sensors

MDPI

Article

A Cross-Layer Routing Protocol Based on Quasi-Cooperative Multi-Agent Learning for Multi-Hop Cognitive Radio Networks

Yihang Du [1], Chun Chen [2,*], Pengfei Ma [2] and Lei Xue [1]

[1] National University of Defense Technology, Shushan District, Hefei 230000, China; yuri_wolfdyh@163.com (Y.D.); xwdxh1965@163.com (L.X.)

[2] Army Academy of Artillery and Air Defense, Shushan District, Hefei 230000, China; xiaoma.2002@163.com

* Correspondence: chen641610939@163.com; Tel.: +86-551-6592-1276

Received: 7 November 2018; Accepted: 28 December 2018; Published: 3 January 2019

check for updates

Abstract: Transmission latency minimization and energy efficiency improvement are two main challenges in multi-hop Cognitive Radio Networks (CRN), where the knowledge of topology and spectrum statistics are hard to obtain. For this reason, a cross-layer routing protocol based on quasi-cooperative multi-agent learning is proposed in this study. Firstly, to jointly consider the end-to-end delay and power efficiency, a comprehensive utility function is designed to form a reasonable tradeoff between the two measures. Then the joint design problem is modeled as a Stochastic Game (SG), and a quasi-cooperative multi-agent learning scheme is presented to solve the SG, which only needs information exchange with previous nodes. To further enhance performance, experience replay is applied to the update of conjecture belief to break the correlations and reduce the variance of updates. Simulation results demonstrate that the proposed scheme is superior to traditional algorithms leading to a shorter delay, lower packet loss ratio and higher energy efficiency, which is close to the performance of an optimum scheme.

Keywords: cognitive radio; cross-layer routing protocol; experience replay; quasi-cooperative multi-agent learning; stochastic game

1. Introduction

Cognitive radio is a technology that is used to settle the problem of spectrum scarcity by enabling secondary users (SUs) to access the licensed spectrum of primary users (PUs) in a dynamic and non-interfering manner. It mediates the contradiction between the regulation and frequency utility via time and spatial multiplexing [1]. One of the major challenges in the design of cognitive radio networks (CRNs) is radio resource management, which efficiently handles the spectrum mobility and quality of service (QoS) requirements of different services and different nodes [2]. Consequently, techniques for resource management have been receiving considerable attention [3,4].

However, most research in the CRN field has focused on the direct transmission network. The research community has focused on applying the cognitive paradigm in multi-hop networks to supply more spectrum resources for a range of applications [5,6]. To fully study the characteristics of multi-hop CRN, it is critical to coordinate route selection with radio resource management and design a cross-layer routing protocol for high-spectrum utility. The regional difference in a multi-hop path leads to the difference in the frequency band for all SUs [7]. The design of efficient and robust spectrum-aware routing protocol is challenging due to the absence of topology information and spectrum dynamics in CRNs.

Traditional protocols used decompose the cross-layer design problem into two sub-problems including route selection and resource management. The sub-problems were optimized separately to reduce the system calculation complexity. Ding et al. [8] proposed a distributed and localized scheme for joint relay selection and channel assignment. A cooperative strategy for the optimization problem based on real-time decentralized policies was studied. However, full cooperation was difficult to achieve for SUs in CRN due to its uncertainty. A centralized routing protocol was proposed by Lai et al. [9], which expanded the paths hop-by-hop and discarded unnecessary paths. Then the resource management problem was formulated into an optimization problem with the aforementioned objective and restriction. Nevertheless, the centralized method was more complex in calculation and less flexible than distributed schemes. An economic framework was presented by Amini et al. [10] that integrated route selection and spectrum assignment for the improvement of QoS performance in cognitive mesh networks. The decomposed model allowed for a decentralized implementation of routing and spectrum allocation, which increased the robustness of the algorithm. These works aimed at finding a fixed path from the source nodes to the destination nodes [11]. However, fixed routing strategies have two disadvantages: priori knowledge of topology and spectrum dynamics is required, which is hard to obtain in multi-hop CRN, and fixed protocols may become invalid due to the dynamic and uncertain nature of frequency.

In order to realize real cognition, a cognitive radio (CR) should be capable of learning and reasoning [12]. Machine learning technology has been widely used to address dynamic channel statistics in CRNs. Raj et al. [13] proposed a two-stage reinforcement approach to select a channel via a multi-armed bandit, and then predict how long the channel would remain unoccupied. Its sensing was more energy efficient and achieved higher throughput by saving on spectrum detection. Al-Rawi et al. [14] developed the cognitive radio Q-routing algorithm, which adopted reinforcement learning (RL) method to enable flexible and efficient routing decisions. Single-agent reinforcement learning with limited capacity was adopted by Raj et al. [13] and Al-Rawi et al. [14]. Nevertheless, learning strategy with multiple agents is more suitable for solving complicated problem in multi-hop CRNs. Multi-agent learning approaches in CRN have drawn the interest of researchers for their superior performance. A conjecture-based multi-agent Q-learning scheme was presented by Chen et al. [15] to execute power adaption in a partially observable environment. However, route selection in multi-hop CRN was not considered in this work. Pourpeighambar et al. [16] modeled the routing problem as a stochastic game (SG). Then, the SG was solved through a non-cooperative multi-agent learning method in which each secondary user (SU) speculated other nodes' strategies without acquisition of global information. The current conjecture belief was determined only by the last one in [16], which caused strong correlations between the samples. Power adaption was not considered when solving the routing problem, which would influence power efficiency and the routing decision.

In our previous work [17], we designed a single-agent based intelligent joint routing and resource assignment scheme for CRN to achieve the maximum cumulative rewards. In this paper, we adopt a quasi-cooperative multi-agent learning scheme for routing and radio resource management, which is more efficient than the single-agent strategy in multi-hop CRN. The scheme tries to achieve the lowest end-to-end delay and improve energy efficiency with finite information exchange between competing SUs. Our contributions are summarized as follows:

(i) In order to jointly capture the end-to-end latency and power efficiency, a comprehensive utility function is designed to form a reasonable tradeoff between the two as well as accommodate the maximal transmission latency requirement. Queuing theory is adopted to analyze single-hop latency and provide a theoretical basis for our cross-layer routing protocol design.

(ii) A quasi-cooperative multi-agent learning framework is presented to solve the cross-layer design problem where every SU node speculates other nodes' strategies from finite information exchange with the previous nodes. The convergence of the quasi-cooperative learning scheme is proven.

(iii) For the purpose of further enhancing performance, experience replay is applied to the update of conjecture belief, which allows for greater data efficiency by using the historical conjectures and breaks the correlations to reduce the variance of updates.

The remainder of this paper is organized as follows: the system model is presented in Section 2. Section 3 models the cross-layer design problem as a SG. The quasi-cooperative multi-agent learning scheme for the cross-layer routing protocol is proposed in Section 4. Section 5 demonstrates the simulation results. Finally, the paper is concluded in Section 6. In addition, summary of acronyms used in this paper is listed in Table 1.

Table 1. List of Acronyms.

Abbreviation	Full Name
SUs	Secondary Users
PUs	Primary Users
CRN	Cognitive Radio Networks
QoS	Quality of Service
CR	Cognitive Radio
SG	Stochastic Game
DTC	Data Transmission Channel
CCC	Common Control Channel
PDF	Probability Density Function
TL	Transmission Latency
PCR	Power Consumption Ratio
AWGN	Additive White Gaussian Noise
SINR	Signal-to-Interference plus Noise Ratio
RL	Reinforcement Learning
MSNE	Mixed-Strategy Nash Equilibrium
PLR	Packet Loss Ratio

2. System Model

A multi-hop CRN comprising M PUs and N SUs is considered. Specific spectrum bands are assigned to PUs according to the fixed spectrum allocation regulation. SUs occupy no licensed channels and transmit data opportunistically when finding that frequency bands are not held by the PUs. Every SU node n_i has a set of available channels that consist of Data Transmission Channel (DTC) and Common Control Channel (CCC). The DTC is used for data transmission and SU n_i's DTC is represented as $C_i = \{c_1, c_2, \ldots, c_m\}$; whereas the CCC is used by SUs to exchange the negotiation. At any time, a directed communication link can be constructed between SU n_i and n_j if at least one common DTC $c \in C_i \cap C_j$ exists. The network model is shown in Figure 1. In the networking scenarios, the multi-hop CRN coexists with two centralized Primary User (PU) networks. As Figure 1 shows, the source SU generates data packets and sends them to the destination node in multi-hop manner through intermediate SUs. Each PU communicates with the PU base station using a licensed frequency, and intermediate SUs in each hop transmit data via the PU channel when the spectrum band is idle.

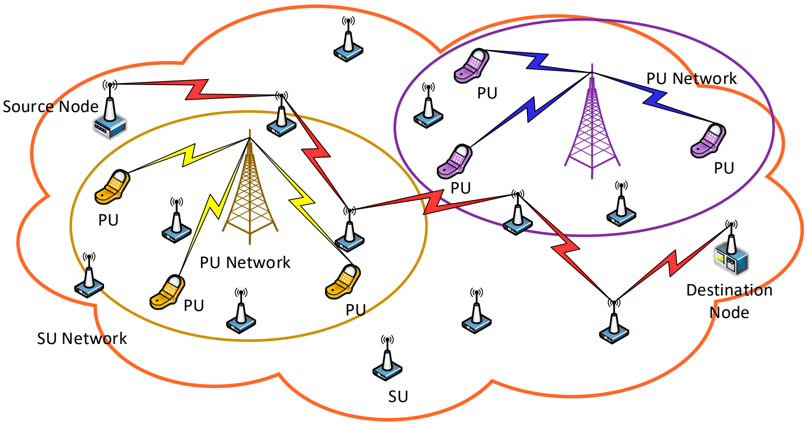


Figure 1. Multi-hop cognitive networking scenarios.

We assume that every node in the multi-hop CRN maintains a queuing buffer for the storage of SU packets. For data flow f generated from source node n_s, packet arrivals are considered as a stationary Bernoulli process with mean λ_s^f that is independent and identical at all time slots [18]. In addition, every SU node has respective queues for each traffic flow, and the packet arrival process of every data stream is independent from each other.

