Cooperative communication has emerged as a new wireless network communication concept, in which parameter optimization such as cross-layer cooperation plays an important role. Heuristic evaluation postdecision state learning algorithm (HE-PDS) is proposed in cross-layer cooperation. The proposed algorithm exploits the determinate state information and jointly considers the transmitting power and channel state condition at the physical layer and the buffer congestion control at the media access control layer. The experimental results show that the cumulative average total costs of HE-PDS algorithm decrease about ten times and 8% under the maximum delay and throughput constraints and the power costs decrease about 50% and 28% under various delay limits and about 100% and 56% under the different throughput constraints than the traditional Q learning algorithm and PDS algorithm, demonstrating that the proposed algorithm has much better energy-efficient performance and faster convergence speed and outperforms the traditional Q learning algorithm and PDS learning algorithm.

1. Introduction

Recently, the merits of cooperative communication in the physical layer have been explored. However, the impact of cooperative communication on the design of the higher layers has not been well understood yet. As wireless devices often rely on battery power sources in wireless communication, how to minimize the energy consumption under the constraints on both delay and throughput has posed a great challenge and attracted lots of research attention in recent years [1]. Besides, affected by the fading channel state, time-varying buffer state, and dynamic traffic characteristics, this problem becomes more sophisticated [2]. Since the unknown environment can be modeled as a Markov decision process (MDP), it is reasonable to build the cross-layer transmission strategy based on this property [3, 4]. The state of the art of research on the energy-efficient problem in wireless communication can be mainly divided into two categories: the cross-layer design and approximate algorithm design. Related research is as follows.

From the view of energy-efficient design, the authors in [5] analysed the throughput performance. However, the feature of limited buffer has not been taken into account during the performance analysis. References [6–8] considered energy-efficient packet transmission under packet delay constraint. In [9, 10], the authors investigated the balance between throughput and the energy consumption. Although all these works obtained good energy-efficient performance, the trade-off among delay, throughput, and energy consumption is not fully considered. Aiming at the characteristics of MDP model, [6, 9–11] formulated the optimal packet transmission as a control policy which was solved by reinforcement learning (RL) algorithm. However, most of these works performed the computation offline which resulted in restricted application. In [7], the authors introduced the postdecision state (PDS) learning to raise the convergence rate. Unfortunately, the state of the transmission power of the model has not been taken into account. Although [11] took the power state into consideration, the effect of trade-off between exploration and exploitation is not been fully considered. Therefore, the convergence performance of the algorithm is needed to be further improved.

To address the aforementioned challenge, this paper extends our prior work [8] by considering the constraints...
of both delay and throughput simultaneously. We propose a heuristic evaluation postdecision state (HE-PDS) algorithm for packet transmission, which has not only low computation complexity but also faster convergence speed. The specific contributions of this paper include the following.

(i) A literature survey about various existing energy-efficient policy, analyzing their advantages and disadvantages.

(ii) An effective energy-efficient optimization models for decreasing the energy consumption is proposed in wireless communication.

(iii) A unified framework to realize a scheduling mechanism is proposed by jointly considering the transmit power and channel state at the physical layer and the buffer congestion control at the media access control layer.

(iv) An online RL algorithm is proposed that fully exploits the known state information about the system’s dynamics to improve learning performance.

(v) Performance analysis of the proposed algorithm and an evaluation of the algorithm with respect to other existing algorithms.

The rest of this paper is organized as follows. A brief overview of the related works is presented in Section 2. The formulation of the problem within the structure of CMDP is presented in Section 3. Section 4 presents an online HE-PDS algorithm for cross-layer optimization. Experiments are given in Section 5. Finally, Section 6 summarizes the anticipated results and discusses some future research directions.

2. Cross-Layer Cooperation Model

Cooperative communication will improve network performance in certain circumstances. However, if the cooperative communication is not necessary, it will make the system more complex, increase the transmission delay, and reduce the efficiency of the system. As illustrated in Figure 1, we consider a point to point system where one single user (a transmitter and receiver pair) transmits data from the finite buffer queue over a time-varying channel. Meanwhile, we divide the transmission time into equal slots of length \( \Delta t \) and time slot \( n \) denotes the discrete time interval \([n \Delta t, (n + 1) \Delta t]\). We assume that the transmission and power management decisions are determined and the system state information remains unchanged in each time slot. According to the feedback of the delay, throughput and channel state information obtained at the receiver, the transmission rate, and transmission power are adaptively adjusted at the transmitter.

2.1. Physical Layer Model. We consider a discrete time block Rayleigh fading channel model with additive white Gaussian noise (AWGN) [12, 13], where its power spectrum density is \( N_0/2 \) and the wireless channel bandwidth is \( W \). During each time slot, we assume that the power gain of the channel state is constant and the transfer of channel state only occurs in the adjacent states. In this paper, we use a finite state Markov channel (FSMC) to describe the wireless channel [14, 15]. As shown in Figure 2, there are \( k \) channel states each of which can be transitioned to its adjacent states.

