Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 960584, 22 pages
http://dx.doi.org/10.1155/2014/960584

Research Article

Reinforcement Learning for Routing in Cognitive Radio Ad Hoc Networks

Hasan A. A. Al-Rawi,[1] Kok-Lim Alvin Yau,[1] Hafizal Mohamad,[2] Nordin Ramli,[2] and Wahidah Hashim[2]

[1] *Department of Computer Science and Networked Systems, Sunway University, No. 5 Jalan Universiti, Bandar Sunway, 46150 Petaling Jaya, Selangor, Malaysia*
[2] *Wireless Network and Protocol Research Lab, MIMOS Berhad, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia*

Correspondence should be addressed to Kok-Lim Alvin Yau; koklimy@sunway.edu.my

Received 13 April 2014; Accepted 31 May 2014; Published 16 July 2014

Academic Editor: Su Fong Chien

Cognitive radio (CR) enables unlicensed users (or secondary users, SUs) to sense for and exploit underutilized licensed spectrum owned by the licensed users (or primary users, PUs). Reinforcement learning (RL) is an artificial intelligence approach that enables a node to observe, learn, and make appropriate decisions on action selection in order to maximize network performance. Routing enables a source node to search for a least-cost route to its destination node. While there have been increasing efforts to enhance the traditional RL approach for routing in wireless networks, this research area remains largely unexplored in the domain of routing in CR networks. This paper applies RL in routing and investigates the effects of various features of RL (i.e., reward function, exploitation, and exploration, as well as learning rate) through simulation. New approaches and recommendations are proposed to enhance the features in order to improve the network performance brought about by RL to routing. Simulation results show that the RL parameters of the reward function, exploitation, and exploration, as well as learning rate, must be well regulated, and the new approaches proposed in this paper improves SUs' network performance without significantly jeopardizing PUs' network performance, specifically SUs' interference to PUs.

1. Introduction

Cognitive radio (CR) has been proposed to enable unlicensed users (or secondary users, SUs) to exploit the underutilized licensed channels (or white spaces) owned by the licensed users (or primary users, PUs). Most traditional routing schemes adopt a rule-based approach [1] in which each node keeps and follows a set of predefined rules in its action selection for different network conditions; and this may not suit CR due to its intrinsic characteristic of the *dynamicity* and *unpredictability* of the network conditions (i.e., PUs' activities and channel quality) which require context awareness and intelligence [2–4]. Context awareness enables a SU node to observe the operating environment; while intelligence enables the SU node to learn and make action selection that maximizes network performance as time goes by. These capabilities are essential as the rule-based approach may not be feasible to define actions for all possible sets of network conditions in CR networks (CRNs).

Reinforcement learning (RL) [5], which is an artificial intelligence approach, has been applied to achieve context awareness and intelligence in CRNs [2]. This article presents a simulation study on the application of RL to routing in CRNs. Firstly, the traditional RL approach is applied in a routing scheme, which we call Cognitive Radio Q-routing (CRQ-routing). Next, a RL feature, namely reward function, is investigated. An enhanced RL-based routing scheme called weighted cognitive radio Q-routing (WCRQ-routing) is proposed. Subsequently, other two RL features, namely, exploitation and exploration, as well as learning rate, are investigated. The network performance of RL-based routing schemes can be enhanced by regulating the RL features; hence, new enhancements are proposed for the reward function, exploitation, and exploration, as well as learning rate.

CRQ-routing is a spectrum-aware scheme that finds least-cost routes in CRNs taking into account the dynamicity and unpredictability of the channel availability and channel quality. Simulation results show that CRQ-routing and its enhancements minimize SUs' interference to PUs, SUs' end-to-end delay, and SUs' packet loss rate, as well as maximizing SUs' throughput.

Our contributions are as follows.

(i) Section 3 presents CRQ-routing which applies the traditional RL approach.

(ii) Section 4 investigates different reward representations for network performance enhancement. In addition to CRQ-routing, this section investigates a variant of the reward function which we conveniently call WCRQ-routing. WCRQ-routing applies a weight factor ω to adjust the tradeoff between PUs' and SUs' network performance. Performance enhancement achieved by CRQ-routing and WCRQ-routing for different PU utilization levels (PULs) and packet error rate (PER) are compared with the traditional shortest path (SP) routing scheme and the optimal primary user-aware shortest path (PASP) routing scheme.

(iii) Section 5 investigates the effects of exploitation and exploration on network performance. A simple and pragmatic exploration approach called dynamic softmax (DS) is proposed to dynamically regulate the frequency of exploration according to the dynamicity of the operating environment. Performance enhancement achieved by DS is compared with two traditional exploration approaches, namely, ε-greedy and softmax.

(iv) Section 6 investigates the effects of learning rate on network performance. A simple and pragmatic learning rate adjustment approach called the counterapproach (CA), which is based on the traditional win-or-learn-fast policy hill climbing (or win-lose) [6], is proposed to dynamically regulate the learning rate according to the dynamicity of the operating environment. Performance enhancement achieved by CA is compared with the traditional approach, namely, win-lose.

Simulation experiment, results, and discussions are presented in each of the sections. Finally, Section 7 concludes this paper.

2. Related Work

While most researches focus on the enhancement of either PUs' or SUs' network performance [7–10], this paper focuses on both PUs' and SUs'. CRQ-routing minimizes SUs' interference to PUs without causing significant detrimental effects on SUs' network-wide performance.

In [7–10], either one or both of the following requirements are applicable. Firstly, information on PUs' and SUs' physical locations is essential. The associated challenges are additional energy consumption, increased hardware cost, and the availability of the physical location information in indoor scenarios [2]. Secondly, network-wide information such as link cost is essential; however, it is difficult to obtain up-to-date information for the entire network in the presence of dynamicity and unpredictability of the channel availability and channel quality in CRNs. For instance, a reactive routing scheme requires a SU destination node to confirm a route prior to data transmission; however, due to the dynamicity and unpredictability of the channel availability and channel quality, a new route may have expired before routing information reaches the SU destination node. CRQ-routing does not require geographical and network-wide information, and it adopts a per-hop routing approach (rather than an end-to-end routing approach) that enables each SU intermediate node to make routing decision for a single hop to its next-hop node based on local information.

The application of RL to routing schemes in CRNs has been limited, such as [11], although it has been shown to improve routing performance in various traditional wireless networks [12, 13]. In [11], the SUs' network performance is shown to be enhanced, while in CRQ-routing, both PUs' and SUs' network performances are enhanced. Using RL, CRQ-routing integrates route discovery mechanism with channel selection. CRQ-routing is a multipath routing scheme that enables a SU node to maintain multiple routes, and this can be well incorporated into RL through its feature called exploration. Generally speaking, the existence of multiple routes helps to enhance network reliability and to achieve load balancing among various routes. This is because a SU can automatically switch its route to another one during route recovery in the event of route failure.

3. CRQ-Routing: Application of the Traditional Reinforcement Learning Approach to Routing

This section presents CRQ-routing that takes account of the PUs' and SUs' network performance by minimizing SUs' interference to PUs along a route without significantly jeopardizing SUs' network-wide performance. It applies a traditional RL approach called Q-learning [14, 15], which is a popular RL approach. CRQ-routing enables a SU to observe its local operating environment regularly and subsequently to learn an action selection policy through exploring various routes, and finally to choose routes with enhanced network performance (i.e., lower SUs' interference to PUs, lower SUs' end-to-end delay, lower SUs' packet loss rate, and higher SUs' throughput). Generally speaking, RL enables a SU node to

(a) estimate the dynamic link cost. This allows a SU to learn about and adapt to the local network conditions (i.e., PUs' activities and channel quality) which are dynamic and unpredictable in nature,

Table 1: CRQ-routing model embedded at SU node i.

State	$s_t^i \in S = \{1, 2, \ldots, N-1\}$, each state s_t^i representing a SU destination node n. N represents the number of SUs in the entire network.
Action	$a_t^i \in A^i = \{1, 2, \ldots, J\}$, each action a_t^i representing the selection of a SU next-hop node j along with its operating channel. J represents the number of SU i's neighboring SU nodes.
Cost	$r_t^i(a_t^i)$ represents the link-layer delay incurred to successfully deliver a packet from SU node i to SU neighbor node $a_t^i = j$, including retransmission delays as a result of PU-SU packet collision and packet loss.

(b) search for the best-possible route using information observed from the local operating environment and information received from neighboring nodes,

(c) incorporate a wide range of factors that can affect the routing performance into consideration, including both PUs' activities and channel quality.

By choosing links with lower link-layer delay, which is the time duration required to deliver a SU's packet to a next-hop node successfully, SUs' interference to PUs can be reduced and SUs' end-to-end network performance can be enhanced. Note that the link-layer delay includes the time duration incurred by retransmission as a result of PU-SU packet collisions. The RL model for CRQ-routing is shown in Table 1, and it is embedded in a SU node i. Using CRQ-routing, each SU chooses a next-hop node and channel pair as part of a route. There are three key representations for the RL model as follows.

(i) State $s_t^i \in S = \{1, 2, \ldots, N-1\}$ represents a SU destination node n, where N represents the number of SUs in the entire network.

(ii) Action $a_t^i \in A^i = \{1, 2, \ldots, J\}$ represents the selection of a next-hop SU neighbor node j along with its operating channel, where J represents the number of SU i's neighboring SU nodes.

(iii) Cost $r_t^i(a_t^i)$, which indicates the consequence upon taking action a_t^i, represents the link-layer delay of a SU communication node pair, namely nodes i and j. The link-layer delay includes retransmission delays caused by packet loss and PU-SU packet collision. Hence, the end-to-end delay (or the accumulated link-layer delay) reflects the accumulated SUs' interference to PUs; and by achieving lower delay, the interference level can be reduced.

Each SU node i keeps track of Q-value $Q_t^i(s_t^i, a_t^i)$, which relates the three representations, in its Q-table (i.e., routing table). For each state-action pair, the Q-value represents the accumulated link-layer delay of a route leading to SU destination node s_t^i by choosing a SU next-hop node $a_t^i = j$. At time t, SU i selects a SU next-hop node $a_t^i = j$ as part of a route to reach destination node s_t^i; and upon a successful transmission at time $t+1$, it receives an end-to-end delay estimate for the route, namely $Q_t^j(s_t^j, k)$, from node j and estimates the link-layer delay $r_{t+1}^i(j)$. Note that the link-layer delay is dynamic and unpredictable due to the nature of the PUs' activities in which there are different levels of PUL. In general, the link-layer delay increases with PUL in a channel. The Q-value $Q_t^i(s_t^i, a_t^i = j)$ is updated as follows:

$$\begin{aligned} Q_{t+1}^i\left(s_t^i, j\right) \longleftarrow{} & (1-\alpha)\, Q_t^i\left(s_t^i, j\right) \\ & + \alpha\left(r_{t+1}^i(j) + \min_{k \in A^j} Q_t^j\left(s_t^j, k\right)\right), \end{aligned} \quad (1)$$

where $0 \le \alpha \le 1$ is the learning rate and node $k \in A^j$ is an upstream node of SU node j. A SU node i adopts a policy $\pi_{t+1}^i(s_t^i)$ that chooses a SU next-hop node with the minimum cost as follows:

$$\pi_{t+1}^i\left(s_t^i\right) = \underset{a \,\in\, A^i}{\operatorname{argmin}}\left(Q_t^i\left(s_t^i, a\right)\right). \quad (2)$$

In the next section, we present a variant of the reward representation in RL to further enhance PUs' and SUs' network performance.