The PUs' occupation model is considered as an ON/OFF process [19]. The probability density function (PDF) of the OFF periods (when PUs do not occupy the channel) is shown as follows:

$$f(t) = \begin{cases} \theta_d e^{-\theta_d t} & t \geq 0 \\ 0 & t < 0 \end{cases}, \tag{1}$$

where θ_d is departure rate of the PU, and $f(t)$ represents the idle probability of PU channel at time step t. Accordingly, the probability that idle period of PU channel is longer than duration τ is denoted as:

$$P(t \geq \tau) = \int_{\tau}^{\infty} f(t)dt = e^{-\theta_d \tau}, \tag{2}$$

where τ is the duration that PU channel is idle. Then, the probability of the collision between PU and SU in the duration τ (i.e., the probability of PU reoccupying the spectrum band in the duration τ) is given by:

$$P_{collision} = 1 - P(t \geq \tau) = 1 - e^{-\theta_d \tau}. \tag{3}$$

We use the analysis described in [20] to calculate the PU departure rate $\theta_d(\mu, \sigma)$ with expected mean μ and deviation σ. The spectrum statistic that is parameterized in [21] changes slowly so that it is assumed to be almost static in this work. Every SU can only locate its own position through some positioning equipment.

3. Formulation for Joint Design Problem

In this section, to minimize transmission latency and ensure power efficiency, a comprehensive utility function is designed to create a reasonable tradeoff between the two with a delay constraint. Then a measurement called responsibility rating is introduced for power assignment and reducing the action space of agents. On the basis of the above considerations, the cross-layer routing problem is modeled as a SG.

3.1. Comprehensive Utility Function

To guarantee the QoS performance of cross-layer routing, a comprehensive utility function is applied to integrate transmission latency and energy efficiency. A multi-hop network must reduce the transmission latency so that the packet loss rate decreases and the routing stability improves. In addition, the requirement of high power efficiency is also a critical factor for energy-sensitive applications. For instance, grid monitoring and control applications only have limited data to send but demand real-time delivery [22]. Energy-constrained networks that traditionally operate powered by batteries are sensitive to power consumption and thus face an inherent challenge in energy efficiency, but are not sensitive to latency. However, some high-level services such as video and audio demand both real-time data transmission and high power efficiency [23]. Low transmission latency demands high power consumption, whereas excessive energy conservation may cause poor QoS performance and result in a long end-to-end delay. There is an inherent tradeoff between the transmission latency and energy consumption. Therefore, a utility function is designed that jointly captures the end-to-end delay and energy efficiency while making a reasonable tradeoff between the two. The resultant design is as follows:

$$r_i^t = -\log_2(\alpha \cdot u_{TD,i} + \beta \cdot u_{PCR,i}), \tag{4}$$

where $u_{TD,i}$ accounts for the single-hop transmission latency (TL), and $u_{PCR,i}$ denotes power consumption ratio (PCR) for SU_i, which will be elaborated in following sections; and α and β are parameters that adjust the tradeoff between transmission delay and energy efficiency, respectively. A larger α increases the weight of the transmission delay in the utility function, whereas larger β emphasizes the power consumption, and vice versa. The logarithmic operation is used for compressing large values of $\alpha \cdot u_{TD,i} + \beta \cdot u_{PCR,i}$ to a relatively small range. We can see from Equation (4) that the larger the TL or PCR is, the lower the utility function r_i^t becomes. This results in little reward for the agent so that it will explore more efficient actions to achieve minimal latency and energy expenditure.

3.1.1. Transmission Delay

The transmission delay consists of queuing waiting time and data transmission time. Firstly, the queuing waiting time is computed based on the packet arrival and service rates of every SU in a multi-hop CRN [18]. The calculation method of queuing waiting time δ_i was proposed in [24], where the packet arrival and service rates of every SU were used to compute the queuing waiting time. In our work, we further combine this method with the concept of strategy $\zeta_k(s_k,\ a_k)$ in RL to obtain the latency for queuing in SU's buffer. A packet from flow f is placed in the SU n_i's queuing buffer at time step t if:

(1) One of the node n_i's neighboring nodes selects node n_i as its next hop and transmits data via an available channel c;
(2) Channel c is idle during time step t, and
(3) There is at least one packet in the queue of the preceding SU node to communicate with node n_i

Therefore, the arrival rate at SU n_i is the joint probability of all events above, which can be represented as the product of these events' probabilities due to the independence among themselves:

$$\lambda_i^t = \sum_{f\in F}\sum_{k\in L_i}\sum_{c\in C_k} \zeta_k(s_k,\ a_k)\cdot\frac{\lambda_f}{\mu_f}\cdot\alpha_c\cdot\left(1-P_{ki}^{out}\right), \tag{5}$$

where $\mathbf{F}$ is the set of all data streams; $\mathbf{L_i}$ is the set of SU_i's previous nodes that select node n_i as their next hop; $\mathbf{C_k}$ is the set of available channels of node $n_k \in \mathbf{L_i}$; $\zeta_k(s_k,\ a_k)$ is the strategy of node $n_k \in \mathbf{L_i}$, i.e., the probability of node n_k choosing action a_k, which corresponds to the next intermediate node n_i and the operating channel $c \in \mathbf{C_k}$; λ_f is the arrival rate of data flow f; and μ_f is its service rate. The probability that the queue has at least one packet can be represented as λ_f/μ_f according to the theory of discrete time Markov chain. α_c is the probability that channel c is idle and P_{ki}^{out} represents the outage probability of the link between node n_k and n_i.

Like the arrival rate, the service rate is equal to the probability of transmitting a packet successfully at SU n_i, which occurs if:

(1) SU_i selects node n_j as its next hop and transmits data via an available channel c', and
(2) Channel c' is idle during time step t.

Accordingly, the service rate at SU n_i can be calculated as the product of these two events' probability:

$$\mu_i^t = \sum_{f\in F}\sum_{j\in N_i}\sum_{c'\in C_i} \zeta_i(s_i,\ a_i)\cdot\alpha_c\cdot\left(1-P_{ij}^{out}\right), \tag{6}$$

where $\mathbf{N_i}$ is the set of SU_i's neighboring nodes; $\mathbf{C_i}$ is the set of node n_i's available channels; $\zeta_i(s_i,\ a_i)$ is the strategy of node n_i, i.e., the probability of node n_i choosing action a_i, which corresponds to the next intermediate node n_j and the data transmission channel $c' \in \mathbf{C_i}$; and P_{ij}^{out} represents the outage probability of link between node n_i and n_j.

As discussed above, the arrival and service processes of every SU are Bernoulli processes with rates λ_i^t and μ_i^t, respectively. This queuing system is modeled as a Geo/Geo/1 queue [25]. Consequently, the queuing waiting time of SU n_i is calculated as:

$$\delta_i = \frac{\lambda_i^t}{\mu_i^t(\mu_i^t - \lambda_i^t)}. \tag{7}$$

Large packet size, interference of PUs and bandwidth constraint will lead to limited channel capability. So, the data transmission time has to be considered. The data transmission time of SU n_i is defined as:

$$\tau_i = R_{packet} \Big/ \left[B \cdot \log_2 \left(1 + \frac{h_{ijc} p_i}{\vartheta + \phi_{ijc}^{PU}} \right) \right], \tag{8}$$

where R_{packet} is the packet size, B is the bandwidth of DTC, h_{ijc} represents the channel gain between the node n_i and n_j, ϕ_{ijc}^{PU} denotes the PU-to-SU interference at the receiver node n_i, and ϑ is the additive white Gaussian noise (AWGN) power. Consequently, the transmission delay is calculated as:

$$u_{TD,i} = \delta_i + \tau_i = \frac{\lambda_i^t}{\mu_i^t(\mu_i^t - \lambda_i^t)} + R_{packet} \Big/ \left[B \cdot \log_2 \left(1 + \frac{h_{ijc} p_i}{\vartheta + \phi_{ijc}^{PU}} \right) \right]. \tag{9}$$

3.1.2. Power Consumption Ratio

The power consumption ratio (PCR) is the energy consumption when obtaining unit throughput. It is proposed to describe power efficiency. A low PCR means that the cognitive node expends less energy when transmitting the same size of SU packet data, which represents high energy efficiency. PCR is given by:

$$\varepsilon_i = p_i \Big/ B \cdot \log_2 \left(1 + \frac{h_{ijc} p_{\psi_i}}{\vartheta + \phi_{ijc}^{PU}} \right), \tag{10}$$

where p_i is the transmission power for node n_i.

3.2. Responsibility Rating

For power efficiency and PU protection, power assignment is considered in this work. However, the action space will be fairly large if we treat power assignment as actions in the joint optimization problem. Huge action space results in intensive computation complexity and low learning efficiency due to the maximum calculation in Q-value updating [17]. In this case, the concept called responsibility rating was introduced in our previous work [17].

SU should improve the transmitting power in its next transmission for reducing the average TL if much time has been wasted in the current transmission. If the latency of the current data transmission is sufficiently short, then the power should be lessened at the next time step to decrease energy expenditure. Based on this principle, the responsibility rating of SU n_i at time step t is given by:

$$\psi_i^{t+1} = \begin{cases} \psi_i^t + 1 & u_{TD,i} > \Lambda_{i,t}^* \\ \psi_i^t - 1 & u_{TD,i} \le \Lambda_{i,t}^* \end{cases}, \tag{11}$$

where responsibility rating ψ_i^t is a nonnegative integer corresponding to one of the transmission power levels. In addition, if $\psi_i^t = \max\{\psi_i^t\}$ and $u_{TD,i} > \Lambda_{i,t}^*$, then $\psi_i^{t+1} = \psi_i^t$; if $\psi_i^t = 0$ and $u_{TD,i} \le \Lambda_{i,t}^*$, then $\psi_i^{t+1} = 0$. $u_{TD,i}$ denotes the single-hop TL of SU n_i, and $\Lambda_{i,t}^*$ is the average value of $u_{TD,i}$ at time step t, which can be calculated in a progressive form via historical information:

$$\Lambda_{i,t}^* = \Lambda_{i,t-1}^* + \frac{1}{t}(u_{TD,i} - \Lambda_{i,t-1}^*). \tag{12}$$

Every responsibility rating ψ_i^t matches a transmission power p_i ($p_{\min} \leq p_i \leq p_{\max}$), and the association is given by:

$$p_i(\psi_i^t) = \left(1 - \frac{\psi_i^t}{|\Psi_i|}\right) p_{\min} + \frac{\psi_i^t}{|\Psi_i|} p_{\max}, \tag{13}$$

where $|\Psi_i|$ represents the size of $\{\psi_i^t\}$, and $p_{\min}$ and $p_{\max}$ are the minimal and maximal value of the transmit power, respectively. The responsibility rating not only adjustments transmitting power for high energy efficiency but compresses huge action space to reduce the computation load.