In Rayleigh fading channel, the received instantaneous signal-to-noise ratio (SNR) \( \phi \) is exponentially distributed with probability density function:

\[
f_{\phi}(\varphi) = \varphi_0 \exp \left(-\frac{\varphi}{\varphi_0}\right),
\]

where \( \varphi_0 = E[\varphi] \) represents the average channel gain. The channel is said to be in state \( h_k \) if the received SNR is in the interval \([\varphi_k, \varphi_{k+1})\]. Let \( N(\varphi) \) be the level crossing rate (LCR), which is given by

\[
N(\varphi) = \sqrt{\frac{2 \pi \varphi}{\varphi_0}} \cdot f_d \cdot \exp \left(-\frac{\varphi}{\varphi_0}\right),
\]

where \( f_d \) is the maximum Doppler frequency. Therefore, the state transition probability can be obtained by the following formula:

\[
p_h(k, k + 1) = N(\varphi_{k+1}) \cdot \frac{\Delta t}{\tau_k}, \quad 1 \leq k \leq K - 1,
\]

\[
p_h(k, k - 1) = N(\varphi_k) \cdot \frac{\Delta t}{\tau_k}, \quad 2 \leq k \leq K,
\]
where the steady state probability (SSP) $\pi_k$ is given by

$$\pi_k = \int p_{k+1} f_k (\phi) d\phi = e^{-\phi_k} - e^{-\phi_{k+1}} \cdot \frac{\phi_k}{\phi_0}. \quad (4)$$

2.2. MAC Layer Model. As shown in Figure 3, let the transmission buffer be the first in first out queue. In the $n$th time slot, the transmitter receives $b_n$ packets, stores them in the finite buffer, and sends some packets from the buffer. The traffic arrival distribution is assumed to follow an independent and identical distribution (IID) during each slot. For simplicity, we assume that the packets arrival follows a Poisson process with rate $\lambda$. Therefore, the probability density with $l$ packets arrival is denoted as

$$p_{\lambda} (l) = \frac{\lambda^l \cdot e^{-\lambda}}{l!}. \quad (5)$$

Afterwards, we define that the backlog at the transmitter buffer is denoted by $b \in \{0, 1, \ldots, \bar{B}\}$, where $\bar{B}$ is the capacity of the finite buffer and each packet contains $L$ bits. Besides, the arrival packets will be dropped if the buffer is full. Meanwhile, we assume a packet arrival occurs at the end of time slot.

Let $z_n$ packets be sent at the transmitter in slot $n$, where $z_n \in \{0, 1, \ldots, \bar{B}\}$. Affected by the bit error ratio (BER), the packets received at the receiver may be smaller than $z_n$; that is, $f_z$ (BER, $z_n$, $x_n$) ≤ $z_n$. Assuming independent packet losses, $f_z$ is represented by a binomial distribution:

$$p^f (f_z | \text{BER}_n, z_n) = b_{\text{int}} (z_n, 1 - \text{PER}_n), \quad (6)$$

where PER is the packet error ratio, which meets $\text{PER}_n = 1 - (1 - \text{BER}_n)^h$. We also define $b_{\text{int}}$ as the initial buffer state and the buffer state at the $n$th slot as $b_n$. Therefore, the buffer state at the transmitter can evolve recursively as follows:

$$b_0 = b_{\text{int}},$$

$$b_{n+1} = \min (b_n - z_n + l_n, B). \quad (7)$$

2.3. Dynamic Power Management Model. To reduce power consumption, we assume that the wireless card can turn to low power state similar to [8, 11, 16]. Specifically, the card may be one of the two power management states; that is, $X \in \{\text{on}, \text{idle}\}$. Furthermore, the power state can be switched to on or idle by the corresponding actions in the set $Y = \{s_{\text{on}}, s_{\text{idle}}\}$. We define $P_{\text{on}}$ and $P_{\text{idle}}$ as the power overhead by the wireless card in the on and idle states, respectively. Let $P_n$ be the power consumption when the state transitions from on to idle or vice versa. In the $n$th slot, if the packet throughput is $z$, then the required power is

$$P_n \left( (h_n, x_n), \text{BER}_n, y_n, z_n \right)$$

$$\rho \left( (h_n, x_n), \text{BER}_n, y_n, z_n \right)$$

$$= \begin{cases} P_{\text{idle}}, & \text{if } x_n = \text{idle}, \quad y_n = s_{\text{idle}}, \\ P_{\text{on}} + P_{tx} (h_n, \text{BER}_n, x_n), & \text{if } x_n = \text{on}, \quad y_n = s_{\text{on}}, \\ P_{tr}, & \text{otherwise}, \end{cases} \quad (8)$$

where $h_n$ is the channel state; $x_n$ is the power management state; and $y_n$ and $P_n$ are the power management action and the transmission power, respectively. Define $u$ as the number of symbols per slot. Following the discussion in [10], the power required for the transmission is given by

$$P_{tx} \geq W \cdot \frac{N_0}{h_n} \cdot \frac{-\log (5 \cdot \text{BER}_n) \cdot \left(2 z_n \cdot \frac{u}{h_n} - 1\right)}{1.5}. \quad (9)$$

In the implementation of power management action, we assume that the delay of the power state switching from one state to another is negligibly small. Let $P_s (y) = [p(x' | x, y)]_{x \neq x'}$ represent the transition probability matrix, which means that the power state is switched from $x$ to $x'$ under the condition that the power management action is $y$. As shown in Figure 4, the sequence of the power management states can be modeled as a constrained Markov chain with transition probabilities.