4. WCRQ-Routing: A Variant of the Reward Representation

WCRQ-routing incorporates a weight factor into the reward representation of CRQ-routing to adjust the tradeoff between PUs' and SUs' network performance. WCRQ-routing provides further enhancement on SUs' network performance without jeopardizing PUs' network performance. The main difference between WCRQ-routing and CRQ-routing is the cost (or negative reward) representation as follows.

(i) CRQ-routing enables a SU to learn about the accumulated cost in terms of link-layer delay along a route.

(ii) WCRQ-routing enables a SU to learn about the accumulated cost in terms of the number of SUs' packet retransmissions and the packet queue length of SUs along a route. Hence, WCRQ-routing enables a SU to take account of packet retransmission which further improves the PUs' network performance, and network performance which further improves the SUs' network performance. WCRQ-routing also incorporates a weight factor ω which adjusts the tradeoff between PUs' and SUs' network performance.

The RL model for WCRQ-routing is shown in Table 2, and it is embedded in a SU node i. The state and action representations are similar for CRQ-routing (see Table 1) and WCRQ-routing, and so only the reward representation is shown. The reward representation for the RL model is as follows.

Table 2: Reward representation for WCRQ-routing model embedded at SU node i.

Cost	$r_t^i(a_t^i) = \omega r_t^{i,j} + (1-\omega)q_t^j$ where $r_t^{i,j}$ represents the number of retransmissions for a packet sent from SU node i to SU neighbor node j at time t, while q_t^j represents the number of packets in the queue of SU neighbor node j. Weight factor $\omega = [0,1]$ is used to adjust the tradeoff between PUs' and SUs' network performance.

(i) Cost $r_t^i(a_t^i) = \omega r_t^{i,j} + (1-\omega)q_t^j$ has two components: (1) $r_t^{i,j}$ represents the number of retransmissions for a packet sent from SU node i to SU neighbor node j at time t as a result of PU-SU packet collisions and SU packet loss and (2) q_t^j represents the number of packets in the queue of SU neighbor node j. Both $r_t^{i,j}$ and q_t^j values in reward $r_t^i(a_t^i)$ are normalized to $[0,1]$. The weight factor $\omega = [0,1]$ adjusts the tradeoff between PUs' and SUs' network performance.

Similar to CRQ-routing, each SU node i keeps track of Q-values $Q_t^i(s_t^i, a_t^i)$ in its Q-table (i.e., routing table). For each state-action pair, the Q-value represents a weighted cost that takes account of the number of packet retransmissions and packet queue length of SUs along a route leading to SU destination node s_t^i by choosing a SU next-hop node a_t^i. At time t, SU i selects a SU next-hop node $a_t^i = j$ as part of a route to reach destination node s_t^i and upon a successful transmission, it receives an estimate of the weighted cost for the route, namely $Q_t^j(s_t^j, k)$, from node j and estimates reward $r_{t+1}^i(j)$, which takes into account the number of retransmissions for a packet and the number of packets in the queue of SU neighbor node j at time $t+1$. Note that both number of retransmissions for a packet and number of packets in the queue of a SU's neighbor node are dynamic and unpredictable due to the nature of the PUs' activities in which there are different levels of PUL. In general, $r_{t+1}^i(j)$ increases with PUL in a channel. The Q-value $Q_t^i(s_t^i, a_t^i)$ is updated using (1) and a SU next-hop node with the minimum cost is chosen using (2).

The rest of this section is organized as follows. Section 4.1 presents simulation setup and parameters. Section 4.2 presents a comparison of network performance achieved by CRQ-routing, WCRQ-routing, and two baseline routing schemes, namely, shortest path (SP) and PU aware shortest path (PASP) routing schemes. Section 4.3 presents the effects of the weight factor ω in reward representation on network performance.

4.1. Simulation Setup and Parameters. Figure 1 shows the system model which is a multihop CRN with N SUs and K PUs [16]. Each PU transmits in one of the K different channels. In Figure 1, we consider a SU source node A and a SU destination node G.

We compare the network performance achieved by CRQ-routing and WCRQ-routing with nonlearning approaches, specifically, the traditional SP routing and the optimal PASP routing approaches. SP routing selects a route that has the minimum number of hops and this has been shown to improve the end-to-end network performance in traditional networks. PASP routing selects a route that has the minimum accumulated amount of PUs' activities. This means that the route encounters the least number of PUs, and so the PUL along the route may be the lowest. However, PASP routing may not be feasible in practice as it is a centralized approach that requires network-wide information of PUL for each link and channel. Nevertheless, PASP routing is an optimal approach that minimizes SUs' interference to PUs and so it serves as a good comparison in our simulation study.

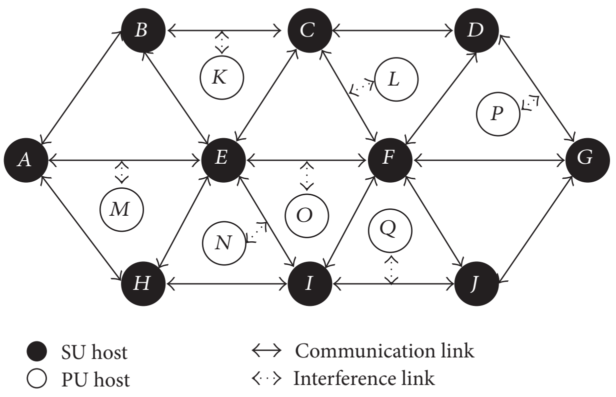


Figure 1: Network scenario.

There are four network performance metrics as follows.

(i) *PU-SU collision probability* indicates the level of SUs' interference to PUs. Reducing this metric improves the PUs' network performance. This metric is a ratio of the number of PU-SU collisions to the number of SUs' packet transmissions.

(ii) *SU end-to-end delay* includes the transmission, processing, backoff, and queuing delays along a route.

(iii) *SU packet loss rate* is a ratio of the number of packet loss to the total number of packets sent.

(iv) *SU throughput* is the number of arriving packets per second (pps) at the SU destination node.

The simulation compares the aforementioned performance metrics with respect to different levels of PULs for CRQ-routing, WCRQ-routing, SP routing, and PASP routing. The dynamicity of the PUs' activities (or PUL) is represented by PU arrival rate μ_{PUL}, and each scenario has a certain level of unpredictability of the PUs' activities (or the standard deviation of PU arrival rate) which is represented by σ_{PUL}.

Table 3 shows a summary of the simulation parameters and values. Generally speaking, each simulation is run for 100 seconds and repeated 50 times with different random seeds. The simulation reporting interval is 1 second which indicates that, at each second of the simulation running time, a mean value of the simulation result is calculated. The default

Table 3: Simulation parameters and values.

Category	Parameter	Value
SU	SU's transmission delay, $d_{\mathrm{SU}}^{\mathrm{tr}}$	1.0 ms
	Processing delay, $d_{\mathrm{SU}}^{\mathrm{pr}}$	1.0 ms
	Mean arrival rate, λ_{SU}	0.6
	Learning rate, α	0.5
	WCRQ-routing weight factor, ω	0.5
PU	PU's transmission delay, $d_{\mathrm{PU}}^{\mathrm{tr}}$	1.2 ms
	Mean arrival rate (or PUL), μ_{PUL}	[0.0, 1.0]
	Standard deviation, σ_{PUL}	{0.0, 0.4, 0.8}
Channel	Mean PER, μ_{PER}	0.05
	Standard deviation of PER, σ_{PER}	0.025

number of SUs is $N = 10$, and PUs is $K = 19$. Each PU does not change its channel and operates in distinctive channels. Each PU activity in channel $k \in K$ is modeled as a Poisson process with mean $\lambda_{\mathrm{PUL}}^{k}(\mu_{\mathrm{PUL}}, \sigma_{\mathrm{PUL}})$, which is assigned using Box-Muller transform [17] according to an expected mean $\mu_{\mathrm{PUL}} = [0, 1]$ and standard deviation $\sigma_{\mathrm{PUL}} = [0, 1]$. The standard deviation of PUL is $\sigma_{\mathrm{PUL}} \in \{0.0, 0.4, 0.8\}$, which indicates low, medium, and high levels of unpredictability of the PUs' activities, and these values are chosen due to their significant effects to the results. Since the focus is on the comparison of network performance in regard to PUL, we assume the channels have low level of noise with $\mu_{\mathrm{PER}} = 0.05$ and $\sigma_{\mathrm{PER}} = 0.025$. Packets are generated at the SU source node using Poisson process with a mean arrival rate of $\lambda_{\mathrm{SU}} = 0.6$. Since the PUs have higher priority than SUs, their transmissions take longer; specifically, the SUs' transmission delay is $d_{\mathrm{SU}}^{\mathrm{tr}} = 1.0$ ms and the PUs' transmission delay is $d_{\mathrm{PU}}^{\mathrm{tr}} = 1.2$ ms. In order to model a simple queuing delay $d_{\mathrm{SU}}^{\mathrm{qu}}$, we assume a finite packet queue size of 1000 packets in each SU node with a constant processing delay $d_{\mathrm{SU}}^{\mathrm{pr}} = 1.0$ ms. The Q-values are initialized to 0 in order to encourage exploration at the start of the simulation. The SU learning rate is $\alpha = 0.5$ (see (1)). For WCRQ-routing, the weight factor is set to $\omega = 0.5$ so that there is a balanced tradeoff between PUs' and SUs' network performance. The effects of ω on the network performance are presented in Section 4.3.