3.3. Problem Definition

In this part, the cross-layer design problem is formulated as stochastic learning processes featured by quasi-cooperative games. The quasi-cooperative game is defined by a tuple $\langle S_i, A_i, T_i, R_i \rangle_{i=1}^N$, where S_i is SU n_i's state space, A_i represents SU n_i's actions space, T_i is the state transferring probability set, and $R_i : S_i \times A_i \mapsto \Re$ specifies the reward received by SU n_i at $s_i \in S_i$ when taking action $a_i \in A_i$. SU's states, available actions and instantaneous reward are precisely defined as follows:

3.3.1. States

For SU n_i, the node state at time step t is defined as:

$$s_i^t = \{\rho_i, \psi_i^t\}, \tag{14}$$

where ψ_i^t is the responsibility rating of SU n_i, and $\rho_i \in \{0, 1\}$ is the Signal-to-Interference plus Noise Ratio (SINR) indicator that indicates whether the SINR γ_i of SU n_i is above or below the threshold γ_{th}:

$$\rho_i = \left\{ \begin{array}{ll} 1, & \text{if } \gamma_i \geq \gamma_{th} \\ 0, & \text{otherwise} \end{array} \right. , \tag{15}$$

where $\gamma_i = h_{ijc} p_{\psi_i} / \left(\vartheta + \phi_{ijc}^{PU}\right)$, p_{ψ_i} is the transmitting power of SU n_i, h_{ijc} represents the channel gain between node n_i and n_j, ϕ_{ijc}^{PU} denotes the PU-to-SU interference at n_i, and ϑ is the AWGN power. In addition, a learning episode of SU n_i terminates when $\rho_i = 0$, i.e., $s_i^t = \{0, \psi_i^t\}$ is the terminal state in the Markov chain.

3.3.2. Actions

For the joint route selection and resource management problem, an action at time step t is defined as $a_i^t = \{n_j, c_i, p_{\psi_i}\}$, where n_j is the next relay node in SU n_i's neighboring nodes set, $c_i \in \mathbf{C_i}$ denotes the DTC of node n_i, and p_{ψ_i} is the transmission power described in Equation (13). Assume that the size of neighboring node set is J, the DTC of node n_i consists of C channels, and the transmitting power is divided into P levels. The size of state space is 2 ($\rho_i = 0$ or 1) and action space is $J \times C \times P$ if the power assignment is set as the action. By applying the responsibility rating to the cross-layer design, the size of state rises to $2 \times P$, while the action space size becomes $J \times C \times 1$. Therefore, a tradeoff occurs between the size of state space and action space, which reduces the calculation complexity when updating the Q-values by compressing huge action space while controlling the size of state space in case of dimension curse.

3.3.3. Rewards

$R_i^t(s_i, a_i, \mathbf{a_{-i}})$ is the instantaneous reward when SU n_i performs action a_i in s_i and other competing SUs execute actions $\mathbf{a_{-i}}$. The considered reward function at time step t is calculated as follows:

$$R_i^t(s_i, a_i, \mathbf{a_{-i}}) = \left\{ \begin{array}{cc} r_i^t(s_i, a_i, \mathbf{a_{-i}}), & \text{if } u_{TD,i} \leq \kappa_{th} \\ 0, & \text{otherwise} \end{array} \right. , \tag{16}$$

where $\mathbf{a_{-i}} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N) \in \mathbf{A_{-i}} = \prod_{j \in N \backslash \{i\}} A_j$ is other SUs' action vector, $r_i^t(s_i, a_i, \mathbf{a_{-i}})$ is the utility function defined in Equation (4), and κ_{th} represents the maximal transmission delay threshold between SU nodes. If TL is smaller than the maximal delay threshold, the transmission is effective and the instantaneous reward is equal to the utility function. Otherwise, the agent will receive no reward.

As described in Equations (5) and (6), SU n_i needs its own strategy $\zeta_i(s_i,\ a_i)$ and the strategies of the previous nodes $\zeta_k(s_k,\ a_k)$ to calculate the transmission delay and the average reward. Since every SU node only needs local observations and information exchange with the previous nodes instead of mutual information sharing between competing SU nodes, the problem is formulated as quasi-cooperative stochastic games, which is formally defined as:

$$\begin{aligned} & \max_{a_i \in \mathbf{A_i}} R_i^t(s_i, a_i, \mathbf{a_{-i}}) \\ s.t. \quad & u_{TD,i} \leq \kappa_{th} \end{aligned} \tag{17}$$

Every agent chooses the action of route selection and spectrum access in terms of the strategy $\zeta_i(s_i,\ a_i)$, which matches the definition of mixed-strategy game. Moreover, each SU cannot acquire the global information of competing SUs due to the uncertainty of multi-hop CRN. One of the significant characteristics in Mixed-Strategy Nash Equilibrium (MSNE) is that the players cannot obtain their opponents' strategies in advance. In other words, MSNE is a rational countermeasure when the strategies of other players are uncertain. Therefore, MSNE is adopted to solve the quasi-cooperative stochastic game [26].

Definition. *A set of M strategies $(a_i^*, \boldsymbol{a}_{-i}^*)$ is an MSNE if, for every SU $n_i \in M$:*

$$R_i(a_i^*, \mathbf{a_{-i}^*}) \geq R_i(a_i, \mathbf{a_{-i}^*}),\ for\ all\ a_i \in \mathbf{A_i}. \tag{18}$$

In the following part, we study the method of speculating competing SUs' strategies only using information exchange with the previous nodes for SU n_i, and solve quasi-cooperative stochastic game through multi-agent Q-learning.

4. Joint Routing and Resource Management with Conjecture Based Multi-Agent Q-Learning

In order to introduce the quasi-cooperative multi-agent learning scheme, a brief introduction to multi-agent Q-learning is provided in Section 4.1. In Section 4.2, the Equal Reward Time-slots based Conjectural Multi-Agent Q-Learning (ERT-CMQL) is presented to solve the cross-layer routing problem. Then, the analysis and proof of its convergence is outlined in Section 4.3.

4.1. Multi-Agent Q-Learning

Among various algorithms in the RL framework, Q-learning is a practical approach adopting Q-value. Q-value is the total expected discounted reward for the pair of state-action and describes the value of choosing a particular action in a given state. It weights and ranks the probabilities of different actions, i.e., the action with a higher Q-value is more valuable and given higher selection probability, and vice versa. To achieve this object, the Boltzmann distribution is used to calculate the probability of choosing action a_i at time slot t:

$$\zeta_i^t(s_i, a_i) = \frac{e^{Q_i^t(s_i, a_i)/\eta}}{\sum_{b \in A_i} e^{Q_i^t(s_i, b)/\eta}}, \tag{19}$$

where $Q_i^t(s_i, a_i)$ is the Q-value for the pair of state-action (s_i, a_i) at time step t, η is a positive number called the temperature. The larger the temperature is, the more balanced the probability of action selection becomes, and vice versa.

M players are considered and every player is fitted with a Q-learning agent that learns its own strategy through limited cooperation with other agents. Thus, the Q-value is updated according to multi-agent Q-learning rule:

$$\begin{aligned} Q_i^{t+1}(s_i, a_i) \quad &= (1-\alpha)Q_i^t(s_i, a_i) + \alpha\left[E[R_i(s_i, \zeta_i, \boldsymbol{\zeta}_{-\mathbf{i}})] + \beta \max_{b_i \in \mathbf{A_i}} Q_i^t(s'_i, b_i)\right] \\ &= Q_i^t(s_i, a_i) + \alpha\left[E[R_i(s_i, \zeta_i, \boldsymbol{\zeta}_{-\mathbf{i}})] + \beta \max_{b_i \in \mathbf{A_i}} Q_i^t(s'_i, b_i) - Q_i^t(s_i, a_i)\right] \end{aligned} \tag{20}$$

where $\alpha \in [0,\ 1)$ is the learning rate, and β is the discount factor, and $E[R_i(s_i, \zeta_i, \boldsymbol{\zeta}_{-\mathbf{i}})]$ is the expected reward for SU n_i at time slot t considering other $M-1$ competing SUs. The variation of Q-value is proportional to the expected reward plus the difference between the target and evaluated Q-value. The detailed definition of $E[R_i(s_i, \zeta_i, \boldsymbol{\zeta}_{-\mathbf{i}})]$ is given by:

$$E[R_i(s_i, \zeta_i, \boldsymbol{\zeta}_{-\mathbf{i}})] = \sum_{(a_i, a_{-i}) \in A}\left[R_i(s_i, a_i, a_{-i}) \prod_{j \in M \backslash \{i\}} \zeta_j(s_j, a_j)\right], \tag{21}$$

where $\prod\limits_{j \in M \backslash \{i\}} \zeta_j(s_j, a_j)$ represents the joint probability of SU_i's competing SUs choosing actions $\mathbf{a_{-i}}$ in their respective states. From Equations (20) and (21), in multi-agent Q-learning, the agent needs not only its own transmission strategy but also the complete information of competing SUs' strategies $\zeta_j(j \in M \backslash \{i\})$ to update the Q-value of SU n_i. However, it is not always practical to observe other SUs' private information in multi-hop CRN with finite cooperation. Therefore, designing a quasi-cooperative multi-agent learning scheme, which only needs private strategy and information exchange with its previous nodes, is challenging.

4.2. Equal Reward Time-Slots Based Conjectural Multi-Agent Q-Learning

Multi-agent learning strategy is more practical for solving the joint design problem in multi-hop CRN. The main drawback of establishing a multi-agent learning framework is the demand for complete information of competing SUs. Due to high communication overhead and topology complexity, it is impractical for SU nodes to cooperate with competing SUs and share their private information in multi-hop CRN. To resolve this contradiction, a conjecture-based multi-agent learning scheme with quasi-cooperative scenario is proposed, where each SU node conjectures other SUs' behavior strategies without full coordination among agents.