3. Problem Formulation

As discussed in the second section, in a given channel state, since the energy consumption function is a convex function when $z$ packets are transmitted, there must exist an optimal solution to this problem [17]. Given the buffer state $b$, channel state $h$, and power management state $x$, we define a joint vector state $s_n (b_n, h_n, x_n) \in S$, where $n$ denotes the $n$th
time slot. Furthermore, we formulate this joint vector state process as a CMDP. Meanwhile, we use \( a_n(BEP_n, y_n, z_n) \in A \) to represent the joint action, where \( BEP \) is the bit-error rate, \( y \) is the power management action, and \( z \) is the number of packets to be transmitted. For simplicity, we use buffer overload instead of queue delay of the transmitter \([8]\). Consequently, the holding cost and overflow cost of the buffer can be obtained by the following:

\[
g_{\text{holding}}([b, x], \text{BER}, y, z) = \sum_{f=0}^{\infty} p_f(f | \text{BER}, z) [b - f],
\]

\[
g_{\text{overflow}}([b, x], \text{BER}, y, z) = \sum_{f=0}^{\infty} \sum_{l=0}^{z} p^l(f | \text{BER}, z) \max([b - f] + l - B, 0),
\]

where the holding cost at the start of each slot stands for the number of packets that still remain in the buffer. We use parameter \( \eta \) to fully analyse the effect of the holding cost and overflow cost on wireless transmission. Thus, the buffer cost can be evaluated as

\[
g([b, x], \text{BER}, y, z) = g_{\text{holding}} + \eta \cdot g_{\text{overflow}}.
\]

For the average packet arrival rate \( \lambda \), the system throughput can be calculated by

\[
\text{Throughput} = \lambda \left(1 - P_{\text{drop}}\right) (1 - \text{PER}),
\]

where \( \text{PER} \) is packet error rate and

\[
P_{\text{drop}} = (1/\lambda) \lim_{N \to \infty} \sum_{n=1}^{N} g_{\text{overflow}}
\]

is the long-term average buffer overflow probability. Therefore, the throughput maximization is equivalent to minimizing the total packet loss number, which contains the number of both buffer overflow and lost packets caused by BER. As in [10], its mathematical model can be defined as

\[
\delta = \lambda \cdot P_{\text{drop}} + z \cdot \text{PER}.
\]

In summary, we can reformulate the cross-layer energy-efficient transmission optimization as a problem of minimizing the long-term average power consumption under transmission delay and throughput constraints. Therefore, the optimization problem can be expressed as

\[
\arg \inf_{\pi} \lim_{N \to \infty} \frac{1}{N} E \left\{ \sum_{n=1}^{N} \gamma \rho (s, \pi (s_n) | s_0 = s) \right\},
\]

subject to:

\[
\lim_{N \to \infty} \frac{1}{N} E \left\{ \sum_{n=1}^{N} \gamma \delta (s, \pi (s_n) | s_0 = s) \right\} \leq T,
\]

\[
\lim_{N \to \infty} \frac{1}{N} E \left\{ \sum_{n=1}^{N} \gamma g (s, \pi (s_n) | s_0 = s) \right\} \leq D,
\]

where \( 0 \leq \gamma \leq 1 \) is the discount factor; \( \pi : S \to A \) is a stationary policy which maps system state into transmission rate for each time slot. \( T \) and \( D \) denote the throughput and delay constraints, respectively. Similar to [18], by introducing lagrange multipliers, \( \mu_1, \mu_2 \), this problem can be reformulated as an unconstrained MDP. Specifically, we define the system Lagrangian cost function as

\[
c(s, a) = \rho(s, a) + \mu_1 \cdot \delta(s, a) + \mu_2 \cdot g(s, a).
\]

### 4. Heuristic Evaluation PDS Learning Algorithm

#### 4.1. Algorithm Description

The state information of the environment is often assumed uncertain when the state-action pairs are learned in the traditional Q learning algorithm. Therefore, the known state information can not be fully utilized in the learning process which will inevitably result in poor convergence performance. However, the known information may be determined in most communication systems. Table 1 gives an example of what is known and what is unknown. In Figure 4, when \( P_{\text{on,idle}} \) and \( P_{\text{idle,off}} \) are known and determined, the \( P_f(y) \) in Table 1 can be defined as known and determined. Besides, if the transmission power \( P_{tx} \) in (8) is known, then the power consumption can be classified as known. Similarly, the broadcast arrival probability and holding cost also can be defined as known and stochastic when BER is known in (4).

