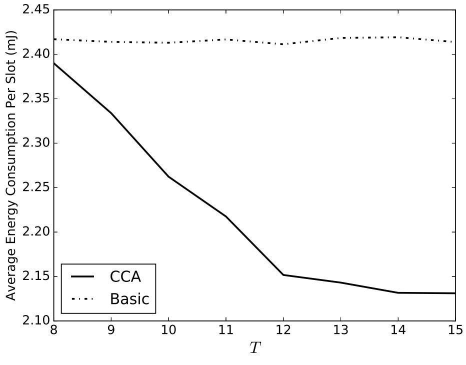

Fig. 3. Average energy consumption per slot v.s. delay constraints.

Fig. 1 are averaged over 200 episodes. From Fig. 1, the basic channel switching scheme has the highest energy consumption. Notice that the basic scheme does not consider the energy consumption incurred by channel switching when making channel switching decisions. Under such a scheme, SUt always attempts to switch to another idle channel when the current channel is sensed busy, which leads to many unnecessary channel switching actions and high energy consumption. Thus, it is important to consider the energy consumption due to channel switching when making channel switching decisions. As shown in Fig. 1, the CCU channel switching scheme has the least energy consumption when p_s is small, and the CCA channel switching scheme gradually becomes more advantageous when p_s increases. For the CCU channel switching scheme with small p_s, channel switching is seldom used and SUt mostly exploits a single channel for data transmissions, which limits SUt's spectrum access opportunities. In other words, the low energy consumption of the CCU channel switching scheme with small p_s is the result of limited data transmission opportunities and minimized channel switching actions. Clearly, with less transmission opportunities, it becomes more difficult for the packets to be delivered before the deadline. In other words, when p_s is small, the small energy consumption of the CCU scheme is achieved at the cost of violating delay constraints. This can be corroborated by the results in Fig. 2 where we present the corresponding number of episodes with delay violation under different schemes during the experiment. According to Fig. 2, the CCU scheme with small p_s could result in a very large number of delay violations. Although the number of delay violations decreases when p_s increases, the energy consumption of the CCU scheme increases. In contrast, as shown in Fig. 1 and Fig. 2, our CCA scheme can achieve a small energy consumption with 0 number of delay violations. This implies that, when compare with the CCU scheme, our CCA scheme can more effectively balance data transmissions and energy consumption. This result not only demonstrates the effectiveness of our CCA scheme but also highlights the importance of adapting SUs' channel switching decisions to the PUs' behaviors on the current channels.

In Fig. 3, we study how the energy consumption of our CCA channel switching scheme varies with the delay constraint T. In the experiment, we adopt the same parameter settings used in Fig. 1 except T, and the results are obtained by averaging the energy consumption in 200 episodes. The CCU channel switching scheme is not compared in Fig. 3 since it does not consider the delay constraint. From Fig. 3, the energy consumption of our CCA scheme increases with decreasing T, whereas, the energy consumption of the basic scheme is not affected by the variations in T. Given the considerable amount of energy consumption during channel switching, the basic idea of our CCA scheme for energy saving is to exploit the delay tolerance of the data to avoid unnecessary channel switching. Specifically, once PUs reclaim the channels, instead of immediately switching to another channel, our CCA scheme lets SUt wait on its current channel for spectrum access opportunities as long as the delay constraints are not violated. With decreasing T, the data packets need to be delivered in a shorter time period, which renders SUt less flexibility in handling these data packets. To ensure timely data delivery, SUt might have to switch to another channel for transmissions rather than waiting on the current channel for future spectrum access opportunities. Thus, the energy consumption of our CCA scheme increases with decreasing T. When T is small enough, our CCA should let SUt take almost all possible opportunities for data transmissions. This explains why the energy consumption of our CCA scheme gets close to that of the basic channel switching scheme.