Specifically, from Equations (20) and (21), the mixed-strategies for other competing SUs is defined as $\varphi_i^t(s_i, \mathbf{a_{-i}}) = \prod_{j \in M \backslash \{i\}} \zeta_j^t(s_j, a_j)$, which represents the joint probability that competing SUs perform strategy vector $\boldsymbol{\zeta}_{-\mathbf{i}} = \left\{\zeta_j^t(s_j, a_j)\right\}_{j \in M \backslash \{i\}}$ at time slot t. In other words, estimating $\varphi_i^t(s_i, \mathbf{a_{-i}})$ becomes the key challenge when applying the multi-agent Q-learning framework. To combat this, the conjecture belief $\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}})$ is introduced to approximate $\varphi_i^t(s_i, \mathbf{a_{-i}})$, and the ERT-CMQL is proposed to asymptotically determine $\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}})$ without complete network information. The probability that the agent chooses a_i in state s_i while other competing SUs execute action vector $\mathbf{a_{-i}}$ is given by:

$$\delta_i = \zeta_i^t(s_i, a_i) \cdot \varphi_i^t(s_i, \mathbf{a_{-i}}). \tag{22}$$

SU_i receives expected reward $R_i(s_i, a_i, \mathbf{a_{-i}})$ when the agent of SU_i performs action a_i, while other nodes select action vector $\mathbf{a_{-i}}$ in state s_i. That is, the probability that SU_i acquires $R_i(s_i, a_i, \mathbf{a_{-i}})$ is δ_i. n is the number of time steps between any two moments in which SU_i achieves the same return $R_i(s_i, a_i, \mathbf{a_{-i}})$. Each n is independent of the others and follows the same distribution of δ_i. The average value of n is denoted as $\overline{n}$, which can be obtained via the private information from historical observation. Then we have the approximate equation $\delta_i \approx 1/(1+\overline{n})$ [15], i.e., $\zeta_i^t(s_i, a_i) \cdot \varphi_i^t(s_i, \mathbf{a_{-i}}) \approx 1/(1+\overline{n})$. Since every SU knows its own transmission strategy $\zeta_i^t(s_i, a_i)$, the agent can estimate $\varphi_i^t(s_i, \mathbf{a_{-i}})$ via:

$$\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}}) = \frac{1}{(1+\overline{n}) \cdot \zeta_i^t(s_i, a_i)}. \tag{23}$$

After obtaining the expression of $\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}})$ using local information shown in Equation (23), the updating rule of the conjecture belief is explored. In quasi-cooperative learning scenarios, agents update their conjecture belief based on new observations. Since n is a stationary stochastic process in the time dimension, its mean value $\overline{n}$ is a constant. Specifically, the quotient of the conjecture belief at time slot $t-1$ and t can be calculated as:

$$\begin{aligned}\frac{\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}})}{\widetilde{\varphi}_i^{t-1}(s_i, \mathbf{a_{-i}})} &= \frac{1}{(1+\overline{n}) \cdot \zeta_i^t(s_i, a_i)} \Bigg/ \left[\frac{1}{(1+\overline{n}) \cdot \zeta_i^{t-1}(s_i, a_i)}\right] \\ &= \frac{\zeta_i^{t-1}(s_i, a_i)}{\zeta_i^t(s_i, a_i)}\end{aligned} \tag{24}$$

Then, the conjecture belief is updated as follows:

$$\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}}) = \widetilde{\varphi}_i^{t-1}(s_i, \mathbf{a_{-i}}) \cdot \frac{\zeta_i^{t-1}(s_i, a_i)}{\zeta_i^t(s_i, a_i)}. \tag{25}$$

Since $\varphi_i^t(s_i, \mathbf{a_{-i}}) = \prod_{j \in M \backslash \{i\}} \zeta_j^t(s_j, a_j) \approx \widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}})$, the updating rule in Equation (20) can be rewritten as:

$$Q_i^{t+1}(s_i, a_i) = (1-\alpha) Q_i^t(s_i, a_i) + \alpha \left[\sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \widetilde{\varphi}_i^t(s_i, a_{-i}) + \beta \max_{b_i \in \mathbf{A_i}} Q_i^t(s'_i, b_i)\right]. \tag{26}$$

Equation (26) shows that every SU node only uses private strategy and limited information exchange with its previous nodes to update its Q-value. SU_i conjectures the mixed-strategies for other competing SUs on the basis of their variations in response to their own strategy.

However, strong correlations exist between $\zeta_i^t(s_i, a_i)$ and $\zeta_i^{t-1}(s_i, a_i)$, which may cause the parameters to easily stick in a poor local optimum and then make $\zeta_i^{t-1}(s_i, a_i)/\zeta_i^t(s_i, a_i)$ close to 1 infinitely. Since the updating rule of $\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}})$ is fractional which has a strong reliance on $\zeta_i^{t-1}(s_i, a_i)/\zeta_i^t(s_i, a_i)$, the conjecture belief is also inclined to fall into the local optimal solution. To avoid the shortage and further improve the system performance, experience replay is applied to the conjecture based multi-agent learning scheme. From long-term-observations, $\overline{n}$ is a constant value due to the time stationarity of n. So the probability that SU_i receives an expected reward $R_i(s_i, a_i, \mathbf{a_{-i}})$ (i.e., agents perform action vector $(a_i, \mathbf{a_{-i}})$ in state s_i) is approximately equal to the reciprocal of mean time interval regardless of time step, that is:

$$\zeta_i^t(s_i, a_i) \cdot \varphi_i^t(s_i, \mathbf{a_{-i}}) \approx \frac{1}{1+\overline{n}} \approx \zeta_i^v(s_i, a_i) \cdot \varphi_i^v(s_i, \mathbf{a_{-i}}), \tag{27}$$

where t and v represent any two time slots. Thus we have $\zeta_i^t(s_i, a_i) \cdot \widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}}) = \zeta_i^v(s_i, a_i) \cdot \widetilde{\varphi}_i^v(s_i, \mathbf{a_{-i}})$. $\zeta_i(s_i, a_i)$ and $\widetilde{\varphi}_i(s_i, \mathbf{a_{-i}})$ at each time step are stored as the agent's experience at each time slot, and pooled over many episodes into a replay memory [27]. During learning, we randomly sample the experience $\widetilde{\varphi}_i^k(s_i, \mathbf{a_{-i}})$ and $\zeta_i^k(s_i, a_i)$ from memory pool to update the conjecture. Then the update of conjecture belief at time step t is given by:

$$\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}}) = \widetilde{\varphi}_i^v(s_i, \mathbf{a_{-i}}) \cdot \frac{\zeta_i^v(s_i, a_i)}{\zeta_i^t(s_i, a_i)}. \tag{28}$$

This approach has several advantages over consecutive updating rule in Equation (25). First, each time-step of the strategy is potentially used in the update of conjecture, which improves data efficiency instead of updating directly from consecutive samples. Second, strong correlations between $\zeta_i^t(s_i, a_i)$ and $\zeta_i^{t-1}(s_i, a_i)$ may result in a local optimal. Randomizing the samples breaks these correlations and reduces the variance in the updates.

The details of ERT-CMQL are obtained as described in Algorithm 1.

Algorithm 1 Equal Reward Time-Slots Based Conjectural Multi-Agent Q-Learning

```
Initialize:
Set $t = 0$ and memory size $N$.
  For each $SU_i$ Do
    For each $s_i \in S_i,\ a_i \in A_i$ Do
        Initialize transmission strategy $\zeta_i^t(s_i, a_i)$, conjecture belief $\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}})$,
        Q-value $Q_i^t(s_i, a_i)$, and replay memory $D = \{\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}}),\ \zeta_i^t(s_i, a_i)\}$.
     End For
   End For
Repeated Learning:
   For each $SU_i$ Do
      For $eposide = 1,\ M$ do
          Initialize state $s_i^1$.
          Loop
          Select action $a_i^t$ according to the strategy $\zeta_i^t(s_i, a_i)$.
          Execute action $a_i^t$, and obtain strategies of previous nodes $\zeta_k(s_k,\ a_k)$
          and the SINR indicator $\rho_i$.
          Observe reward $R_i^t(s_i, a_i, \mathbf{a_{-i}})$ and state $s_i^{t+1}$ according to (14) and (16).
          Update $Q_i^{t+1}(s_i, a_i)$ based on $\widetilde{\varphi}_i^t(s_i, \mathbf{a_{-i}})$ according to (26).
          Update the strategy $\left\{\zeta_i^{t+1}(s_i, a_i)\right\}_{a_i \in A_i}$ according to (19).
          Sample experience $\widetilde{\varphi}_i^v(s_i, \mathbf{a_{-i}})$ and $\zeta_i^v(s_i, a_i)$ from $D$.
          Update the conjecture belief $\widetilde{\varphi}_i^{t+1}(s_i, \mathbf{a_{-i}})$ according to (28).
          Store $\widetilde{\varphi}_i^{t+1}(s_i, \mathbf{a_{-i}})$ and $\zeta_i^{t+1}(s_i, a_i)$ in $D$.
          $s_i = s_i^{t+1}$
          $t = t + 1$
        Until $s_i$ is the terminal state
      End For
    End For
```

4.3. Analysis of ERT-CMQL

Littleman [28] provided the convergence proof of the standard Q-learning. Based on the theory, the convergence of ERT-CMQL is investigated in this section.

Lemma. Suppose there is a mapping $P : \mathbf{Q} \to \mathbf{Q}$, and $\mathbf{Q}$ denotes the set of all SUs' Q functions. The updating rule $Q^{t+1} = (1 - \alpha)Q^t + \alpha \cdot P(Q^t)$ converges to Q^* with probability of 1, if:

(1) $Q^* = E[P(Q^*)]$

(2) A number $0 < \sigma < 1$ exists such that $\|P(Q^t) - P(Q^*)\| \le \sigma \|Q^t - Q^*\|$ for all $Q^t \in Q$.

To apply the lemma in the convergence proof of our proposed ERT-CMQL, the definition is given as follows:

Definition. *Let $Q^t = \left(Q_1^t, \ldots, Q_M^t\right)$, where $Q_i^t \in \mathbf{Q_i}$ for $i \in M$, and $\mathbf{Q} = \prod_{i \in M} Q_i$. Then the mapping $P : \mathbf{Q} \rightarrow \mathbf{Q}$ is defined as $P\left(Q^t\right) = \left[P\left(Q_1^t\right), \ldots, P\left(Q_M^t\right)\right]$, where:*

$$P\left(Q_i^t(s_i, a_i)\right) = \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \widetilde{\varphi}_i^t(s_i, a_{-i}) + \beta \max_{b_i \in \mathbf{A_i}} Q_i^t({s'}_i, b_i). \tag{29}$$

In addition, for any $Q,\ Q' \in \mathbf{Q}$, the definition of the distance between Q-values is given as:

$$\|Q - Q'\| = \max_{i \in M} \max_{s_i \in \mathbf{S_i}} \max_{a_i \in \mathbf{A_i}} \left|Q_i(s_i, a_i) - {Q_i}'(s_i, a_i)\right|. \tag{30}$$

Firstly, we prove the first condition in Lemma 1 for ERT-CMQL.