In order to use the known state information, we introduce postdecision state (PDS) and PDS value function as in [7]. In PDS learning algorithm, the search of optimal strategy is mainly performed by PDS. Specifically, as shown in Figure 5, the PDS is a virtual state of the system after performing a selected action. In addition, we further assume that the buffer state changes from the current state to the PDS, and, afterwards, the channel state and power management state change from the PDS to the next state.

#### Table 1: Classification of the dynamic environment.

| Determined | Power management state transition \( P_f(y) \) |
| Stochastic | Energy consumption (8) |
| Known | Channel state transition probability \( p(k + 1, k) \) |
| Stochastic | Packets arrival probability \( p_f(l) \) |
| Unknown | Holding cost (12) |

#### Figure 5: PDS model.
Defining the PDS set $\tilde{s}_n(\tilde{b}_n, \tilde{h}_n, \tilde{x}_n) \in \tilde{S}$, therefore, the system probability function can be organized as

$$p(s', s) = \tilde{p}_n(p(s, a), s'),$$

(16)

where $p : S \times A \rightarrow \tilde{S}$ is the transition probability from the current state to the PDS, which decides the known impacts of the performed action $a$. The transition probability from PDS to the next state is defined as $\tilde{p}_n : S \times S \rightarrow [0, 1]$, which determines the stochastic impacts caused by the action $a$. The design objective for PDS learning is to obtain an optimal action ($a^*)$ to maximize the long-term $Q$ value denoted by $Q^*(s_n, a_n)$. Define the PDS value function for PDS learning algorithm as

$$\tilde{Q}^*(\tilde{s}) = \tilde{r}(\tilde{s}) + \gamma \sum_{s'} \tilde{p}_n(s' | s, a) Q^*(s'),$$

(17)

$$Q^*(s) = \min_{a \in A} \left\{ r(s, a) + \gamma \sum_{s'} p(s' | s, a) Q^*(s') \right\},$$

(18)

where $\tilde{r}(\tilde{s})$ is the immediate reward obtained from PDS to the next state. Meanwhile, the immediate reward obtained from the current state to the PDS is denoted by $r(s, a)$. The discount factor $\gamma$ ($0 \leq \gamma \leq 1$) is the level of "foresight" in making decisions.

The optimal scheme can be calculated by the following formula in traditional $Q$ learning [19]:

$$\pi^*_Q(s) = \min_{a \in A} Q^*(s, a)$$

$$= \min_{a \in A} \left\{ r(s, a) + \gamma \sum_{s'} p(s' | s, a) Q^*(s') \right\},$$

(19)

where $r(s', a)$ is the reward obtained by taking action $a$ in state $s$. From the proof described in Appendix A, the optimal strategy for PDS algorithm can be calculated by the following formula:

$$\pi^*_\text{PDS}(s) = \min_{a \in A} \left\{ r(s, a) + \sum_{s'} p(s' | s, a) \tilde{Q}^*(\tilde{s}) \right\}.$$  

(20)

Although the PDS learning can reduce action exploration by using the determined information, the action does not balance the trade-off between the exploration and exploitation. To overcome the problem, we propose an HE-PDS learning algorithm that uses heuristic function and evaluation function to improve the algorithm performance. Specifically, the heuristic function stands for the importance when executing an action and the evaluation function for the feasibility. Thus, the optimal scheme can be written as follows:

$$\pi(s) = \begin{cases} \arg \min_{a_n} \left[ Q(s_n, a_n) + \epsilon H(s_n, a_n) \right] + \omega E(s_n, a_n), & \text{if } q \leq p, \\ a_{random}, & \text{otherwise}, \end{cases}$$

(21)

where $a_{random}$ is an action randomly chosen among the available action set $A$, which means that a nonoptimal action is intentionally selected to obtain the information of the unknown state. Besides, $\epsilon$ and $\omega$ are used to control the influence of the heuristic function and evaluation function, respectively; $q$ is a random value in the interval $(0, 1)$. The trade-off between exploration and exploitation is controlled by $p$ ($0 \leq p \leq 1$). Specifically, if $p$ is larger, the random selection probability is smaller. The heuristic function $H_n(s_n, a_n)$ is used to affect the choice of the actions. However, since the majority of the actions cannot meet the optimal requirements, we use the evaluation function $E_n(s_n, a_n)$ to reduce the number of the action to be selected. In order to minimize the error of the heuristic function and evaluation function, the corresponding definitions are given by

$$H_n(s_n, a_n)$$

$$= \begin{cases} \min_{a} Q(s_n, a) - Q(s_n, a_n) + \sigma, & \text{if } a_n = \pi^H(s_n), \\ 0, & \text{otherwise}, \end{cases}$$

(22)

$$E_n(s_n, a_n)$$

$$= \begin{cases} Q(s_n, a_n) + \epsilon H_n(s_n, a_n), & \text{if fail}, \\ -\min_{a} (Q(s_n, a), -\epsilon H_n(s_n, a)), & \text{otherwise}, \end{cases}$$

(23)

where $\sigma$ is a small real value and $\pi^H(s_n)$ is the action suggested by the heuristic policy. In order to ensure the validity of the exploration process for all state-action pairs, simulated annealing algorithm is used similar to [20]. Thus, the probability that the action $a$ is executed in the current state is given by

$$p(a_n = a_{n+1} | s) = \frac{\exp(Q(s_n, a_{n+1})/\tau_n)}{\sum_{a \in A} \exp(Q(s_n, a)/\tau_n)},$$

(24)

where $\tau_n$ is the temperature parameter, which controls randomness of the action selection.