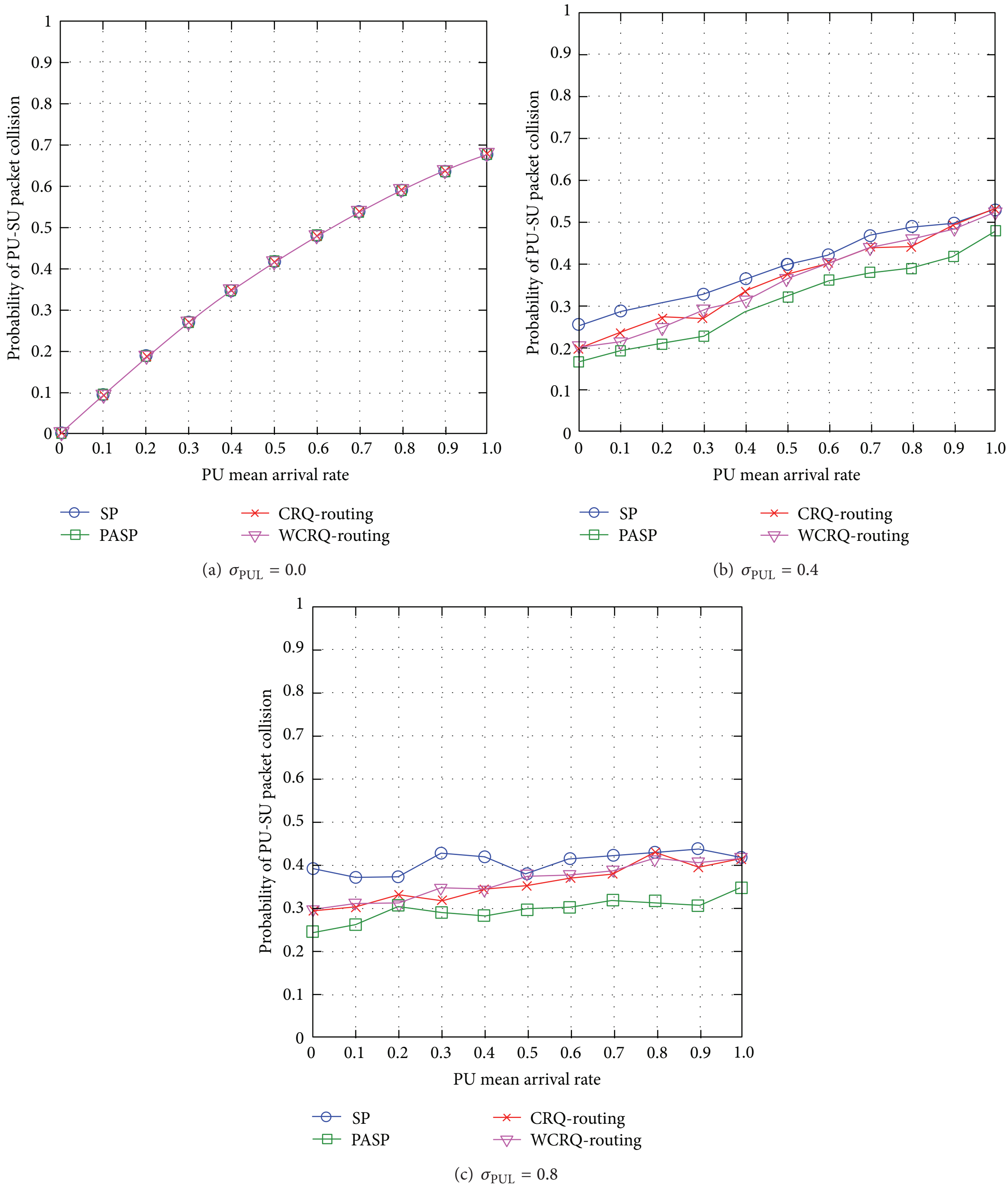
4.2. Comparison of CRQ-Routing, WCRQ-Routing, SP, and PASP Routing Schemes. We present simulation results for the four performance metrics in this section.

4.2.1. SUs' Interference to PUs. When the standard deviation of PUL is low $\sigma_{\mathrm{PUL}} = 0$, most next-hop node (or link) and channel pairs have the same PU mean arrival rate μ_{PUL}, so all routing schemes achieve similar probability of PU-SU packet collisions across a CRN (see Figure 2(a)). When the unpredictability level of PUL is $\sigma_{\mathrm{PUL}} = 0.4$, the link and channel pairs have greater difference in the levels of PU mean arrival rate μ_{PUL}, WCRQ-routing, and CRQ-routing choose routes that minimize SUs' interference to PUs. Both WCRQ-routing and CRQ-routing reduce collisions with PUs for up to 19% compared to SP routing, whereas the empirical PASP routing scheme, which provides the best results, reduces collisions with PUs for up to 30% compared to SP routing (see Figure 2(b)). When $\sigma_{\mathrm{PUL}} = 0.8$, the link and channel pairs have the greatest difference in the levels of PU mean arrival rate μ_{PUL}. Similar trends to the network scenario of $\sigma_{\mathrm{PUL}} = 0.4$ are observed although collisions with PUs have generally increased due to the increased unpredictability of PUs' activities (see Figure 2(c)).

Generally speaking, both WCRQ-routing and CRQ-routing achieves almost similar performance in terms of SUs' interference to PUs; and the SUs' interference to PUs increases with the PU mean arrival rate μ_{PUL} and the standard deviation of PUL σ_{PUL}.

4.2.2. SU End-to-End Delay. When the standard deviation of PUL is low $\sigma_{\mathrm{PUL}} = 0$, most link and channel pairs have the same PU mean arrival rate μ_{PUL}. The SU end-to-end delay increases with PU mean arrival rate μ_{PUL} (see Figure 3(a)). When the PU mean arrival rate μ_{PUL} is low, the SU end-to-end delay consists of mainly transmission and queuing delays, and when μ_{PUL} becomes higher (i.e., $\mu_{\mathrm{PUL}} > 0.3$), the retransmission and backoff delays caused by PU-SU packet collisions and overflow of SUs' packet queues increase contributing to higher SU end-to-end delay. WCRQ-routing reduces SU end-to-end delay for up to 72% compared to the other routing schemes. This is because WCRQ-routing takes into account the queue length of SUs along the route, so it chooses routes with lower network congestion contributing to lower SU end-to-end delay.

When the unpredictability level of PUL increases from $\sigma_{\mathrm{PUL}} = 0.4$ to $\sigma_{\mathrm{PUL}} = 0.8$, the link and channel pairs have greater difference in the levels of PU mean arrival rate μ_{PUL}. WCRQ-routing chooses routes that minimize the number of packet retransmissions caused by PU-SU packet collisions and overflow of SUs' packet queues contributing to lower SU end-to-end delay (see Figures 3(b) and 3(c)). With greater unpredictability level of PUL σ_{PUL}, it is easier for WCRQ-routing to find a route with better performance; hence, the lower SU end-to-end delay when $\sigma_{\mathrm{PUL}} = 0.8$ in Figure 3(c) compared to $\sigma_{\mathrm{PUL}} = 0.4$ in Figure 3(b), and WCRQ-routing achieves lower SU end-to-end delay for up to 89% when $\sigma_{\mathrm{PUL}} = 0.8$ compared to other routing schemes. Additionally,



(a) $\sigma_{\text{PUL}} = 0.0$

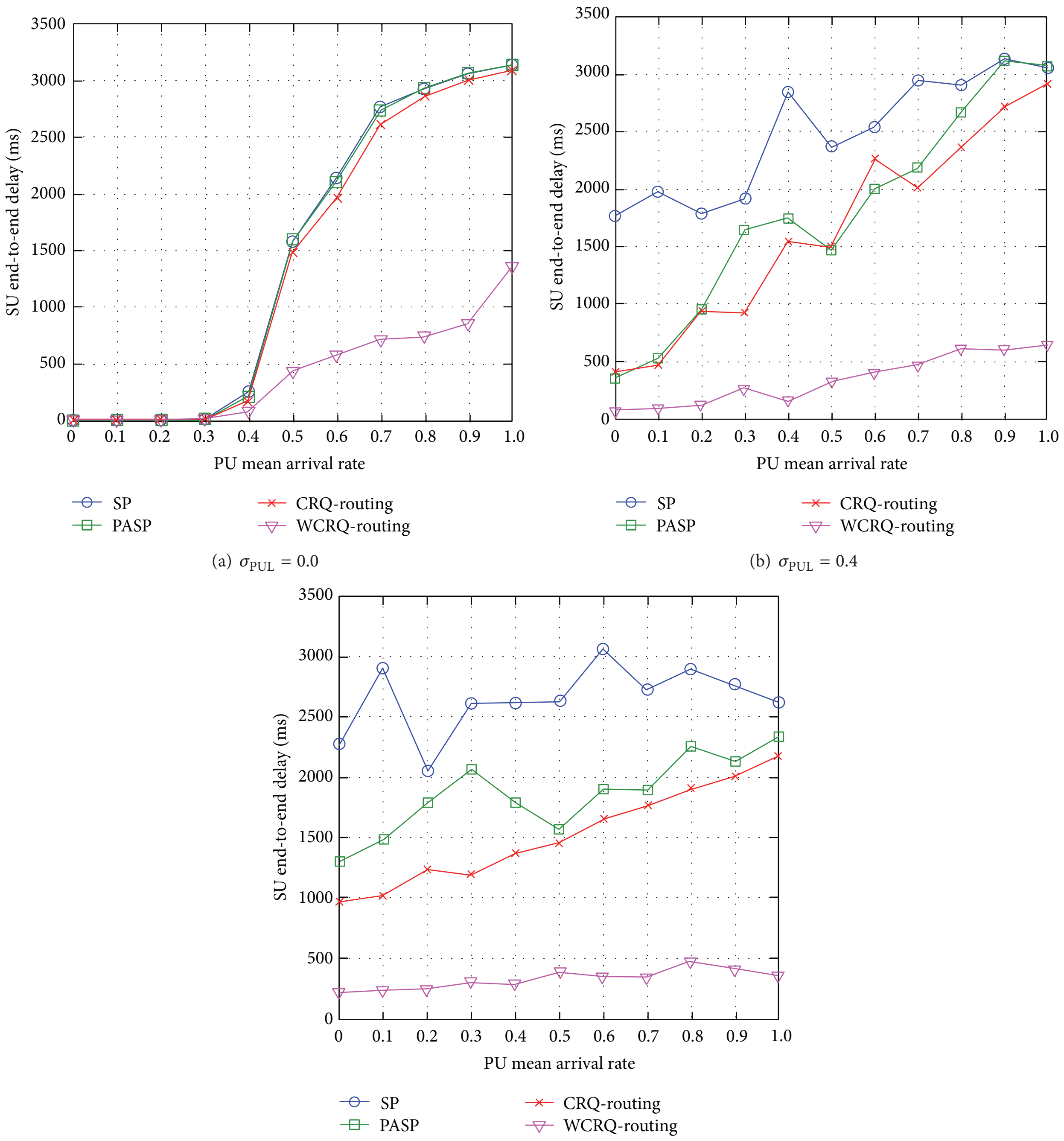
(b) $\sigma_{\text{PUL}} = 0.4$

(c) $\sigma_{\text{PUL}} = 0.8$

Figure 2: SUs' interference to PUs for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

there are two main observations. Firstly, the fluctuations of SU end-to-end delay are observed because the routes of SP routing and PASP routing are static as they are unaware of the unpredictability of PUL, while the routes of CRQ-routing and WCRQ-routing are dynamic in nature. Secondly, PASP routing and CRQ-routing deteriorate to the network performance of SP routing with increasing PU mean arrival rate μ_{PUL} because both schemes take account of PU mean arrival rate μ_{PUL} only in routing decision, and so when μ_{PUL} becomes higher, they may choose longer routes resulting in higher SU end-to-end delay. In contrast, WCRQ-routing takes account of network congestion as well.