Proposition 1. *Q^* is equal to the expectation of its map $P(Q^*)$, i.e., $Q^* = E[P(Q^*)]$, where $Q^* = \left(Q_1^*, \ldots, Q_M^*\right)$.*

Proof. According to the Bellman's optimality equation [29], we have the following expression:

$$Q_i^*(s_i, a_i) = E\left[R_i(s_i, a_i, \zeta^*_{-\mathbf{i}})\right] + \beta \sum_{{s'}_i \in S_i} P_{s_i, {s'}_i}(a_i, \zeta^*_{-i}) \max_{b_i \in A_i} Q_i^*({s'}_i, b_i), \tag{31}$$

Since the reward $R_i(s_i, a_i, \zeta^*_{-\mathbf{i}})$ is irrelevant to ${s'}_i$, then Equation (31) can be modified as:

$$\begin{aligned} Q_i^*(s_i, a_i) &= \sum_{{s'}_i \in S_i} P_{s_i, {s'}_i}(a_i, \zeta^*_{-i}) \left\{ E\left[R_i(s_i, a_i, \zeta^*_{-i})\right] + \beta \max_{b_i \in A_i} Q_i^*({s'}_i, b_i) \right\} \\ &= \sum_{{s'}_i \in S_i} P_{s_i, {s'}_i}(a_i, \zeta^*_{-i}) \left\{ \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \prod_{j \in M \backslash \{i\}} \zeta_j(s_j, a_j) + \beta \max_{b_i \in A_i} Q_i^*({s'}_i, b_i) \right\} \end{aligned} \tag{32}$$

Based on the previous analysis in Section 4.2, we have $\varphi_i^t(s_i, \mathbf{a_{-i}}) = \prod_{j \in M \backslash \{i\}} \zeta_j^t(s_j, a_j)$. Then we prove that $Q_i^*(s_i, a_i) = E\left[P\left(Q_i^*(s_i, a_i)\right)\right]$. □

Proposition 2. *There is a number $0 < \sigma < 1$ such that $\|P\left(Q^t\right) - P(Q^*)\| \leq \sigma \|Q^t - Q^*\|$.*

Proof. In accordance with the definition of the distance between Q-values, we have:

$$\begin{aligned} \|P(Q) - P(Q')\| &= \max_{i \in M} \max_{s_i \in \mathbf{S_i}} \max_{a_i \in \mathbf{A_i}} \left| P(Q_i(s_i, a_i)) - P\left({Q_i}'(s_i, a_i)\right) \right| \\ &= \max_{i \in M} \max_{s_i \in \mathbf{S_i}} \max_{a_i \in \mathbf{A_i}} \left| \begin{array}{c} \sum\limits_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \left[\widetilde{\varphi}_i(s_i, a_{-i}) - \widetilde{\varphi}_i'(s_i, a_{-i})\right] \\ + \beta \left[\max\limits_{b_i \in \mathbf{A_i}} Q_i({s'}_i, b_i) - \max\limits_{b_i \in \mathbf{A_i}} {Q'}_i({s'}_i, b_i) \right] \end{array} \right| \\ &\leq \max_{i \in M} \max_{s_i \in \mathbf{S_i}} \max_{a_i \in \mathbf{A_i}} \left| \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \left[\widetilde{\varphi}_i(s_i, a_{-i}) - \widetilde{\varphi}_i'(s_i, a_{-i})\right] \right| \\ &\quad + \max_{i \in M} \max_{s_i \in \mathbf{S_i}} \max_{a_i \in \mathbf{A_i}} \beta \left| \max_{b_i \in \mathbf{A_i}} Q_i({s'}_i, b_i) - \max_{b_i \in \mathbf{A_i}} {Q'}_i({s'}_i, b_i) \right| \\ &\leq \max_{i \in M} \max_{s_i \in \mathbf{S_i}} \max_{a_i \in \mathbf{A_i}} \left| \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \left[\widetilde{\varphi}_i(s_i, a_{-i}) - \widetilde{\varphi}_i'(s_i, a_{-i})\right] \right| \\ &\quad + \max_{i \in M} \max_{s_i \in \mathbf{S_i}} \max_{a_i \in \mathbf{A_i}} \left[\max_{b_i \in \mathbf{A_i}} \beta \left| Q_i({s'}_i, b_i) - {Q'}_i({s'}_i, b_i) \right| \right] \\ &= \max_{i \in M} \max_{s_i \in \mathbf{S_i}} \max_{a_i \in \mathbf{A_i}} \left| \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \left[\widetilde{\varphi}_i(s_i, a_{-i}) - \widetilde{\varphi}_i'(s_i, a_{-i})\right] \right| + \beta \|Q - Q'\| \end{aligned} \tag{33}$$

where the second equation is derived from Equation (29), and the first inequality is obtained according to the transformation $|A + B| \leq |A| + |B|$. It can be easily proven that $|\max A - \max B| \leq \max |A - B|$, so we can attain the second inequality. Since a_i is unrelated to $Q_i(s'_i, b_i) - Q'_i(s'_i, b_i)$, $\max\limits_{i \in M} \max\limits_{s_i \in \mathbf{S_i}} \max\limits_{a_i \in \mathbf{A_i}} \left[\max\limits_{b_i \in \mathbf{A_i}} \beta |Q_i(s'_i, b_i) - Q'_i(s'_i, b_i)| \right]$ can be rewritten as $\max\limits_{i \in M} \max\limits_{s_i \in \mathbf{S_i}} \max\limits_{b_i \in \mathbf{A_i}} \beta |Q_i(s'_i, b_i) - Q'_i(s'_i, b_i)|$ and the last equation is attained using Equation (30).

Next, we apply Equation (23) to the item $\sum\limits_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \left[\widetilde{\varphi}_i(s_i, a_{-i}) - \widetilde{\varphi}'_i(s_i, a_{-i}) \right]$, and the expression can be rewritten as:

$$\begin{aligned}
& \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \left[\widetilde{\varphi}_i(s_i, a_{-i}) - \widetilde{\varphi}'_i(s_i, a_{-i}) \right] \\
= & \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \cdot \left[\tfrac{1}{(1+\overline{n})\zeta_i(s_i, a_i)} - \tfrac{1}{(1+\overline{n})\zeta'_i(s_i, a_i)} \right] \\
= & \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \cdot \left[\tfrac{\zeta'_i(s_i, a_i)}{(1+\overline{n})\zeta_i(s_i, a_i)\zeta'_i(s_i, a_i)} - \tfrac{\zeta_i(s_i, a_i)}{(1+\overline{n})\zeta_i(s_i, a_i)\zeta'_i(s_i, a_i)} \right] \\
= & \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \cdot \tfrac{\zeta'_i(s_i, a_i) - \zeta_i(s_i, a_i)}{(1+\overline{n})\zeta_i(s_i, a_i)\zeta'_i(s_i, a_i)}
\end{aligned} \tag{34}$$

where $R_i(s_i, a_i, \mathbf{a_{-i}})$ is the reward when SU_i selects action a_i in state s_i, while other nodes select action vector $\mathbf{a_{-i}}$, $\overline{n}$ is the average number of time steps between any two moments in which SU_i achieves the same return $R_i(s_i, a_i, \mathbf{a_{-i}})$, and $\zeta_i(s_i, a_i)$ and $\zeta'_i(s_i, a_i)$ are the strategies of SU_i in state s_i at different time-step.

When η is sufficiently large, we have:

$$e^{Q_i(s_i, a_i)/\eta} = 1 + \frac{Q_i(s_i, a_i)}{\eta} + \left(\left(\frac{Q_i(s_i, a_i)}{\eta} \right)^2 \right) = 1 + \frac{Q_i(s_i, a_i)}{\eta} + \omega\left(\frac{Q_i(s_i, a_i)}{\eta} \right), \tag{35}$$

and:

$$\sum_{b \in A_i} e^{Q_i(s_i, b)/\eta} = |\mathbf{A_i}| + \sum_{b \in A_i} \left[\frac{Q_i(s_i, b)}{\eta} + \omega\left(\frac{Q_i(s_i, b)}{\eta} \right) \right], \tag{36}$$

where $\omega\left(\frac{Q_i(s_i, a_i)}{\eta} \right)$ is a polynomial of order $\left(\left(\frac{Q_i(s_i, a_i)}{\eta} \right)^2 \right)$. By applying Equations (35) and (36) to Equation (19), it can be verified that:

$$\zeta_i(s_i, a_i) = \frac{1}{|\mathbf{A_i}|} + \frac{1}{|\mathbf{A_i}|} \cdot \frac{Q_i(s_i, a_i)}{\eta} + \varpi\left(\left\{ \frac{Q_i(s_i, b)}{\eta} \right\}_b \right), \tag{37}$$

and:

$$\zeta'_i(s_i, a_i) = \frac{1}{|\mathbf{A_i}|} + \frac{1}{|\mathbf{A_i}|} \cdot \frac{Q'_i(s_i, a_i)}{\eta} + \varpi\left(\left\{ \frac{Q'_i(s_i, b)}{\eta} \right\}_b \right), \tag{38}$$

where $\varpi\left(\left\{ \frac{Q_i(s_i, b)}{\eta} \right\}_b \right)$ is a polynomial of order smaller than $\left(\left\{ \frac{Q_i(s_i, a_i)}{\eta} \right\}_b \right)$.

Substituting Equations (37) and (38) into Equation (34), we have:

$$\begin{aligned}
& \sum_{(a_i, a_{-i}) \in A} R_i(s_i, a_i, a_{-i}) \cdot \tfrac{\zeta'_i(s_i, a_i) - \zeta_i(s_i, a_i)}{(1+\overline{n})\zeta_i(s_i, a_i)\zeta'_i(s_i, a_i)} \\
= - & \sum_{(a_i, a_{-i}) \in A} \tfrac{R_i(s_i, a_i, a_{-i})}{(1+\overline{n})\zeta_i(s_i, a_i)\zeta'_i(s_i, a_i)} \left(\begin{array}{c} \frac{1}{\eta |A_i|} \cdot [Q_i(s_i, a_i) - Q'_i(s_i, a_i)] \\ + \varpi\left(\left\{ \frac{Q_i(s_i, b)}{\eta} \right\}_b \right) - \varpi\left(\left\{ \frac{Q'_i(s_i, b)}{\eta} \right\}_b \right) \end{array} \right) \\
= - & \sum_{(a_i, a_{-i}) \in A} \tfrac{R_i(s_i, a_i, a_{-i})}{\eta(1+\overline{n})\zeta_i(s_i, a_i)\zeta'_i(s_i, a_i)} \cdot \tfrac{1}{|A_i|} \cdot [Q_i(s_i, a_i) - Q'_i(s_i, a_i)] + \varpi\left(\left\{ \tfrac{Q_i(s_i, b)}{\eta} \right\}_b \right) - \varpi\left(\left\{ \tfrac{Q'_i(s_i, b)}{\eta} \right\}_b \right)
\end{aligned} \tag{39}$$

A sufficiently large η can be taken so that:

$$\left| \frac{R_i(s_i, a_i, \mathbf{a_{-i}})}{\eta(1+\overline{n})\zeta_i(s_i, a_i)\zeta'_i(s_i, a_i)} \right| \leq 1 - \beta. \tag{40}$$

Then we have the following inequality:

$$\left|\sum_{(a_i,a_{-i})\in A} R_i(s_i,a_i,a_{-i})\left[\widetilde{\varphi}_i(s_i,a_{-i}) - \widetilde{\varphi}'_i(s_i,a_{-i})\right]\right| \le \frac{1-\beta}{|\mathbf{A_i}|}\cdot\left|Q_i(s_i,a_i) - Q'_i(s_i,a_i)\right|, \tag{41}$$

which leads to:

$$\begin{aligned} \|P(Q) - P(Q')\| \le \quad & \max_{i\in M}\max_{s_i\in\mathbf{S_i}}\max_{a_i\in\mathbf{A_i}}\tfrac{1-\beta}{|\mathbf{A_i}|}\cdot|Q_i(s_i,a_i) - Q'_i(s_i,a_i)| + \beta\|Q-Q'\| \\ & \le \tfrac{1-\beta}{v}\|Q-Q'\| + \beta\|Q-Q'\| = \tfrac{1-\beta+\beta v}{v}\|Q-Q'\| \end{aligned} \tag{42}$$

where $v = \min_{i\in N}|\mathbf{A_i}| > 1$. Then we have $v\cdot(1-\beta) > 1-\beta$, so that $v - \beta v > 1-\beta$, which leads to $\frac{1-\beta+\beta v}{v} < 1$. Consequently, condition (2) is satisfied in the Lemma, and ERT-CMQL is proven to converge if η is large enough for all agents.