In summary, the solving process of energy-efficient problem is as follows. In the $n$th slot, The HE-PDS first observes the current state $s_n$ and then, based on the observations, selects and executes an action $a_n$. Finally, the algorithm obtains immediate reward $r(s, a)$ and $r(\tilde{s})$ and enters next learning cycle. During the learning process, the $Q_{n+1}(\tilde{s}_n)$ value can be adjusted by the following formula:

$$Q_{n+1}(\tilde{s}_n) \leftarrow (1 - \alpha_n) Q(\tilde{s}_n) + \alpha_n \left[ \tilde{r}(\tilde{s}_n) + \gamma \min_{a \in A} Q_n(s_n, a_n) \right],$$

(25)

where $\alpha_n$ is the learning rate. $Q_{n+1}(s_n, a)$ can converge to the optimal $Q^*(s, a)$ when the sequence of learning rates $\alpha_n$ meets $\sum_{n=0}^{\infty} \alpha_n = 1$, $\sum_{n=0}^{\infty} (\alpha_n)^2 < \infty$ and the maximum errors of $Q_n(s, a)$ and $Q^*(s, a)$ are bounded. The proof can be found in Appendix B.
5. Numerical Results and Discussion

In this section, we will compare the performance of the proposed algorithm with that of the traditional Q learning and PDS learning algorithm. In the numerical computation, we assume that the bits can be mapped into QAM symbols by Gray code in physical layer similar to [8, 11]. The buffer length is $B = 25$ packets and the packet length is equal to $l = 5000$ bits. Assume that the channel transition distribution is known. In particular, the channel state and its transition probability are described in Table 2 similar to [10]. The noise power density $N_0/2$ is set to $10^{-11}$ Watt/Hz. We let the channel bandwidth $W$ be equal to symbol rate ($W = 1/T_s$), where $T_s$ is the duration of one MQMA symbol and $1/T_s = 500 \times 103$ symbol/second.

In the typical 802.11 $a/b/g$ applications, we let $P_{idle}$ be 0.05 Watt and let both $P_{on}$ and $P_{tr}$ be 0.31 Watt. In addition, the BER is set to $[2, 4, 8, 16, 32] \times 10^{-8}$ and the number of transmitted packets per time slot can be from 1 to 10. Thus, the state-action pairs number in our simulation is $7 \times 26 \times 2 \times 5 \times 11 \times 2$. Besides, the other parameters are set as follows: $\Delta t = 1$ ms, $\gamma = 0.98$, $\epsilon = \omega = 1$, $P = 0.78$, $\sigma = 0.005$, and $\tau = 5000$.

5.1. Performance Comparison under the Fixed Delay and Throughput Constraints. Figure 6 compares the cumulative average costs for 80000 time slots under the maximum delay ($4/B$ packets) and throughput ($0.1/B$ packets) constraints.

In each subgraph, horizontal coordinate represents the simulation time slot, and the vertical coordinate denotes the cumulative average total cost, delay overhead, throughput cost, and energy consumption in the corresponding slot, respectively. From (a), (b), and (c), we observe that the HE-PDS algorithm and PDS algorithm reduce their cumulative average total costs, cumulative average delay overhead, and cumulative average throughput cost by around ten times compared to the Q algorithm. In addition, the three metrics of the HE-PDS algorithm decrease about 8%, 10%, and 9% than PDS algorithm, respectively. Furthermore, as described in (d), the Q algorithm has lower energy consumption at the beginning of the simulation, but its costs increase and stabilize after about 40000th time slot. Since there has been no experience about the environment, the PDS strategy and HE-PDS strategy have larger power costs at the start of the simulation; however, the costs decrease over increasing the simulation time. In addition, the PDS consumes higher power costs than HE-PDS since HE-PDS balances the trade-off between exploration and exploitation which results in a sharper consumption decline in (d).

5.2. Performance Comparison under Various Delay Limits. To validate the performance of the HE-PDS algorithm for different delay limits, taking values of $[3, 4, 5, 6, 7, 8, 9, 10, 11]/B$ packets, respectively, Figure 7 shows the delay-energy trade-off obtained by these three algorithms. From Figure 7, we observe that the power costs of the HE-PDS algorithm decrease about 50% and 28% than the traditional Q algorithm and PDS algorithm, respectively. We also observe that the power costs of all these algorithms decrease as the delay constraint values increase. Besides, the Q algorithm gets into steady state at the 9/B packet/slot and the PDS will get it at the 8/B packet/slot; however, the times that are required to enter the steady state is significantly reduced to 5/B packet/slot for HE-PDS algorithm. This suggests that HE-PDS has an obvious advantage in energy-efficient under various delay constraints.