Generally speaking, WCRQ-routing reduces SU end-to-end delay, and the metric increases with the PU mean arrival



(a) $\sigma_{\text{PUL}} = 0.0$

(b) $\sigma_{\text{PUL}} = 0.4$

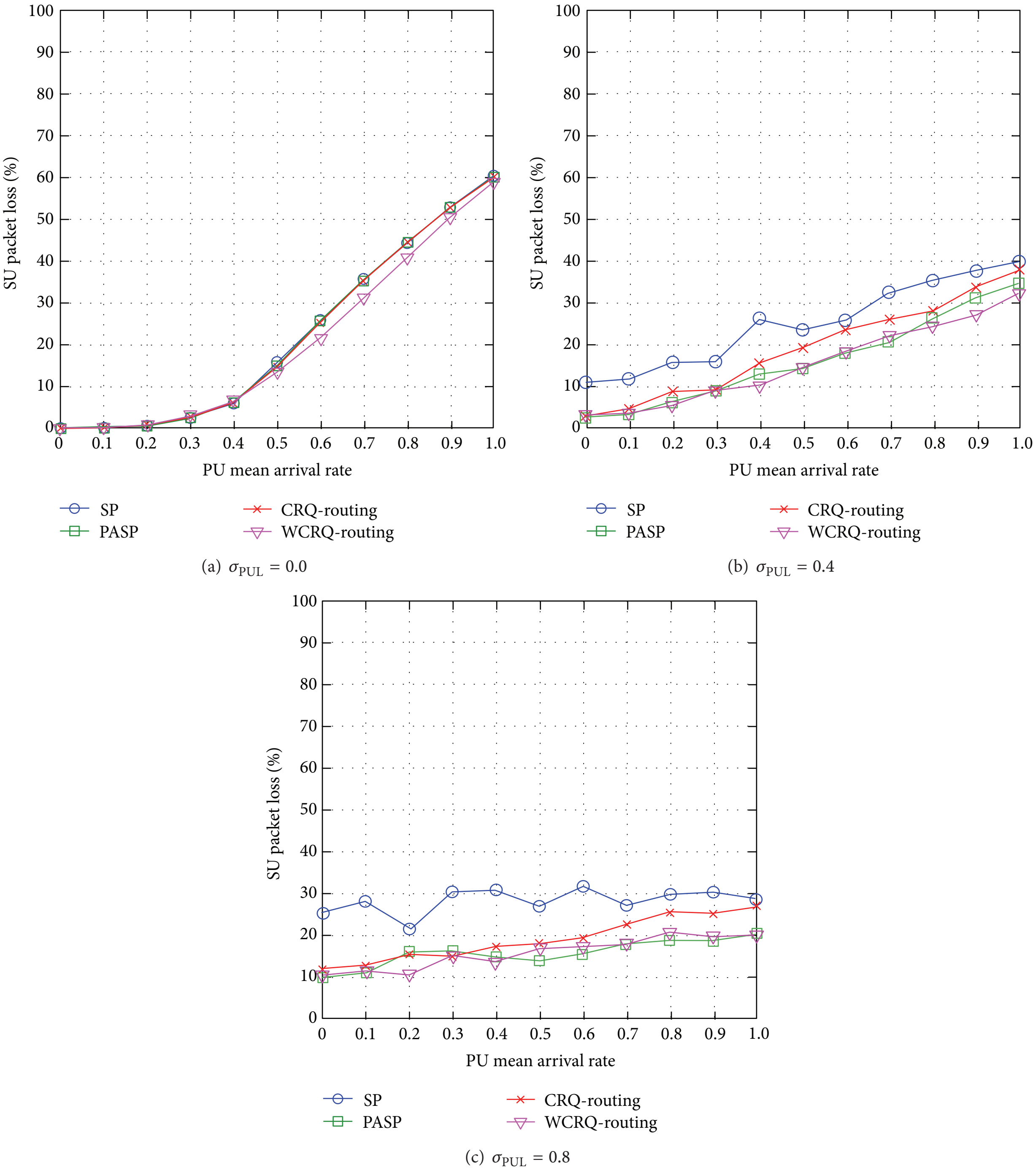
(c) $\sigma_{\text{PUL}} = 0.8$

FIGURE 3: SU end-to-end delay for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

4.2.3. SU Packet Loss. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0$, WCRQ-routing achieves load-balancing among the available routes as it takes into account the queue length of SUs along a route, so it reduces network congestion, and hence there is lower SU packet loss of up to 16% compared to the other routing schemes (see Figure 4(a)).

When the unpredictability level of PUL is $\sigma_{\text{PUL}} = 0.4$ and $\sigma_{\text{PUL}} = 0.8$, WCRQ-routing achieves almost similar network performance to PASP routing with lower packet loss for up to 56% compared to SP routing and 25% compared to CRQ-routing when $\sigma_{\text{PUL}} = 0.8$ (see Figures 4(b) and 4(c)). This is because WCRQ-routing chooses routes with lower network congestion in order to minimize packet retransmission caused by PU-SU packet collisions and overflow of SU's packet queues leading to lower number of packet



(a) $\sigma_{\text{PUL}} = 0.0$

(b) $\sigma_{\text{PUL}} = 0.4$

(c) $\sigma_{\text{PUL}} = 0.8$

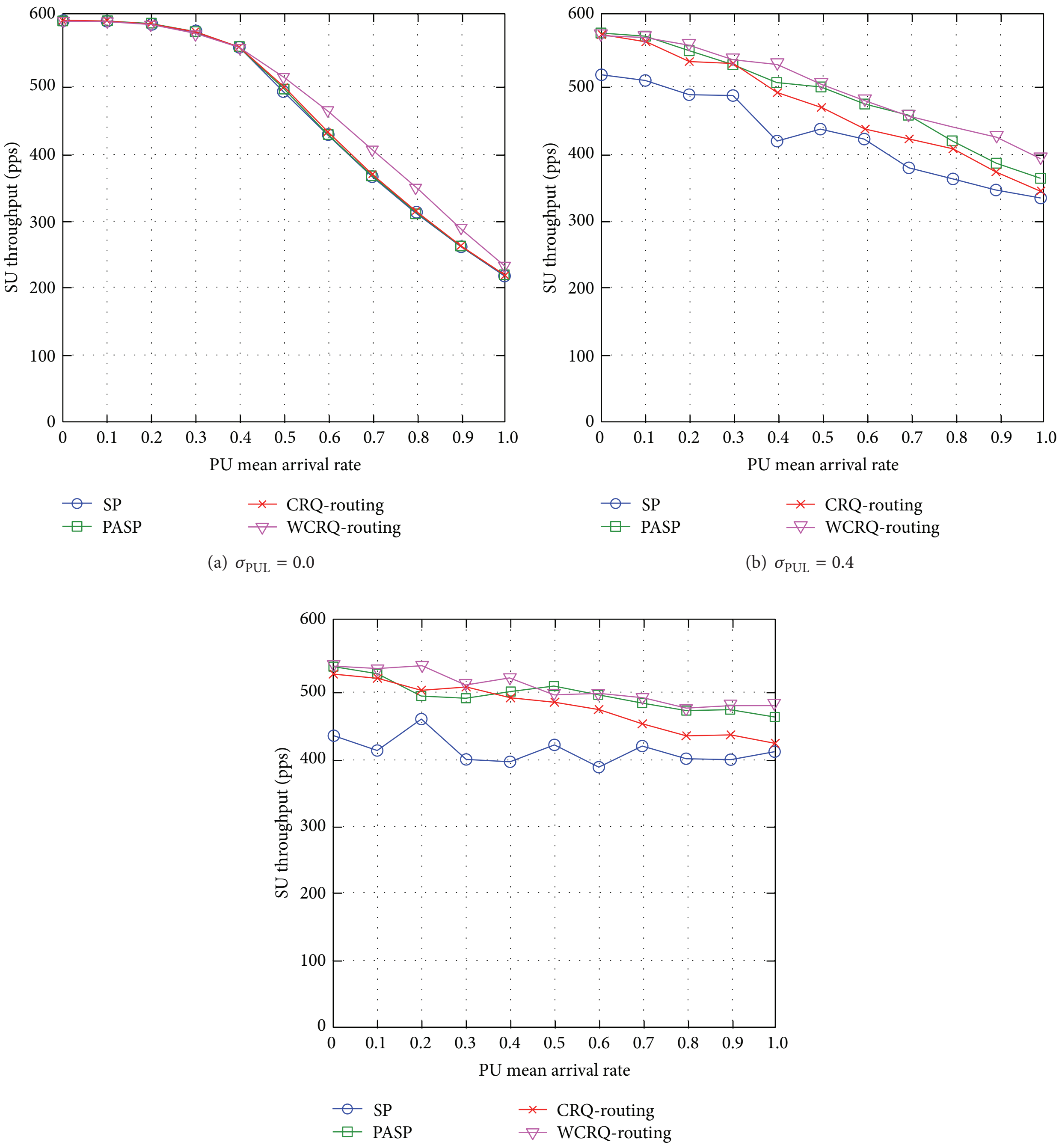
Figure 4: SU packet loss for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

retransmissions. Both network scenarios of $\sigma_{\text{PUL}} = 0.4$ and $\sigma_{\text{PUL}} = 0.8$ share almost similar trends although the SU packet loss has generally decreased in the case of $\sigma_{\text{PUL}} = 0.8$.

Generally speaking, WCRQ-routing reduces SU packet loss, and the metric increases with the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

4.2.4. SU Throughput. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0$, WCRQ-routing achieves load-balancing among available routes as it takes into account the queue length of SUs along a route, so it reduces network congestion, and hence there is higher SU throughput of up to 11% compared to the other routing schemes (see Figure 5(a)).

When the unpredictability level of PUL is $\sigma_{\text{PUL}} = 0.4$ and $\sigma_{\text{PUL}} = 0.8$, WCRQ-routing achieves higher SU throughput of up to 27% compared to SP routing, up to 12% compared to CRQ-routing, and up to 9% compared to PASP routing when $\sigma_{\text{PUL}} = 0.8$ (see Figures 5(b) and 5(c)). This is because



(a) $\sigma_{\text{PUL}} = 0.0$

(b) $\sigma_{\text{PUL}} = 0.4$

(c) $\sigma_{\text{PUL}} = 0.8$

FIGURE 5: SU throughput for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

WCRQ-routing chooses routes with lower network congestion in order to minimize packet retransmission caused by PU-SU packet collisions and overflow of SU's packet queues leading to higher SU throughput.

Generally speaking, WCRQ-routing increases SU throughput, and the metric reduces with the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

4.2.5. Section Summary. We summarize simulation outcomes for the comparison of CRQ-routing, WCRQ-routing, SP, and PASP routing schemes as follows.

(i) Network performance degrades as the dynamicity of PUs' activities μ_{PUL} increases.

(ii) WCRQ-routing and CRQ-routing minimize SUs' interference to PUs with respect to PUL in the

Table 4: Simulation parameters and values for investigating the effects of weight factor in reward representation.

Category	Parameter	Value
SU	Mean arrival rate, λ_{SU}	0.8
PU	Mean arrival rate (PUL), μ_{PUL}	{0.2, 0.4}
	Standard deviation, σ_{PUL}	0.5
Channel	Mean PER, μ_{PER}	{0.2, 0.4}
	Standard deviation of PER, σ_{PER}	0.2

presence of dynamicity and unpredictability of the channel availability.

(iii) WCRQ-routing enhances network performance of SUs compared to other routing schemes including lower SU end-to-end delay, lower SU packet loss, and higher SU throughput. Using a weight factor in WCRQ-routing, the cost represents two factors including the number of packet retransmissions and the packet queue length of SUs.