5. Simulation Results

In this section, the performance of our quasi-cooperative multi-agent learning scheme is evaluated using an event-driven simulator coded in Python 3.5. The network model and learning framework are built based on the Python packages Networkx and Numpy, respectively. The results of the proposed ERT-CMAQL are compared with (1) Cooperative Multi-Agent Q-Learning (CMAQL) which is the ideal scheme and has complete information of the competing SUs; (2) Conjectural Multi-Agent Q-Learning without Experience Replay (CMAQL-ER); (3) Fixed Power- based Conjectural Multi-Agent Q-Learning (FP-CMAQL) proposed in [16] which transmits data with a fixed power level; (4) a single agent Q-learning scheme called Q-routing presented in [14] and (5) Prioritized Memories Deep Q-Network (PM-DQN) based joint design scheme proposed in our previous work [17].

In the multi-agent learning framework, we initialize the conjecture belief $\widetilde{\varphi}_i^0(s_i,\mathbf{a_{-i}}) = 1$, the Q-value $Q_i^0(s_i,a_i) = 0$, and the transmission strategy $\zeta_i^0(s_i,a_i) = 1/|\mathbf{A_i}|$ for each $s_i \in S_i,\ a_i \in A_i$. Other system parameters are given in Table 2.

Table 2. System Parameters.

Parameters	Values
Link Gain	$h = \varepsilon G(r/r_0)^{-m}$, for $r > r_0$ [15]
Available Spectrum	56 MHz − 62 MHz
Bandwidth, B	1 MHz
AWGN power, ϑ	10^{-7} mW
PU-to-SU interference, ϕ_{ijc}^{PU}	$\phi_{ijc}^{PU} \sim \left[10^{-7},\ 10^{-6}\right]$ mW
Packet Size, R_{packet}	2×10^5 bit
Mean of PU Departure Rate, μ	0.1
Deviation of PU Departure Rate, σ	0.05
SINR threshold, γ_{th}	60 dBm
Outage probability, P_{ij}^{out}	$P_{ij}^{out} \sim N(0.1,\ 0.05)$
Data flow arrival rate, λ_f	0.4
Data flow service rate, μ_f	0.6
Maximal transmission delay, κ_{th}	200 ms
Discount factor, β	0.9
Learning rate, α	0.1
Time step size, t	500 ms
Temperature, η	0.005

To verify the performance of algorithms, a small CRN containing 10 SUs and 4 PUs is simulated at first. SUs in CRN are uniformly deployed in a 300 × 300 m region. In addition, the available transmitting power consists of five levels: $\{50, 100, \ldots, 250 \text{ mW}\}$. The network topology is shown in Figure 2.

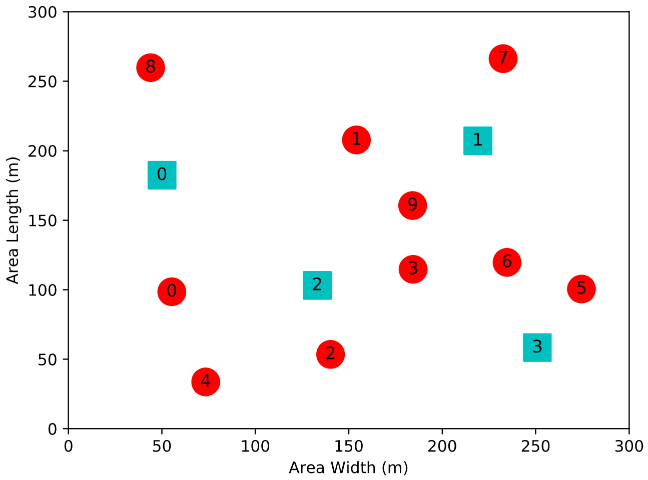


Figure 2. Network topology consisting of 10 SUs and 4 PUs.

Without loss of generality, SU 6 is taken as an example. The single-node performance of SU 6 for different algorithms is shown in Figures 3 and 4. Figure 3 illustrates the average reward of SU 6 versus the iteration index. The expected reward firstly rises and then stays almost steady for all schemes. Furthermore, we find that, when converged, CMAQL outperforms all other algorithms. The reward of ERT-CMAQL is slightly lower than CMAQL, followed by the CMAQL-ER scheme, and FP-CMAQL obtains the lowest reward. This occurs mainly because, in the CMAQL scheme, agents have true strategies of competing SUs through global information exchange. In ERT-CMAQL, each agent approximates mixed-strategies of other SUs via the conjecture belief that may be not sufficiently accurate. We can see that the reward of CMAQL-ER is slightly higher than that of ERT-CMAQL before 200 iterations, and afterward, that ERT-CMAQL is superior to CMAQL-ER. The reason for this is that at first few samples are stored in replay memory and the correlation is weak between the samples, so that experience replay is inefficient compared to the consecutive updating rule. At the later stage the advantage of experience replay is fully demonstrated when samples are abundant. In addition, FP-CMAQL obtains the lowest average reward, which illustrates the importance of power allocation.

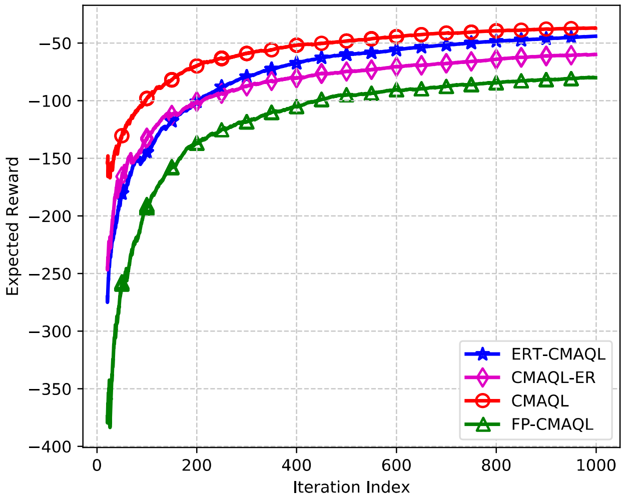
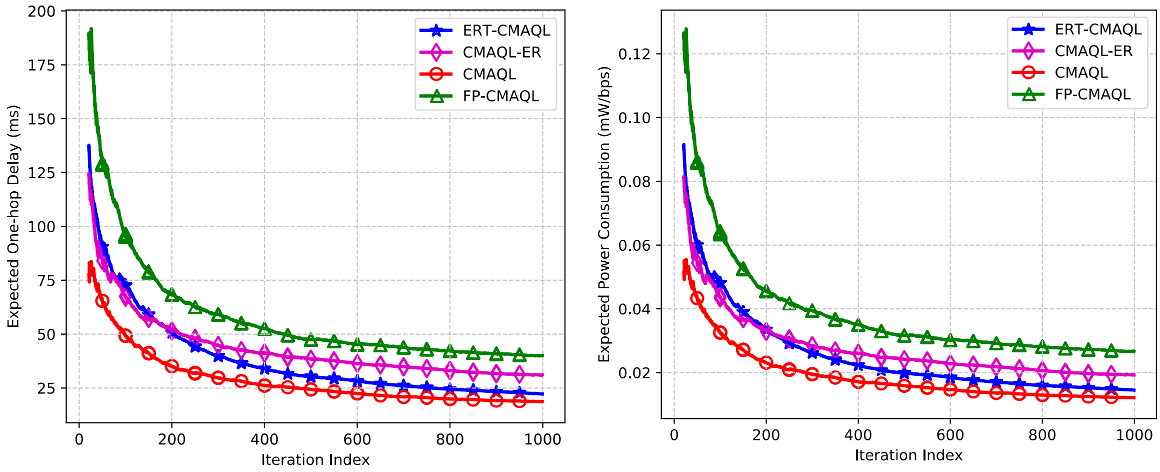


Figure 3. Expected reward of the SU 6 versus the iteration index.

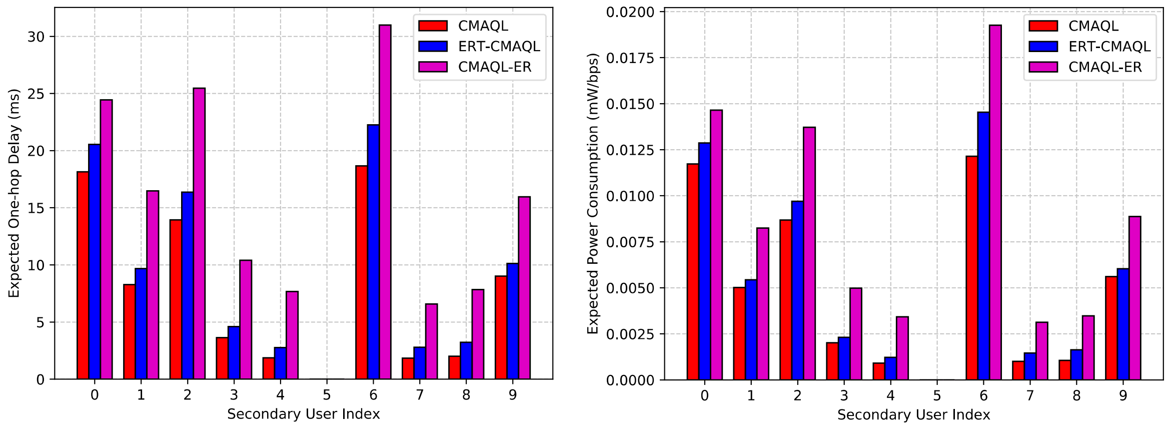


(**a**) (**b**)

Figure 4. (**a**) Expected end-to-end delay and (**b**) expected power consumption ratio.