5.3. Performance Comparison under Various Throughput Limits. To verify the performance under various throughput limits, Figure 8 shows the performance comparison under the throughput limits $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]/B$ packets for 80000 time slots simulation. From Figure 8, it is clear that the proposed HE-PDS algorithm can find the optimal policy at the 0.5/B packet/slot which reduces

---

### Table 2: Channel states and transition probabilities.

| Channel states | $\varphi_k$ | $p(k, k - 1)$ | $p(k, k)$ | $p(k, k + 1)$ |
|----------------|------------|---------------|-----------|---------------|
| 1              | 0.1068     | 0             | 0.8387    | 0.1613        |
| 2              | 0.2301     | 0.1613        | 0.6669    | 0.1718        |
| 3              | 0.3760     | 0.1718        | 0.6612    | 0.1670        |
| 4              | 0.5545     | 0.0670        | 0.6841    | 0.1489        |
| 5              | 0.7847     | 0.1489        | 0.7330    | 0.1811        |
| 6              | 1.0990     | 0.1811        | 0.8096    | 0.0723        |
| 7              | 1.6636     | 0.0723        | 0.9277    | 0             |

Algorithm 1: HE-PDS learning algorithm.

1. Initialization: Simulation times $N$, $n = 0$, PDS value function $\tilde{Q}_n$, heuristic function $H(s_n, a_n)$, evaluation function $E(s_n, a_n)$.
2. While $(n < N)$ do
3. Observe the current state $s_n$.
4. Select and execute action $a_n$ obtained from (21).
5. Observe the immediate reward $r(\tilde{s})$ and the next state $a_n + 1$.
6. Update the functions of $H_n(s_n, a_n)$, $E_n(s_n, a_n)$ and Q value $\tilde{Q}_{a_n}(\tilde{E}_n)$ by (22), (23) and (25), respectively.
7. Set $n = n + 1$.
8. End while

4.2. The Procedure of the HE-PDS Learning Algorithm. According to the analysis stated above, the working procedure of HE-PDS learning algorithm is summarized in Algorithm 1.
Figure 6: Performance comparison under the constraint of fixed delay and throughput. (a) Cumulative average total cost. (b) Cumulative average delay overhead. (c) Cumulative average throughput cost. (d) Cumulative average energy consumption.

Figure 7: Delay-energy trade-off for different algorithms.

Figure 8: Throughput-energy trade-off for different algorithms.

about 0.2/B packet/slot compared with the other two algorithms. In addition, Figure 8 further confirms that the proposed algorithm significantly outperforms the Q and PDS algorithms since HE-PDS can decrease the power costs about 100% and 56% compared to the Q algorithm and PDS algorithm under the different throughput constraints.

Obviously, this observation is in accordance with formula (14).

5.4. Algorithm Convergence Analysis. In this section, we evaluate how the parameter $\gamma$ will affect the convergence of the HE-PDS energy-efficient algorithm. $\gamma$ will be set to 0.98 and 0.85, respectively. We also set $T$ to $4/B$ packets.
and \( D \) to 0.1/B packets. As can be seen in Figure 9, the results of the energy consumption under the fixed delay and throughput constraints and different \( \gamma \) are illustrated in (a). The convergence fluctuations of these algorithms are shown in (b) and (c). For simplicity of illustration, we define the relative fluctuation function at the time slot \( n \) as 

\[
\psi(n) = \log_M(\|\rho(n+1) - \rho(n)\|/\|\rho(n+1)\|),
\]

where \( \rho \) is the energy consumption and \( M \) is a real value. Therefore, the smaller \( \psi \) value will reflect smaller energy fluctuation which means faster convergence speed. From Figure 9(a), we observe that the proposed algorithm converges with lower energy consumption than the other two algorithms. For example, when \( \gamma \) is 0.98, HE-PDS algorithm can converge with approximate energy consumption value of 170 mJ, while Q and PDS algorithms converge with energy consumptions of about 300 mJ and 290 mJ, respectively. In addition, when \( \gamma \) is reduced to 0.85, the proposed algorithm can converge with energy consumption value by 70 mJ lower than the other two algorithms, both which converge with the same consumption of about 220 mJ. Furthermore, as shown in (b) and (c), our proposed algorithm can obtain the lowest relative fluctuation values, which means that HE-PDS has the fastest convergence rate. Specifically, the \( \psi \) value of HE-PDS algorithm is smaller than the PDS and Q algorithms about 15% and 17%. This improvement of the performance is due to the fact that our proposed algorithm explicitly uses the heuristic function and evaluation function to effectively reduce the number of actions to be chosen. Consequently, the relative fluctuation results confirm that HE-PDS algorithm can achieve the obvious convergence improvement.