4.3. Effects of Weight Factor in Reward Representation. This section investigates the effects of the weight factor ω of WCRQ-routing on network performance. WCRQ-routing applies a weight factor ω to adjust the tradeoff between PUs' and SUs' network performance. Based on Table 2, with a higher value of ω, there is greater consideration on $r_t^{i,j}$ which represents the number of retransmissions for a packet sent from SU node i to SU neighbor node j at time t and indicates the probability of PU-SU packet collision, and lesser consideration on SU neighbor node j's queue length q_t^j which indicates SU network congestion. This means a higher value of weight factor ω improves the PUs' network performance, while a lower value of ω improves the SUs' network performance.

While Table 3 presents the default simulation parameters and values, Table 4 presents the specific simulation parameters and values for this investigation. Generally speaking, we simplify the simulation values in order to focus on the effects of weight factor ω. This explains why we consider two different PUL μ_{PUL} and PER μ_{PER} values only and a fixed value of the standard deviation of PUL σ_{PUL}. We increase the network congestion level with increased SU mean arrival rate to $\lambda_{\text{SU}} = 0.8$.

In this section, we assume that when PU-SU packet collision occurs, a SU packet is transmitted successfully while the PU's packet is lost; therefore, the PUs' activities can be easily affected and it is prone to SUs' interference. We present the simulation results for the four performance metrics in the rest of this section.

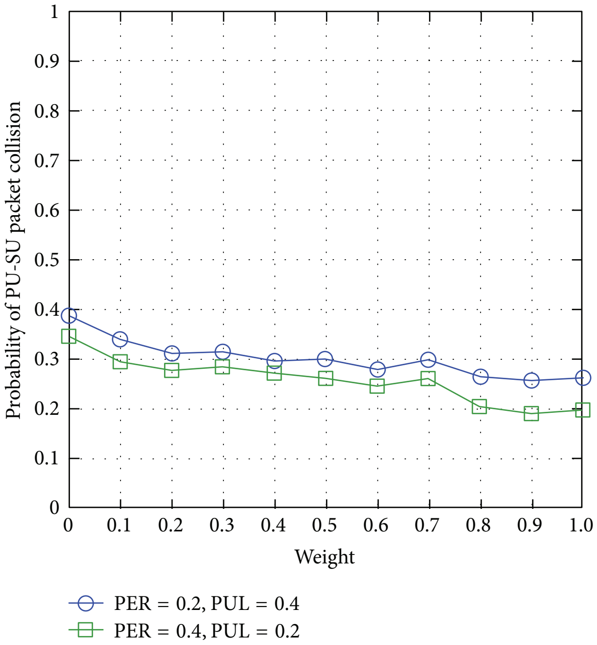


Figure 6: SUs' interference to PUs for varying weight factor ω.

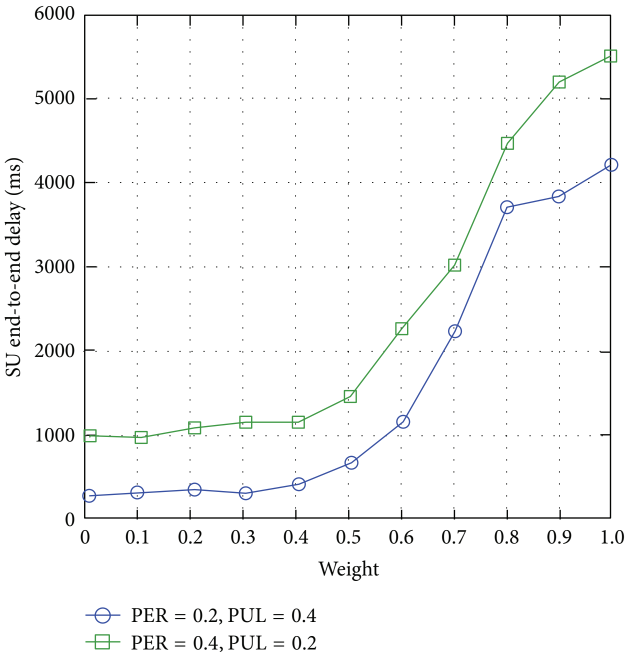


Figure 7: SU end-to-end delay for varying weight factor ω.

4.3.1. SUs' Interference to PUs. Figure 6 shows that the SUs' interference to PUs decreases with the weight factor ω of WCRQ-routing. Hence, a higher value of weight factor ω improves the PUs' network performance, while a lower value of ω improves SUs' network performance. In addition, with respect to PUL and PER, Figure 6 shows that PUL has greater effects on PUs' network performance compared to PER because PUL has a direct effect on the PU-SU packet collisions; hence, higher PULs indicate higher probability of PU-SU collisions.

4.3.2. SU End-to-End Delay. Figure 7 shows that the SU end-to-end delay increases with the weight factor ω of WCRQ-routing. Hence, a higher value of weight factor ω improves the PUs' network performance; while a lower value of ω improves

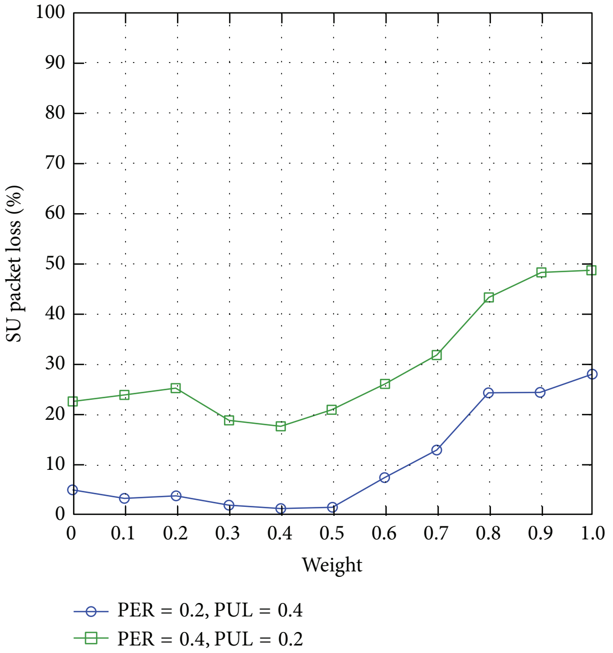


Figure 8: SU packet loss for varying weight factor ω.

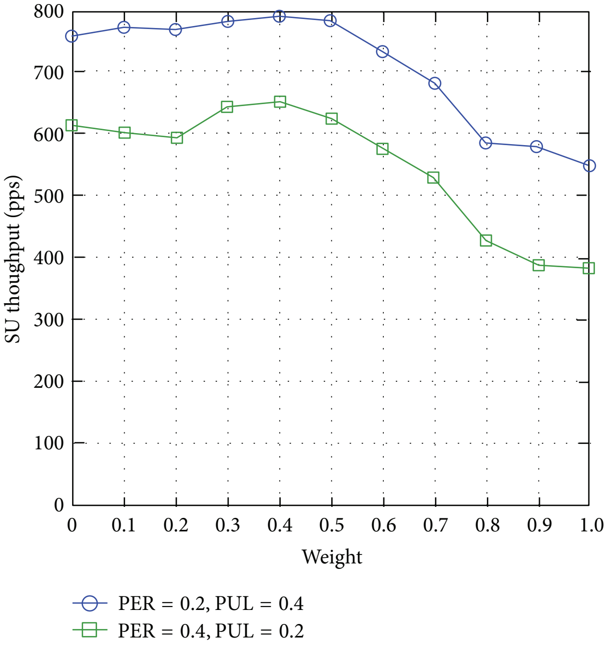


Figure 9: SU throughput for varying weight factor ω.

SUs' network performance. In addition, with respect to PUL and PER, Figure 7 shows that PER has greater effects on SUs' network performance compared to PUL because PER has a direct effect on the packet length and a SU's packet can be transmitted successfully during a PU-SU packet collision. With a higher value of PER (or a noisier channel), SU network congestion increases due to increasing number of SU packet retransmissions leading to higher SU end-to-end delay.

4.3.3. SU Packet Loss. Figure 8 shows that the SU packet loss increases with the weight factor ω of WCRQ-routing. The explanations leading to this circumstance are similar to those found in the investigation of SU end-to-end delay (see Section 4.3.2). Interestingly, Figure 8 shows that when weight factor is $\omega \cong 0.5$, WCRQ-routing achieves the lowest packet loss. At $\omega = 0.5$, a SU node i gives equal consideration to both link quality $r_t^{i,j}$, which aims to avoid routes with higher PUs' activities in order to reduce SUs' backoff delays, and SU j's queue length q_t^j, both of which reduce the overflow of SUs' queues and SUs' packet loss.

4.3.4. SU Throughput. Figure 9 shows that the SU throughput reduces with the weight factorω of WCRQ-routing. The explanations leading to this circumstance are similar to those found in the investigation of SU end-to-end delay (see Section 4.3.2); and when weight factor is $\omega \cong 0.5$, WCRQ-routing achieves the highest throughput.

4.3.5. Section Summary. We summarize simulation outcomes for the effects of weight factor in reward representation as follows.

(i) A higher value of weight factor ω improves the PUs' network performance; while a lower value of ω improves the SUs' network performance.

(ii) A balanced weight factor $\omega \cong 0.5$ achieves the best-possible SUs' network performance, particularly lower SU packet loss and higher SU throughput.

5. Enhancement of Exploration Mechanism

Traditionally, during route selection, there are two types of actions, namely, exploitation and exploration. Exploitation selects the best-known route $a_t^i = \text{argmin}_{a \in A} Q_t^i(s_t^i, a)$, which has the lowest cost, in order to improve network performance. Exploration selects a random route $a_t^i \in A$ in order to improve knowledge, specifically, the estimation of Q-values for various routes. Two traditional exploration schemes are ε-greedy and softmax [5], and these schemes have been applied to regulate the exploration probability. A well-balanced tradeoff between exploitation and exploration helps to maximize network performance as time goes by. There have been some limited efforts to investigate this tradeoff in wireless networks; and this investigation applies to routing in CRNs. We present an overview of exploration probability and the traditional (called ε-greedy and softmax) and our proposed (called dynamic softmax) exploration approaches to dynamically regulate the exploration probability, as well as simulation results for the four performance metrics, in the rest of this section.