Finally, the convergence of the learning algorithm is studied in Fig. 4. We use the same parameter settings as defined at the beginning of this section. The only difference is that we also consider the case with the step-size parameter $\alpha = 0.5$. From Fig. 4, the learning algorithm can converge in 2000 episodes with a step-size parameter $\alpha = 0.5$ and 6000 episodes with $\alpha = 0.1$. In practice, we can adopt advanced learning algorithms (e.g., using function approximators) to expedite the learning process and exploit the periodical pattern in PUs' activities to enable the reuse of the previous training results [11], [13], [14].

V. Conclusion

In this paper, we develop an energy-efficient channel switching scheme for cognitive radio networks based on Q-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3006471, IEEE Transactions on Vehicular Technology

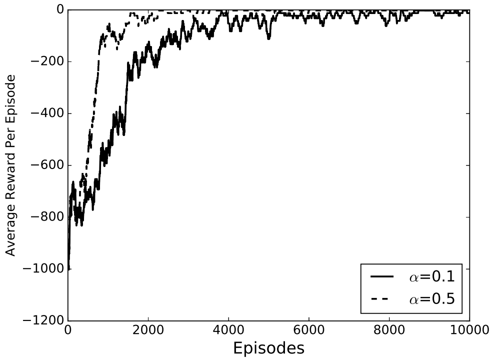

Fig. 4. The convergence of the learning algorithm.

learning. Unlike existing work, in our scheme, SUs' channel switching strategies is closely related to the PUs' behaviors on the current channel. Through comparison with existing channel switching schemes, we demonstrate that our scheme can more effectively balance data transmissions and energy consumption. Our results imply that, with properly designed channel switching schemes, we can effectively exploit the delay tolerance of the data to avoid unnecessary channel switching and achieve considerable energy saving.

References

[1] Y. Zhao, Z. Hong, Y. Luo, G. Wang, and L. Pu, "Prediction-based spectrum management in cognitive radio networks," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3303–3314, Dec. 2018.

[2] Y. Wu, F. Hu, S. Kumar, Y. Zhu, A. Talari, N. Rahnavard, and J. D. Matyjas, "A learning-based qoe-driven spectrum handoff scheme for multimedia transmissions over cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 2134–2148, Nov. 2014.

[3] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, "On a novel deep-learning-based intelligent partially overlapping channel assignment in sdn-iot," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 80–86, Sept. 2018.

[4] S. Wang, Y. Wang, J. P. Coon, and A. Doufexi, "Energy-efficient spectrum sensing and access for cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 2, p. 906, Feb. 2012.

[5] Y. Wu, Q. Yang, X. Liu, and K. S. Kwak, "Delay-constrained optimal transmission with proactive spectrum handoff in cognitive radio networks," *IEEE Trans. Commun.*, vol. 64, no. 7, pp. 2767–2779, Jul. 2016.

[6] S. Agarwal and S. De, "Impact of channel switching in energy constrained cognitive radio networks," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 977–980, Jun. 2015.

[7] J. Ren, Y. Zhang, N. Zhang, D. Zhang, and X. Shen, "Dynamic channel access to improve energy efficiency in cognitive radio sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3143–3156, May 2016.

[8] K. Wu, H. Jiang, and C. Tellambura, "Sensing, probing, and transmitting policy for energy harvesting cognitive radio with two-stage after-state reinforcement learning," *IEEE Trans. Veh. Technol.*, Accepted for publication 2018.

[9] M. Gregori and M. Payaró, "On the optimal resource allocation for a wireless energy harvesting node considering the circuitry power consumption," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 5968–5984, Nov. 2014.

[10] J. Xu and R. Zhang, "Throughput optimal policies for energy harvesting wireless transmitters with non-ideal circuit power," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 2, pp. 322–332, Feb. 2014.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction (2nd Edition)*. MIT Press, 1998.

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory (2nd Edition)*. John Wiley & Sons, 2006.

[13] F. Xu, Y. Li, H. Wang, P. Zhang, and D. Jin, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1147–1161, Apr. 2017.

[14] B. Mao, F. Tang, Z. M. Fadlullah, and N. Kato, "An intelligent route computation approach based on real-time deep learning strategy for software defined communication systems," *IEEE Trans. Emerg. Topics Comput.*, Early Access 2019.