The effectiveness properties of transmission latency and power efficiency are demonstrated in this part. In Figure 4a, single-hop latency declines in the beginning and flattens after about 700 iterations for all kinds of protocols. CMAQL achieves the lowest transmission latency, which is a little shorter than that of ERT-CMAQL. The expected delay of ERT-CMAQL is about 32% lower than CMAQL-ER, which benefits from experience replay to avoid a poor local minimum and enhace data efficiency. The transmission delay of FP-CMAQL is much longer than the other three schemes because it fails to adjust the transmission power with channel status causing larger overall latency. Figure 4b shows the average power consumption versus iteration index. The PCR of the four algorithms, in increasing order, is as follows: CMAQL, ERT-CMAQL, CMAQL-ER and FP-CMAQL. The reason for this is the same as in Figure 4a. Consequently, the results illustrate that the energy efficiency of proposed ERT-CMAQL is close to the optimum scheme, and holds a clear advantage over other algorithms.

Figure 5a shows the expected one-hop delay of different SU nodes for CMAQL, ERT-CMAQL, and CMAQL-ER schemes. The transmission delay of CMAQL is the lowest compared with the other two algorithms, the latency of ERT-CMAQL is slightly higher than that of CMAQL, and CMAQL-ER achieves the longest one-hop delay for all SU nodes in the network. This illustrates the effect of experience replay, which makes the performance of ERT-CMAQL close to the optimum, demonstrating a clear advantage over the schemes applying the consecutive updating rule.



(**a**) (**b**)

Figure 5. (**a**) Expected single-hop delay and (**b**) expected power consumption.

In addition, we find that SU 6 achieves the longest expected one-hop delay of all SUs, while the transmission latency of SU 4 and SU 7 is relatively low on average. This is due to SU 6 being the closest SU to the destination node so that data flows pass through it with higher probability. The locations of SU 4 and SU 7 are relatively isolated, so the arrival packets are scarce. SU 5 is the destination node and no packet is transmitted forward so that its transmission delay is 0. Comparison of PCR for the three kinds of protocols varying in the SU index is shown in Figure 5b. We can find that the PCR of ERT-CMAQL is close to CMAQL for all SUs. CMAQL-ER achieves the highest PCR of the three schemes for all SU nodes, which consumes 46% more power per throughput on average than ERT-CMAQL. The reason for this is similar to the reason for the results in Figure 5a. Therefore, the proposed ERT-CMAQL achieves relatively low transmission latency and power consumption close to the optimum for every SU node in the network. Figure 6 illustrates the effect of PU arrival probability on expected end-to-end delay and system PCR. As shown in Figure 6a, it is found that the transmission latency of the four algorithms grows as the PU arrival probability increases. This is because the larger PU arrival probability results in more interruption and transmission failure, which causes longer delays for data retransmission. CMAQL achieves the lowest transmission delay, followed by ERT-CMAQL. The latency of CMAQL-ER is higher than that of ERT-CMAQL, and FP-CMAQL has the longest expected end-to-end latency. This demonstrates the advantage of our proposed ERT-CMAQL, which produces performance closest to the ideal value. Furthermore, the transmission latency values of the four algorithms are relatively close when PU arrival probability is low. However, they differ considerably as PU arrival probability increases. When the probability of PU arrival is low, there is little conflict between PUs and SUs so the four algorithms achieve almost the same latency. Since CMAQL and ERT-CMAQL can better avoid conflicts with PU, CMAQL and ERT-CMAQL are capable of maintaining relatively low latency when PU arrival probability increases. However, in CMAQL-ER and FP-CMAQL, the data transmission of SUs is often interrupted by PU arrival so the transmission latency is high. The effect of PU arrival probability on PCR shown in Figure 6b has a similar trend to Figure 6a, which will not be detailed here.

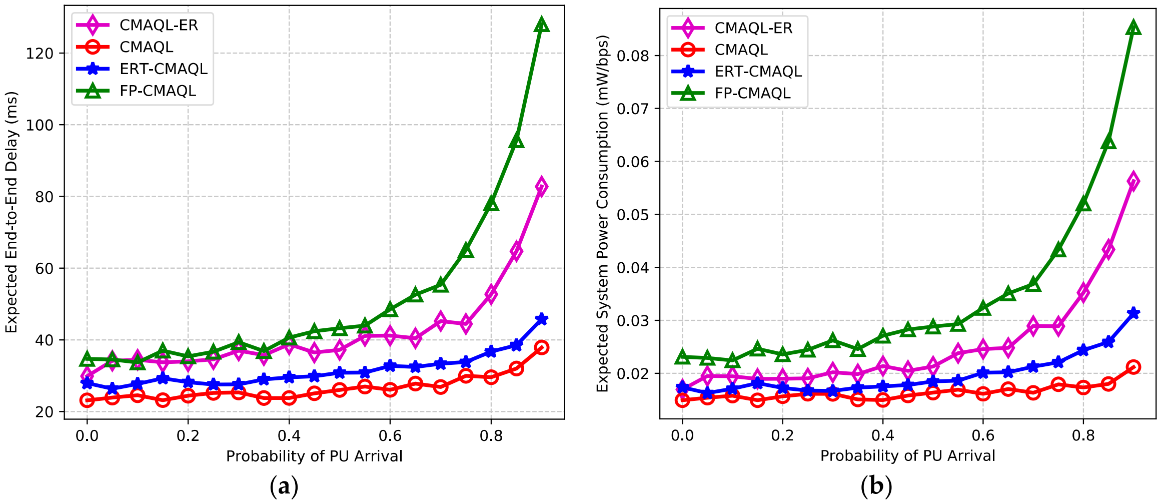


Figure 6. (**a**) Expected end-to-end delay and (**b**) expected system power consumption.

Next, for a more general case, a networking scenario comprising 20 SUs and 10 PUs uniformly deployed in a 500 × 500 m area is considered. The available transmit power contains ten levels: {50, 100, . . . , 500 mW}. The network topology of the second experiental scenario is shown in Figure 7.

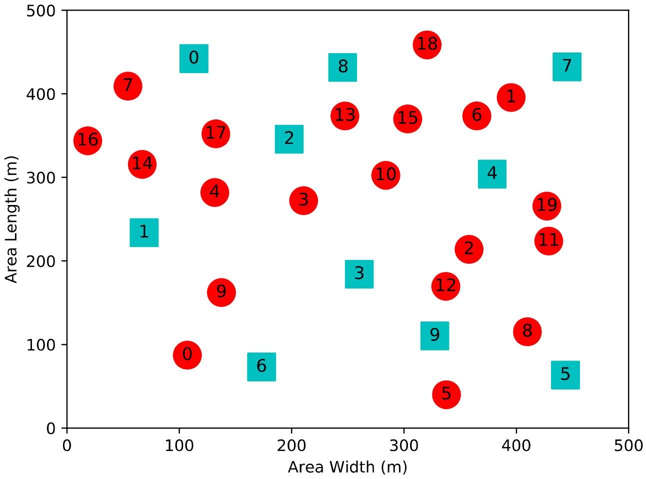


Figure 7. Network topology consisting of 20 SUs and 10 PUs.

The comparison of system performance for different kinds of experimental environments is illustrated in this section. Figure 8 depicts the expected end-to-end latency of six algorithms in networks with 10 SUs and 20 SUs. It can be seen that with increasing number of routes, the end-to-end delay sharply declines and then remains steady for all algorithms in both networking scenarios. When converged, CMAQL achieves the lowest end-to-end delay due to the complete information, which helps agents make more accurate and comprehensive decisions. Given the conjecture belief and experience replay, the total transmission delay of ERT-CMAQL is close to CMAQL. CMAQL-ER, with its consecutive updating rule, consumes more time transmiting packets from the source to the destination than ERT-CMAQL, followed by FP-CMAQL. The transmission latency of the two single-agent schemes is particularly larger because, in these two schemes, all the information and computations are processed by a separate agent, which is inherently less efficient than multi-agent schemes. PM-DQN produces a longer end-to-end delay than Q-routing in the network with 10 SUs, but its performance is superior to Q-routing in a large network. This illustrates the advantage of PM-DQN in the networking scenarios with large state space.

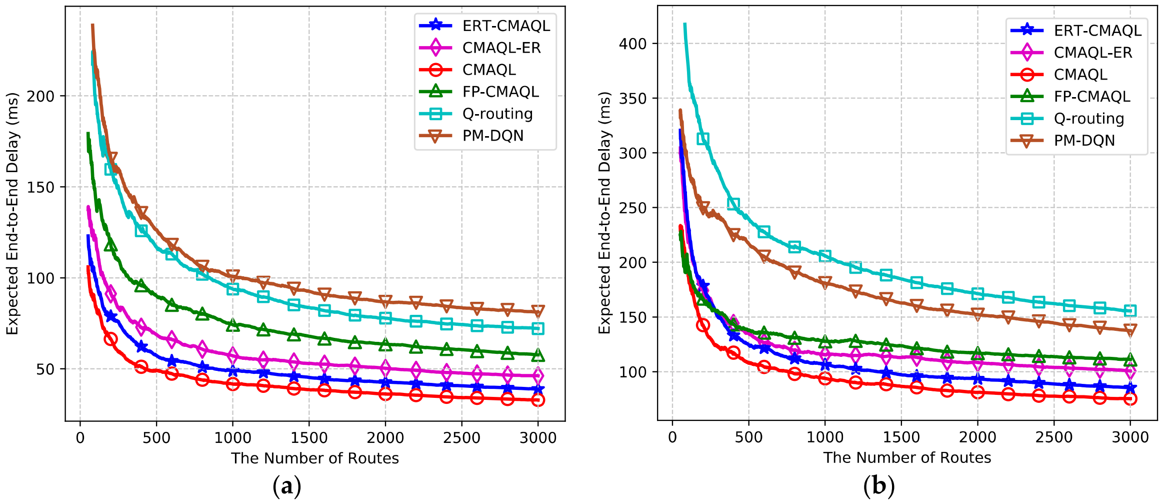


Figure 8. Expected end-to-end delay versus the number of routes: (**a**) 10 SU nodes, 4 PU channels and (**b**) 20 SU nodes, 10 PU channels.

By comparing the performance of the two networking scenarios, we can find that the end-to-end latency of all protocols in the network with 20 SUs is relatively longer than in the small-scale network, and the latency of single-agent schemes increases more apparently than in other algorithms. The reason for this is that the links of the route increase with increasing SUs in the network, so that the accumulated single-hop latency along the route, i.e., the end-to-end delay, grows as well. Single-agent schemes are more sensitive to the number of SUs, which leads to longer latency in total. Furthermore, it is observed that the convergence speed of multi-agent schemes remains almost the same in both networking scenarios. However, Q-routing and PM-DQN converge at around 1700 routes in the first experiment, and nearly 3000 routes in the second. From the theoretical analysis, we find that multi-agent learning schemes are not affected by network scale because each SU equips an agent and follows the same learning rule. The calculation load rises as the number of SUs grows, which has heavy impact on single-agent schemes with only one agent in the network.