5.1. An Overview of Exploitation and Exploration. While route exploitation may seem to improve network performance as it forwards SUs' packets using the best-known route, it may cause network congestion and subsequently degrade SUs' network performance. On the other hand, route

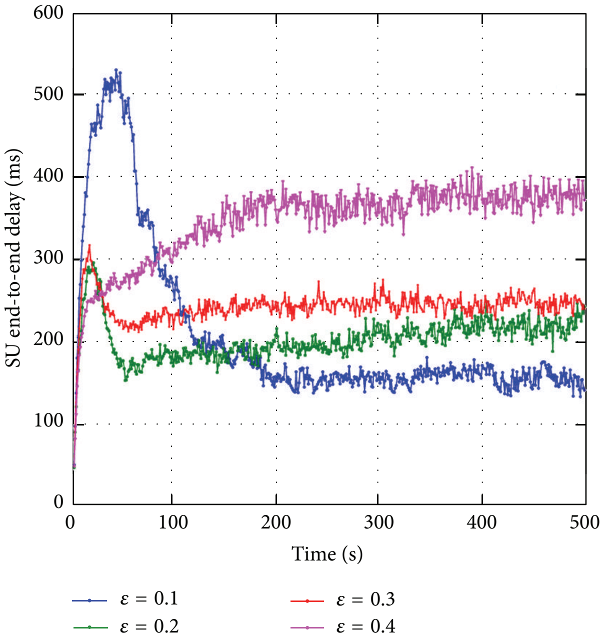


Figure 10: SU end-to-end delay for varying exploration probability ε.

exploration reduces traffic load on the best-known route, and so it may increase the convergence rate to an optimal route in a dynamic and unpredictable operating environment. However, if the exploration probability is high, forwarding the SUs' packets along nonoptimal routes may degrade SUs' network performance causing higher end-to-end delay and packet loss, as well as lower throughput.

We present simulation results of a preliminary investigation to show the effects of exploration probability ε on the convergence rate of SUs' network performance, particularly SU end-to-end delay. Note that more details on simulation setup and parameters are presented in later section. Using ε-greedy, a SU explores with a small probability ε and exploits with probability $1 - \varepsilon$. Convergence rate is the time duration for a SU node to find an optimal or near-optimal route, which provides a stable and enhanced SUs' network performance. Figure 10 shows that, as the exploration probability increases, the convergence rate increases (e.g., from approximately 200 seconds using $\varepsilon = 0.1$ to 100 seconds using $\varepsilon = 0.2$). The average SU end-to-end delay increases with the exploration probability ε. Additionally, higher exploration probability causes instability of SUs' network performance (e.g., SU end-to-end delay has higher fluctuations using $\varepsilon = 0.4$). Also, a peak is observed when exploration probability is low using $\varepsilon = 0.1$ because a SU may exploit nonoptimal routes at most of the time since the Q-values are initialized to 0 values which encourages exploration.

This initial investigation motivates us to achieve a balanced tradeoff between exploitation and exploration in route selection.

5.2. Exploration Approaches. We present two traditional exploration approaches (i.e., ε-greedy and softmax) and a variant of the exploration approach (i.e., dynamic softmax) in this section.

5.2.1. The ε-Greedy Approach. The ε-greedy approach performs exploration with a small probability ε (e.g., $\varepsilon = 0.1$) and exploitation with probability $1 - \varepsilon$ [5]. A drawback is that, during exploration, it selects nonoptimal routes with below-average performance in a random manner with equal probability, and if these routes are chosen most of the times, the SUs' network performance may degrade.

5.2.2. The Softmax Approach. The softmax approach chooses an exploration action based on Q-values [5]. Specifically, routes with lower Q-values (i.e., lower cost) are likely to be explored than routes with higher Q-values (i.e., higher cost), and this addresses the drawback of ε-greedy; specifically, it minimizes the detrimental effects of exploration to SUs' network performance while exploring nonoptimal routes.

Using softmax (or the Boltzmann distribution), SU node i chooses its next-hop SU neighbor node $a_t^i \in A$ with the following probability:

$$P\left(s_t^i, a_t^i\right) = \frac{e^{-Q_t^i(s_t^i, a_t^i)/M}}{\sum_{a \in A^i} e^{-Q_t^i(s_t^i, a)/M}}, \tag{3}$$

where A^i represents a set of SU node i's neighbor nodes and M is the temperature that determines the level of exploration. Higher M value indicates higher possibility of exploration, whereas lower M value indicates higher possibility of exploitation.

5.2.3. The Dynamic Softmax Approach. In ε-greedy and softmax, the exploration probability is predefined; specifically, the exploration probability ε and tempreture M are predefined. There are two shortcomings. Firstly, while the optimal ε and M values are dependent on the dynamicity and unpredictability of the network conditions, it may not be feasible to determine these values on the fly. Secondly, routing in CRNs involves a number of SU nodes, and it may not be feasible to determine the optimal ε and M values for each SU node, each of which may operate in distinctive channels.

We propose a simple and pragmatic exploration approach called dynamic softmax, which is based on the traditional softmax approach. Dynamic softmax regulates the exploration temperature M (see (3)) of a SU based on the dynamicity and unpredictability of the network conditions. In [18], a similar approach to regulate the exploration probability of a nontraditional exploration approach is applied in wireless sensor networks.

In dynamic softmax, temperature M is increased and decreased by a constant factor f based on the network conditions, so that more exploration is performed only when necessary (see Algorithm 1). Higher values of f may increase the convergence rate at the expense of higher fluctuations in SUs' network performance, whereas lower values of f may reduce the convergence rate; however, it achieves lower fluctuations (or higher stability) of SUs' network performance.

```
initialize $M = [M_{\min}, M_{\max}]$
while (updating a new Q-value $Q^i_{t+1}$)
    if $(Q^i_{t+1} > (Q^i_t + \vartheta)$ && $M < M_{\max})$ then $M+ = f$
    else if $(Q^i_{t+1} < (Q^i_t - \vartheta))$ then $M = M$
    else if $((Q^i_t - \vartheta) \le Q^i_{t+1} \le (Q^i_t + \vartheta)$ && $M > M_{\min})$ then $M- = f$
    end if
end
```

Algorithm 1: Dynamic softmax algorithm at SU node *i*.

Table 5: Simulation parameters and values for investigating the exploration approaches.

Category	Parameter	Value
SU	Traditional ε-greedy exploration probability, ε	{0.07, 0.14}
	Traditional softmax exploration temperature, M	{0.04, 0.05}
	Initial dynamic softmax temperature, M	0.05
	Dynamic softmax adjustment factor, f	0.01
	Dynamic softmax temperature range, $[M_{\min}, M_{\max}]$	[0.01, 0.1]
	Dynamic softmax Q-value threshold, ϑ	0.1
PU	Standard deviation of PUL, σ_{PUL}	{0.2, 0.8}
Channel	Mean PER, μ_{PER}	0
	Standard deviation of PER, σ_{PER}	0

The adjustment of temperature M is based on the trend of the Q-value of a route as follows.

(i) When the Q-value increases (i.e., higher routing cost), temperature M is increased in order to encourage exploration of other routes as this may indicate the emergence of PUs' activities which have degraded the SUs' network performance.

(ii) When the Q-value decreases (i.e., lower routing cost), temperature M is left unchanged as this may indicate a forthcoming convergence to an optimal route which provides greater stability to SUs' network performance.

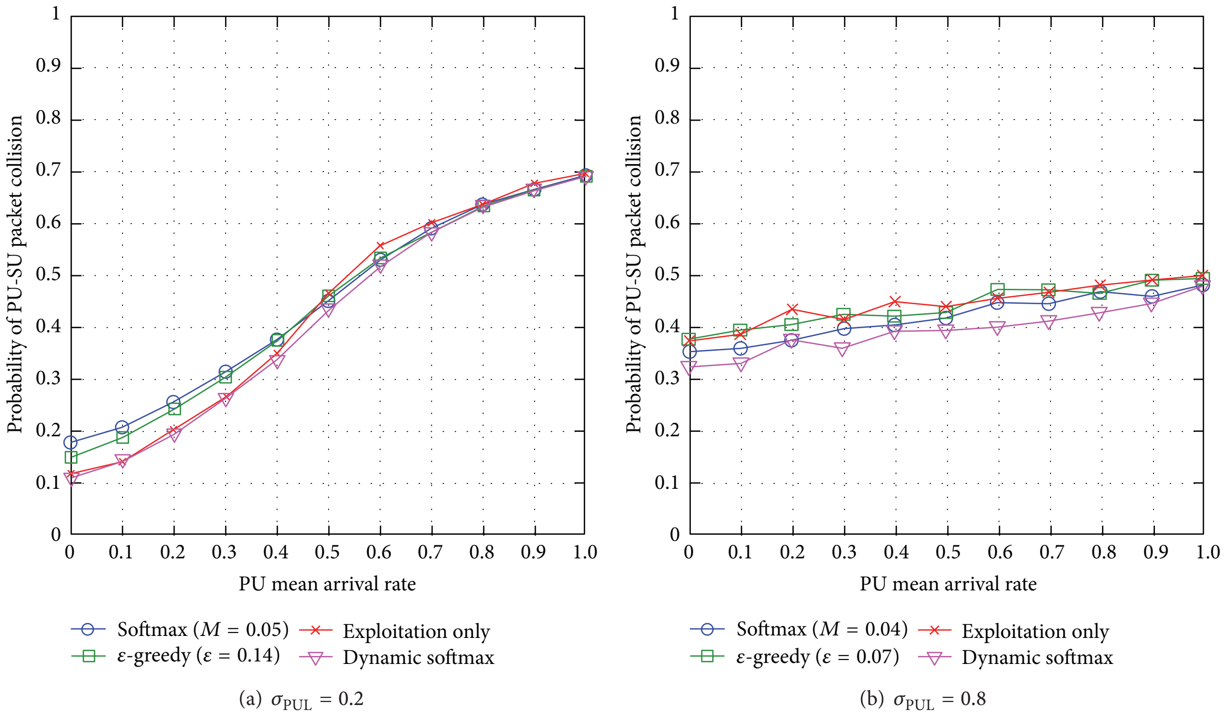
(iii) When the change of Q-value is less than a threshold ϑ, the temperature M is decreased in order to encourage exploitation as this indicates that there has been convergence to an optimal route which provides a stable network performance.

5.3. Comparison of ε-Greedy, Softmax, Dynamic Softmax, and Exploitation-Only Approaches. This section investigates the effects of various exploration approaches applied to WCRQ-routing on network performance. We compare the network performance achieved by WCRQ-routing using two traditional exploration approaches, namely, ε-greedy and softmax, an exploitation-only approach and a variant of the exploration approach which we propose, namely dynamic softmax. The exploitation-only approach exploits at all times and does not explore.

While Table 3 presents the default simulation parameters and values, Table 5 presents the specific simulation parameters and values for this investigation. Generally speaking, we simplify the simulation values in order to focus on the effects of various exploration approaches on network performance in the presence of the dynamicity of the channel availability with respect to PUL, so we assume a noiseless channel with $\mu_{\text{PER}} = 0$ and $\sigma_{\text{PER}} = 0$. The simulation results are shown for two conditions, in which the standard deviation of PUL $\sigma_{\text{PUL}} \in \{0.2, 0.8\}$ indicates low and high levels of unpredictability of the channel availability, respectively. The exploration metrics of the traditional approaches are $\varepsilon \in \{0.07, 0.14\}$ and temperature $M \in \{0.04, 0.05\}$, and these are chosen empirically; specifically, these are the average optimal values estimated by running extensive simulations using different values of exploration metrics under different levels of channel unpredictability σ_{PUL}. Hence, we compare the dynamic softmax approach with the best possible network performance achieved by ε-greedy and softmax. For dynamic softmax (see Algorithm 1), the temperature range is $[M_{\min}, M_{\max}] = [0.01, 0.1]$ because network performance degrades significantly when $M > 0.1$. The initial temperature is set to an average value of $M = 0.05$. Finally, the adjustment factor is set to $f = 0.01$ and the Q-value threshold is set to $\vartheta = 0.1$, and these values are chosen empirically.