### 6. Conclusion

In this paper, we investigated the impacts of the cooperative communications and designed cooperative cross-layer algorithm on energy-efficient policy in wireless networks while subjected to both transmission delay and throughput constraints. Given the dynamic buffer, time-varying channel states, and system-level power consumption in a point to point transmission environment, the problem is formulated as a CMDP and further converted into an UMDP by Lagrange multiplier. We propose an HE-PDS learning algorithm based on the determinate state information to achieve an optimal energy-efficient strategy by using the heuristic function and evaluation function. Furthermore, the performance of different energy-efficient strategies is compared and the proposed scheme is verified through simulations. Through the discussions, we highlight that the proposed algorithm has much better energy-efficient performance and faster convergence speed than the other typical state-of-the-art schemes.
Appendices

A. The Optimal Strategy of PDS

Theorem A.1 \(\pi^*_Q(s) = \pi^*_{PDS}(s)\). As we know, from formula (18), the optimization function is as follows:

\[
\pi^*_Q(s) = \min_{a \in A} \left\{ r(s', s, a) + \gamma \sum_{s'} p(s' | s, a) Q^*(s') \right\},
\]

(A.1)

where \(r(s', s, a)\) is the reward gained from the current state \(s\) to the next state \(s'\) when action \(a\) is executed. It contains two parts, the first one \(r(s, a)\) is the reward acquired from \(s\) to the PDS and \(\bar{r}(s)\) is the reward acquired from PDS to \(s'\).

Thus, formula (18) can be rewritten as

\[
\pi^*_Q(s) = \min_{a \in A} \left\{ r(s, a) + \sum_{s'} p(s | s, a) \bar{r}(s) + \gamma \sum_{s'} p(s' | s, a) Q^*(s') \right\}.
\]

(A.2)

From (16), we know for the transition probability that the state transfers from \(s\) to \(s'\), and we can further decompose (A.2) as

\[
\pi^*_Q(s) = \min_{a \in A} \left\{ r(s, a) + \sum_{s'} p(s | s, a) \bar{r}(s) + \gamma \sum_{s'} p(s' | s, a) Q^*(s') \right\}.
\]

(A.3)

Substituting (17) into the above equation,

\[
\pi^*_Q(s) = \min_{a \in A} \left\{ r(s, a) + \sum_{s'} p(s | s, a) Q^*(s') \right\} = \pi^*_{PDS}(s).
\]

(A.4)

Therefore, the optimal strategy of \(\pi^*_{PDS}(s)\) and \(\pi^*_Q(s)\) is equivalent.

B. The Convergence of the HE-PDS Learning Algorithm

Theorem B.1. In the determined MDP system, let all the immediate rewards be bounded, which means that \(\forall (s, a) r(s', | s, a) \leq c\). Besides, all the state-action pairs can be visited many times, moreover, when the learning rate satisfies the following prerequisites. Then, the HE-PDS learning algorithm converges to the optimal value function

\[
\sum_{n=0}^{\infty} \alpha_n = \infty, \quad \sum_{n=0}^{\infty} (\alpha_n)^2 < \infty.
\]

(B.1)

Based on formula (17), we know that

\[
Q(s, a) = \bar{r}(s) + \gamma \min_{a'} Q^*(s', a').
\]

(B.2)

Define \(f([{Q}]) = \bar{r}(s) + \gamma \min_{a'}\{Q_n(s', a') - Q(s, a)\}\). Then, we can rewrite formula (25) as \(Q_{n+1}(s, a) = Q_n(s, a) + \alpha_n f([{Q}])\).

After the \((n + 1)th\) learning, the error of the Q value table can be calculated as

\[
Q_{n+1}(s, a) - Q^*(s, a) = Q_n(s, a) + \alpha_n f([{Q}]) - \gamma \min_{a'}\{Q_n(s', a') - Q(s, a)\}.
\]

(B.3)

Therefore,

\[
|Q_{n+1}(s, a) - Q^*(s, a)| \leq |1 - \alpha_n| |Q_n(s, a) - Q^*(s, a)|
\]

(B.4)
Define that $\min_{a_1} Q_n(s', a')$ and $Q^*_n(s', a')$ is the optimal $Q$ value when the action $a_1$ and $a_2$ are executed, respectively. Then, we can deduce that

$$\left| \min_{a_1} Q_n(s', a') - \min_{a_1} Q^*_n(s', a') \right|$$

$$= \left| Q_n(s', a_1) - Q^*_n(s', a_1) \right|$$

$$\leq \min_{a_2 \in \{a_1, a_2\}} \left| Q_n(s', a') - Q^*_n(s', a') \right|$$

$$\leq \min_{a' \in A} \left| Q_n(s', a') - Q^*_n(s', a') \right|.$$ (B.5)

We further define that $\delta_n(s, a) = |Q_n(s, a) - Q^*(s, a)|$ is the maximum error after $n$ updates. Thus,

$$\left| Q_{n+1}(s, a) - Q^*(s, a) \right|$$

$$= \delta_{n+1}(s, a)$$

$$\leq |1 - \alpha_n| \delta_n(s, a) + |\alpha_n| \gamma \min_{a'} \delta_n(s, a')$$

$$\leq \left( |1 - \alpha_n| + |\alpha_n| \gamma \right) \min_{a'} \delta_n(s, a').$$ (B.6)

Note that we have assumed that all the state-action pairs can be visited by the current policy. Therefore, we have

$$|1 - \alpha_n| + |\alpha_n| \gamma < 1, \quad 0 < \alpha_n < \frac{2}{1 + \gamma}.$$ (B.7)

This yields the fact that

$$\left| Q_{n+1}(s, a) - Q^*(s, a) \right| < \min_{a'} \delta_n(s, a)$$

$$= \left| Q_n(s, a) - Q^*(s, a) \right|.$$ (B.8)

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61072138 and no. 61379005) and Southwest University of Science and Technology (12zx7127).