We further investigate the packet loss ratio (PLR) of the six protocols for different networking scenarios in Figure 9. In both experimental environments, CMAQL has the lowest PLR, followed by the proposed ERT-CMAQL. The PLR of Q-routing and PM-DQN is obviously higher than other multi-agent learning schemes, which illustrates the reliability of using multiple agents. Comparing Figure 9a,b, we find that the PLR of Q-routing is larger than PM-DQN in the first network, whereas PM-DQN is more robust than Q-routing in large-scale networks. This is because PM-DQN has higher efficiency in networks with large state space due to the capability of the neural network. In addition, the PLR of all multi-agent schemes remains almost the same, which demonstrates that the routing reliability is not affected by network scale for multi-agent learning algorithms. The reason for this finding is that, in multi-agent collaboration schemes, every SU node equips an agent regardless of the network size, which improves the robustness as the number of SUs increases.

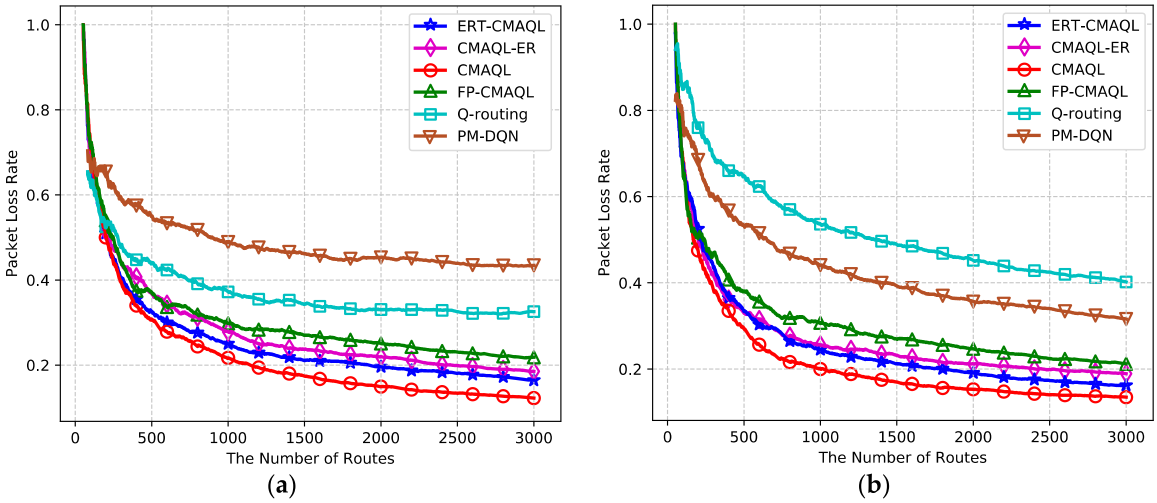


Figure 9. Packet loss ratio versus the number of routes: (**a**) 10 SU nodes, 4 PU channels and (**b**) 20 SU nodes, 10 PU channels.

6. Conclusions

In this paper, we developed a quasi-cooperative multi-agent learning scheme for multi-hop CRN called ERT-CMAQL. The simulation results show that ERT-CMAQL reduces the expected end-to-end latency, guarantees the robustness of routing and achieves higher power efficiency compared to traditional learning algorithms, and its performance is close to CMAQL using complete information. In this paper, every SU agent learns the information of topology and channel statistics by itself. However, self-learning faces two crucial challenges: it requires a large number of interactions between agents and environment, which takes considerable time, and some energy-constraint applications cannot afford to the large power expenditure due to the trial and error manner of RL. Unlike general learning strategies, apprenticeship learning allows newly-jointed SUs to learn from the expert nodes with mature experience, which makes the joint optimization algorithm converge faster and achieve better performance. Our future work will aim to adopt the apprenticeship learning strategy to accelerate the learning process in CRN.

Author Contributions: The individual contributions of authors are as follows. Y.D. developed the joint design scheme and wrote the paper. C.C. directed the research and contributed to the refinement of the algorithm. The analysis and performance discussion were performed by P.M. The paper was subsequently revised by L.X.

Funding: This research received no external funding.

Acknowledgments: The authors wish to thank the editor and anonymous referees for their helpful comments in improving the quality of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hossain, E.; Niyato, D.; Kim, D.I. Evolution and Future Trends of Research in Cognitive Radio: A Contemporary Survey. *Wirel. Commun. Mob. Comput.* **2015**, *15*, 1530–1564. [CrossRef]
2. Ahmad, A.; Ahmad, S.; Rehmani, M.H.; Hassan, N.U. A Survey on Radio Resource Allocation in Cognitive Radio Sensor Networks. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 888–917. [CrossRef]
3. Ma, Y.; Zhou, L.; Liu, K. A Subcarrier-Pair based Resource Allocation Scheme Using Proportional Fairness for Cooperative OFDM-based Cognitive Radio Networks. *Sensors* **2013**, *13*, 10306–10332. [CrossRef] [PubMed]
4. Huang, J.; Zeng, X.; Jian, X.; Tan, X.; Zhang, Q. Opportunistic Capacity-based Resource Allocation for Chunk-based Multi-carrier Cognitive Radio Sensor Networks. *Sensors* **2017**, *17*, 175. [CrossRef]
5. Zareei, M.; Islam, A.K.M.M.; Baharun, S.; Vargasrosales, C.; Azpilicueta, L.; Mansoor, N. Medium Access Control Protocols for Cognitive Radio Ad Hoc Networks: A Survey. *Sensors* **2017**, *17*, 2136. [CrossRef] [PubMed]
6. Ruby, E.D.K.; Saranya, N.; Santhkumar, W.E. A survey on distributed channel selection technique using surf algorithm for information transfer in multi-hop cognitive radio networks. *Int. Conf. Comput. Sci. Comput. Intell.* **2014**, *1*, 96–100.
7. Liu, Y.; Cai, L.X.; Shen, X.S. Spectrum-Aware Opportunistic Routing in Multi-Hop Cognitive Radio Networks. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 1958–1968. [CrossRef]
8. Ding, L.; Melodia, T.; Batalama, S.N.; Matyjas, J.D. Distributed Routing, Relay Selection, and Spectrum Allocation in Cognitive and Cooperative Ad Hoc Networks. In Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Boston, MA, USA, 21–25 June 2010; pp. 1–9.
9. Lai, L.; Wang, J.; Huang, A.; Shan, H. Routing and Resource Allocation with Collision Constraint in Multi-Hop Cognitive Radio Networks. In Proceedings of the GLOBECOM Workshops, Anaheim, CA, USA, 3–7 December 2012; pp. 974–979.
10. Amini, R.M.; Dziong, Z. An Economic Framework for Routing and Channel Allocation in Cognitive Wireless Mesh Networks. *IEEE Trans. Netw. Serv. Manag.* **2014**, *11*, 188–203. [CrossRef]
11. Royer, E.M.; Toh, C.K. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. *IEEE Pers. Commun.* **2002**, *6*, 46–55. [CrossRef]
12. Bkassiny, M.; Li, Y.; Jayaweera, S.K. A Survey on Machine-Learning Techniques in Cognitive Radios. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1136–1159. [CrossRef]
13. Raj, V.; Dias, I.; Tholeti, T.; Kalyani, S. Spectrum Access in Cognitive Radio Using A Two Stage Reinforcement Learning Approach. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 20–34. [CrossRef]
14. Al-Rawi, H.A.A.; Yau, K.L.A.; Mohamad, H. A Reinforcement Learning-based Routing Scheme for Cognitive Radio Ad Hoc Networks. In Proceedings of the 7th IFIP Wireless and Mobile Networking Conference (WMNC), Vilamoura, Portugal, 20–22 May 2014; pp. 1–8.
15. Chen, X.; Zhao, Z.; Zhang, H. Stochastic Power Adaptation with Multiagent Reinforcement Learning for Cognitive Wireless Mesh Networks. *IEEE Trans. Mobile Comput.* **2013**, *12*, 2155–2166. [CrossRef]
16. Pourpeighambar, B.; Dehghan, M.; Sabaei, M. Non-Cooperative Reinforcement Learning based Routing in Cognitive Radio Networks. *Comput. Commun.* **2017**, *106*, 11–23. [CrossRef]

17. Du, Y.; Zhang, F.; Xue, L. A Kind of Joint Routing and Resource Allocation Scheme based on Prioritized Memories-Deep Q Network for Cognitive Radio Ad Hoc Networks. *Sensors* **2018**, *18*, 2119. [CrossRef] [PubMed]
18. El-Sherif, A.A.; Mohamed, A.; Hu, Y.C. Joint Routing and Resource Allocation for Delay Sensitive Traffic in Cognitive Mesh Networks. In Proceedings of the IEEE Globecom Workshops, Houston, TX, USA, 5–9 December 2011.
19. Singh, K.; Moh, S. An Energy-Efficient and Robust Multipath Routing Protocol for Cognitive Radio Ad Hoc Networks. *Sensors* **2017**, *17*, 2027. [CrossRef] [PubMed]
20. Al-Rawi, H.A.A.; Yau, K.L.A. Route Selection for Minimizing Interference to Primary Users in Cognitive Radio Networks: A Reinforcement Learning Approach. In Proceedings of the IEEE Symposium on Computational Intelligence for Communication Systems and Networks (CIComms), Singapore, 16–19 April 2013; pp. 24–30.
21. Wellens, M.; Riihijarvi, J.; Mahonen, P. Evaluation of Adaptive MAC-Layer Sensing in Realistic Spectrum Occupancy Scenarios. In Proceedings of the IEEE Symposium on New Frontiers in Dynamic Spectrum, Singapore, 6–9 April 2010; pp. 1–12.
22. Xu, Y.; Wang, W. Wireless Mesh Network in Smart Grid: Modeling and Analysis for Time Critical Communications. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 3360–3371. [CrossRef]
23. Cao, Y.; Duan, D.; Cheng, X.; Yang, L. Qos-Oriented Wireless Routing for Smart Meter Data Collection: Stochastic Learning on Graph. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 4470–4482. [CrossRef]
24. El-Sherif, A.A.; Mohamed, A. Joint Routing and Resource Allocation for Delay Minimization in Cognitive Radio based Mesh Networks. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 186–197. [CrossRef]
25. Li, J.H.; Tian, N.S. Analysis of the Discrete Time Geo/Geo/1 Queue with Single Working Vacation. *Qual. Technol. Quantit. Manag.* **2016**, *5*, 77–89. [CrossRef]
26. Hilhorst, H.J.; Appert-Rolland, C. Mixed-Strategy Nash Equilibrium for A Discontinuous Symmetric N-Player Game. *J. Phys. Math. Gen.* **2017**, *51*, 095001. [CrossRef]
27. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv*, 2013; arXiv:1312.5602.
28. Littman, M.L. *A Unified Analysis of Value-Function-Based Reinforcement Learning Algorithms*; MIT Press: Cambridge, MA, USA, 1999.
29. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control Through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529. [CrossRef] [PubMed]