We present simulation results for the four performance metrics in the rest of this section.

5.3.1. SUs' Interference to PUs. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0.2$, most next-hop node (or link) and channel pairs have very similar PU mean arrival rate μ_{PUL}; however, dynamic softmax outperforms the traditional exploration approaches, specifically up to 39% compared to softmax and up to 22% compared to ε-greedy while achieving very similar network performance with the exploitation-only



(a) $\sigma_{\text{PUL}} = 0.2$

(b) $\sigma_{\text{PUL}} = 0.8$

Figure 11: SUs' interference to PUs for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

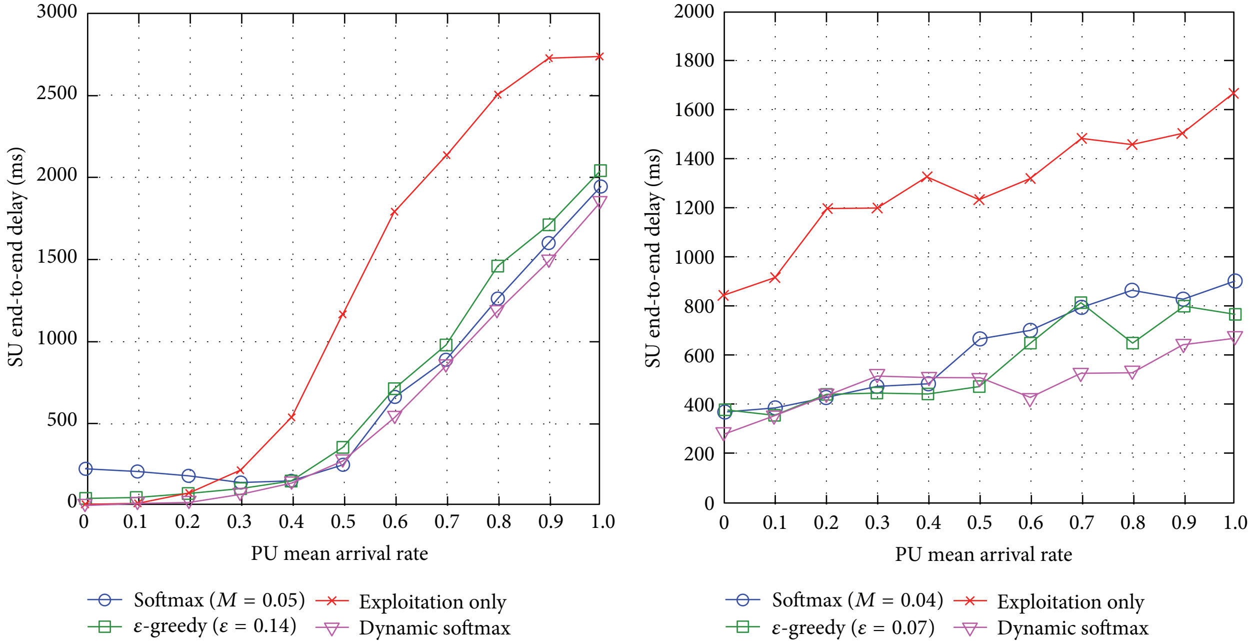
approach (see Figure 11(a)). When the PU mean arrival rate μ_{PUL} becomes higher, all channels have high levels of PUs' activities and Q-values among the channels do not generally vary. So, there is lack of exploration and all approaches achieve very similar network performance. When $\sigma_{\text{PUL}} = 0.8$, the link and channel pairs have the greatest difference in the levels of PU mean arrival rate μ_{PUL}. Dynamic softmax outperforms the other exploration approaches, and the exploitation-only approach causes the highest SUs' interference to PUs (see Figure 11(b)). When the standard deviation of PUL σ_{PUL} becomes higher, all channels have different levels of PUs' activities, and Q-values among the channels generally vary. So, more explorations are necessary explaining why the exploitation-only approach causes the worst network performance with the highest SUs' interference to PUs. In general, softmax outperforms ε-greedy in most cases because softmax explores lesser below-average routes compared to ε-greedy.

Generally speaking, dynamic softmax achieves similar or better network performance in terms of SUs' interference to PUs compared to the traditional exploration approaches with optimal exploration metrics. This is because dynamic softmax learns the optimal exploration temperature dynamically based on the dynamicity and unpredictability levels of the network conditions. Additionally, the SUs' interference to PUs increases with the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

5.3.2. SU End-to-End Delay. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0.2$, most link and channel pairs have very similar PU mean arrival rate μ_{PUL}. Dynamic softmax outperforms the traditional exploration approaches, and up to 52% compared to exploitation-only which incurs the highest end-to-end delay (see Figure 12(a)). The exploitation-only approach exploits the best-possible route at all times and there is lack of load balancing among routes causing network congestion and increased SU end-to-end delay. Note that when there are low levels of PU mean arrival rate $\mu_{\text{PUL}} = [0.0, 0.4]$, softmax incurs the highest end-to-end delay because it performs the most unnecessary explorations. When $\sigma_{\text{PUL}} = 0.8$, the link and channel pairs have great difference in the levels of PU mean arrival rate μ_{PUL}. Dynamic softmax outperforms the other exploration approaches (see Figure 12(b)). However, when there are low levels of PU mean arrival rate $\mu_{\text{PUL}} = [0.0, 0.5]$, dynamic softmax incurs higher SU end-to-end delay because it performs more unnecessary explorations as a result of greater variations in Q-values, while ε-greedy explores routes randomly which increases load-balancing, and softmax chooses more above-average routes which increases network congestion.

Generally speaking, dynamic softmax reduces SU end-to-end delay compared to the traditional exploration approaches with optimal exploration metrics. Additionally, the SUs' interference to PUs increases with the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

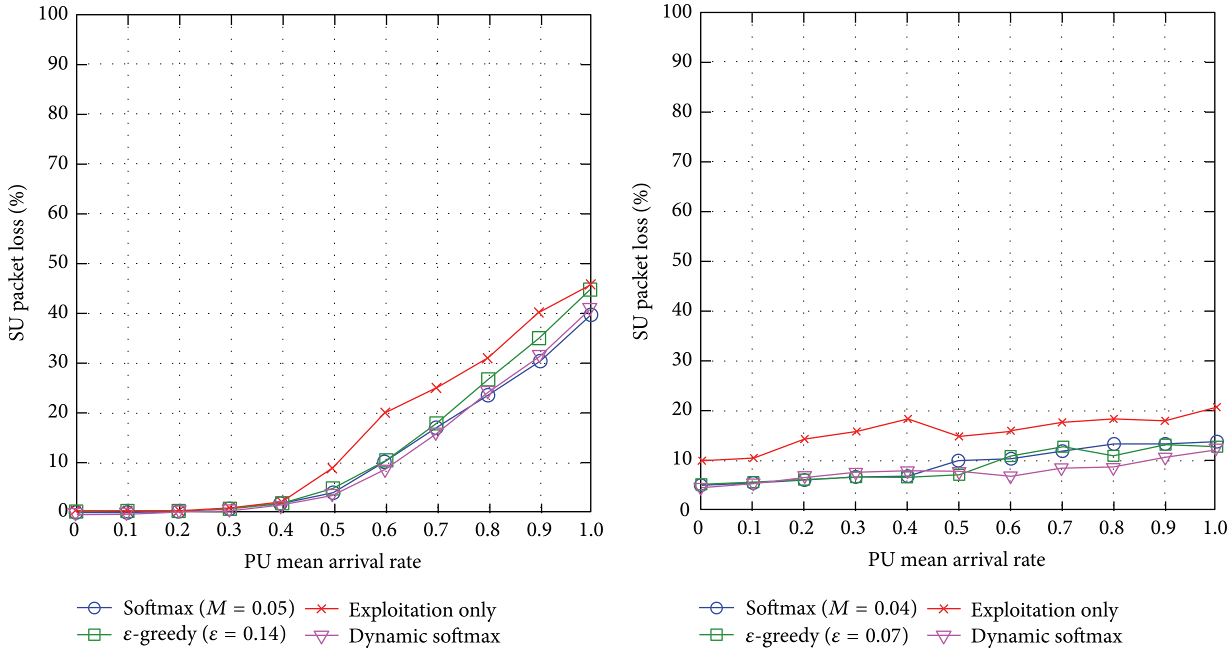
5.3.3. SU Packet Loss. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0.2$, dynamic softmax either achieves similar network performance or outperforms the traditional exploration approaches, specifically up to 57% compared to the exploitation-only approach which incurs the highest packet loss (see Figure 13(a)). The exploitation-only approach



(a) $\sigma_{\mathrm{PUL}} = 0.2$

(b) $\sigma_{\mathrm{PUL}} = 0.8$

FIGURE 12: SU end-to-end delay for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

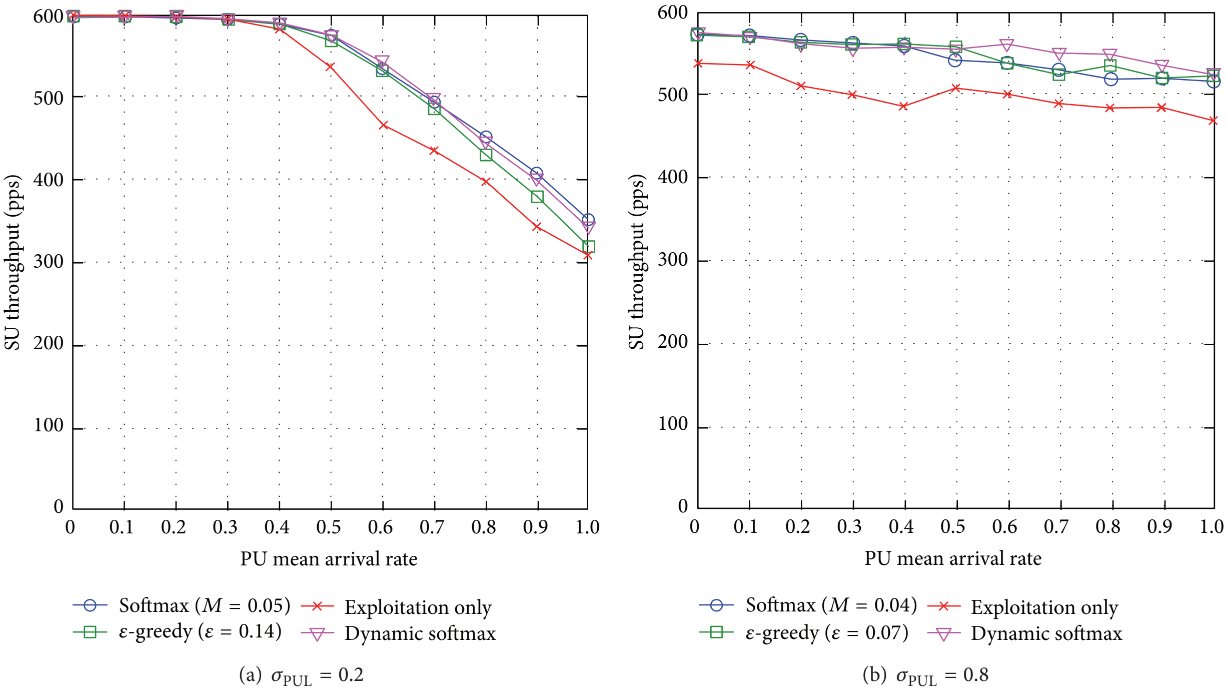


(a) $\sigma_{\mathrm{PUL}} = 0.2$

(b) $\sigma_{\mathrm{PUL}} = 0.8$

FIGURE 13: SU packet loss for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

exploits the best-possible route at all times and there is lack of load balancing among routes causing network congestion and increased SU packet loss. When $\sigma_{\mathrm{PUL}} = 0.8$, dynamic softmax outperforms the other exploration approaches (see Figure 13(b)). The ε-greedy approach explores routes randomly which increases load-balancing, and so it outperforms softmax at times, while softmax chooses more above-average routes causing greater network congestions.