### References

[1] Y. Chen, S. Zhang, S. Xu, and G. Y. Li, "Fundamental trade-offs on green wireless networks," IEEE Communications Magazine, vol. 49, no. 6, pp. 30–37, 2011.

[2] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," IEEE Transactions on Communications, vol. 60, no. 1, pp. 220–230, 2012.

[3] A. Munir and A. Gordon-Ross, "An MDP-based dynamic optimization methodology for wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 4, pp. 616–625, 2012.

[4] A. T. Hoang and M. Motani, "Cross-layer adaptive transmission with incomplete system state information," IEEE Transactions on Communications, vol. 56, no. II, pp. 1961–1971, 2008.

[5] O. Gungor, J. Tan, C. E. Koksal, H. El Gamal, and N. B. Shroff, "Joint power and secret key queue management for delay limited secure communication," in Proceedings of the IEEE INFOCOM, San Diego, Calif, USA, March 2010.

[6] X. Li, H. He, and Y. D. Yao, "Reinforcement learning based adaptive rate control for delay-constrained communications over fading channels," in Proceedings of the International Joint Conference on Neural Networks, 2010.

[7] N. Salodkar, A. Bhorkar, A. Karandikar, and V. Borkar, "An online learning algorithm for energy efficient delay constrained scheduling over a fading channel," IEEE Journal on Selected Areas in Communications, vol. 26, no. 4, pp. 732–742, 2008.

[8] G. Xu, H. Jiang, C. Liu, and Y. Wang, "A heuristic evaluation PDS algorithm for energy-efficient delay constrained scheduling over wireless communication," in Proceedings of the IEEE International Conference on Communications (ICC ‘12), pp. 6013–6017, Ottawa, Canada, June 2012.

[9] X. Zhong and C. Xu, "Energy-efficient wireless packet scheduling with quality of service control," IEEE Transactions on Mobile Computing, vol. 6, no. 10, pp. 1158–1170, 2007.

[10] A. T. Hoang and M. Motani, "Cross-layer adaptive transmission: optimal strategies in fading channels," IEEE Transactions on Communications, vol. 56, no. 5, pp. 799–807, 2008.

[11] N. Mastronarde and M. van der Schaar, "Fast reinforcement learning for energy-efficient wireless communication," IEEE Transactions on Signal Processing, vol. 59, no. 12, pp. 6262–6266, 2011.

[12] Q. Liu, S. Zhou, and G. B. Giannakis, "Queueing with adaptive modulation and coding over wireless links: cross-layer analysis and design," IEEE Transactions on Wireless Communications, vol. 4, no. 3, pp. 1142–1153, 2005.

[13] S. I. Hussain, M. O. Hasna, and M. Alouini, "Performance analysis of selective cooperation with fixed gain relays in Nakagami-m channels," Physical Communication, vol. 5, no. 3, pp. 272–279, 2012.

[14] G. Miaow, N. Himayat, Y. Li, and A. Swami, "Cross-layer optimization for energy-efficient wireless communications: a survey," Wireless Communications and Mobile Computing, vol. 9, no. 4, pp. 529–542, 2009.

[15] A. Kumar, Enhancing coexistence, quality of service, and energy performance in dynamic spectrum access networks [Ph.D. thesis], 2011.

[16] E. Chung, L. Benini, A. Bogliolo, H. Y. Lu, and G. de Micheli, "Dynamic power management for nonstationary service requests," IEEE Transactions on Computers, vol. 51, no. 11, pp. 1345–1361, 2002.

[17] H. Naqvi, S. Berber, and Z. Salcic, "Energy efficient collaborative communication with imperfect phase synchronization and Rayleigh fading in wireless sensor networks," Physical Communication, vol. 3, no. 2, pp. 119–128, 2010.

[18] J. Zhu, J. Wang, T. Luo, and S. Li, "Adaptive transmission scheduling over fading channels for energy-efficient cognitive radio networks by reinforcement learning," Telecommunication Systems, vol. 42, no. 1-2, pp. 123–138, 2009.
[19] S. Adam, L. Buşoniu, and R. Babuška, “Experience replay for real-time reinforcement learning control,” *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 42, no. 2, pp. 201–212, 2012.

[20] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, “A simulated annealing-based multiobjective optimization algorithm: AMOSA,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269–283, 2008.
Submit your manuscripts at http://www.hindawi.com