(a) σ_{PUL} = 0.2

(b) σ_{PUL} = 0.8

FIGURE 14: SU throughput for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

Generally speaking, dynamic softmax reduces SU packet loss compared to the traditional exploration approaches with optimal exploration metrics. Additionally, the SU packet loss increases with the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

5.3.4. SU Throughput. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0.2$, dynamic softmax either achieves similar network performance or outperforms the traditional exploration approaches, specifically, up to 17% compared to the exploitation-only approach which incurs the highest packet loss (see Figure 14(a)). The exploitation-only approach exploits the best-possible route at all times and there is lack of load balancing among routes causing network congestion and reduced SU throughput. When $\sigma_{\text{PUL}} = 0.8$, similar trends are observed in network scenario of $\sigma_{\text{PUL}} = 0.8$ in the investigation of SU end-to-end delay (see Section 5.3.3).

Generally speaking, dynamic softmax increases SU throughput compared to the traditional exploration approaches with optimal exploration metrics. Additionally, the SU throughput reduces with the PU mean arrival rate μ_{PUL} and the standard deviation of PUL σ_{PUL}.

5.3.5. Section Summary. We summarize simulation outcomes for the comparison of ε-greedy, softmax, dynamic softmax, and exploitation-only approaches as follows.

(i) Network performance degrades as the dynamicity of PUs' activities μ_{PUL} increases.

(ii) Exploration approaches (i.e., ε-greedy, softmax, and dynamic softmax) achieve load-balancing among the available routes, and hence they improve SUs' (and PUs' in some cases) network performance compared to the exploitation-only approach.

(iii) Traditional exploration approaches, namely ε-greedy and softmax, outperform each other under different network conditions. For instance, when the unpredictability of the channel availability σ_{PUL} is low, softmax achieves better SUs' network performance than ε-greedy; however, when the unpredictability level of the channel availability σ_{PUL} becomes higher, ε-greedy achieves better SUs' network performance than softmax due to greater load balancing among routes, so there is lesser network congestion. Hence, neither ε-greedy nor softmax achieves the best-possible network performance under all network conditions.

(iv) WCRQ-routing and CRQ-routing minimize SUs' interference to PUs with respect to PUL in the presence of dynamicity and unpredictability of the channel availability.

(v) Dynamic softmax achieves the best-possible PUs' and SUs' network performances in most cases compared to other approaches because it learns the best-possible exploration temperature dynamically based on the dynamicity and unpredictability levels of the channel availability.

6. Enhancement of Learning Rate Adjustment

The learning rate α, which is used to regulate the speed of convergence to the optimal or near-optimal action, is another important parameter. A suitable learning rate α

value is essential. There have been some limited efforts to investigate the learning rate α in the domain of routing in wireless networks [19], and this investigation applies to routing in CRNs. We present an overview of learning rate, the traditional (called the Win-or-Learn-Fast Policy Hill Climbing approach or win-lose [6]) and our proposed (called the counterapproach) learning rate adjustment approaches to dynamically regulate the learning rate, as well as the simulation results for the four performance metrics, in the rest of this section.

6.1. An Overview of Learning Rate. Higher learning rate $0 \leq \alpha \leq 1$ indicates higher speed of learning and convergence rate, and it is more responsive to the dynamicity of the operating environment. Higher learning rate may cause fluctuation in Q-value because the Q-value is now more dependent on its recent estimates, which may be unstable, rather than its previous experience. On the other hand, lower learning rate may cause very low convergence rate because the Q-value is now more dependent on its previous experience rather than its recent estimates.

We present simulation results of a preliminary investigation to show the effects of learning rate α on the SUs' network performance, particularly SU end-to-end delay. Note that more details on simulation setup and parameters are presented in later section. Figure 15 shows that neither the lowest (i.e., $\alpha = 0.0001$) nor the highest (i.e., $\alpha = 1$) learning rate achieves the lowest SU end-to-end delay, and learning rate $\alpha = 0.0016$ achieves the lowest SU end-to-end delay. Hence, too low learning rate causes a SU to learn about the available routes slowly and so it does not adapt well to the changes in the PUs' mean arrival rate, while too high a learning rate causes a SU to learn very fast and so it changes its route more frequently causing the SU end-to-end delay to increase.

Hence, a suitable learning rate α value is essential to provide network performance enhancement. Traditionally, the learning rate is a predefined and fixed value which is not adaptive to the dynamicity and unpredictability levels of the operating environment, and so it does not provide the best network performance in a dynamic and unpredictable operating environment.

This initial investigation motivates us to search for a suitable learning rate.

6.2. Learning Rate Adjustment Approaches. We present a traditional exploration approach (i.e., the win-lose approach) and a variant of the exploration approach (i.e., the counterapproach), as well as a baseline approach (i.e., the random approach), in this section.

6.2.1. The Win-Lose Approach. In [20], a stochastic learning algorithm, namely, win-or-learn-fast policy hill climbing (or win-lose for simplicity) [6] is applied to regulate the learning rate dynamically based on the dynamicity of the operating environment in a wireless network. Given that higher Q-values provide network performance enhancement, the algorithm defines *winning* and *losing* as receiving higher and lower Q-values than its current Q-value, which is its expectation, respectively. When the algorithm is winning, the learning rate is set to a lower value, and vice-versa. The reason is that when a SU wins, it must be cautious in changing its routing policy and more time must be given to other SUs to adjust their own policies in favor of this winning. On the other hand, when a SU node loses, it must adapt faster to changes in the operating environment because its network performance (or rewards) is lower than its expectation.

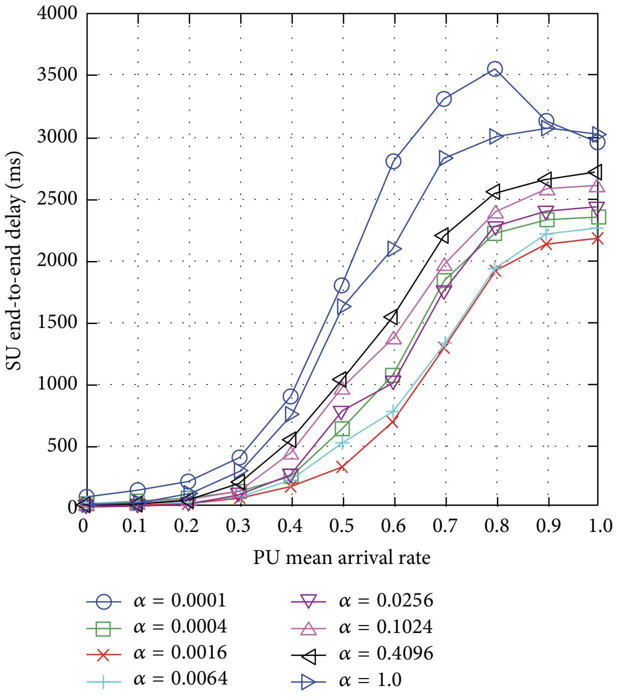


Figure 15: SU end-to-end delay for varying learning rate α.

6.2.2. The Counterapproach. The traditional win-lose approach (see Section 6.2.1) regulates the learning rate for each update of Q-value. Nevertheless, in a highly dynamic and unpredictable operating environment, Q-values may vary greatly and this causes frequent changes to learning rate. As a consequence, there are frequent changes to routing decision causing higher fluctuation in SUs' network performance.

The proposed counterapproach addresses the aforementioned issue by regulating the learning rate based on the historical Q-values. Specifically, for each Q-value (or state-action pair), it keeps track of a counter for winning c_t^p and another counter for losing c_t^n (see Algorithm 2). Subsequently, it calculates a ratio $o_t^n = c_t^p / c_t^n$, which represents the number of winning to the number of losing for a Q-value. When the ratio is $o_t^n > 1$, the learning rate is set to a lower value because this indicates a winning event, while the learning rate is set to a higher value when the ratio is $o_t^n < 1$ because this indicates a losing event, and when the ratio is $o_t^n = 1$, the learning rate is unchanged as it indicates a stable network performance. Similarly, when the newly received Q-value is similar to the previous value, the counter is not

```
initialize $\alpha c^{p}_{t+1} + + = [\alpha_{\min}, \alpha_{\max}]$
while (updating a new $Q$-value $Q^{i}_{t+1}$)
    if ($Q^{i}_{t+1} < Q^{i}_{t}$) then $c^{p}_{t+1} + +$
    else $c^{n}_{t+1} + +$
    end if
    compute $o^{n}_{t+1} = c^{p}_{t+1}/c^{n}_{t+1}$
    if ($o^{n}_{t+1} < 1 \,\&\&\, \alpha < \alpha_{\max}$) then $\alpha + = f$
    else if ($o^{n}_{t+1} > 1 \,\&\&\, \alpha > \alpha_{\min}$) then $\alpha - = f$
    end if
end
```

Algorithm 2: Counterapproach algorithm at SU node i.

updated. Note that we assume the learning rate α is increased and decreased by a small constant factor f in order to avoid fluctuations in SUs' network performance.

6.2.3. The Random Approach. The random approach, which does not apply any learning mechanism, regulates the learning rate in a round robin fashion without considering the dynamicity and unpredictability of the operating environment. This scheme serves as a baseline for comparison with win-lose and the counterapproach, and more importantly, it is used to show the effects of nonoptimal learning rates on network performance. This approach regulates the learning rate whenever it loses. Specifically, an agent decreases its learning rate α until the minimum learning rate limit $\alpha_{\min}$ is reached, and then it increases the learning rate α again until the maximum learning rate limit $\alpha_{\max}$ is reached, and this continues to loop in between the minimum $\alpha_{\min}$ and the maximum $\alpha_{\max}$ limits.

6.3. Comparison of Win-Lose and Counter and Random Approaches. This section investigates the effects of various learning rate adjustment approaches applied to WCRQ-routing on network performance. We compare the network performance achieved by WCRQ-routing using a traditional learning rate adjustment approach (i.e., win-lose), a variant of the win-lose (i.e., the counterapproach), a baseline (i.e., the random approach), and another baseline that provides the best empirical learning approach (which we may conveniently call *best*). The best empirical approach provides the best possible network performance, and the learning rate α is obtained by running extensive simulations under different levels of channel unpredictability.

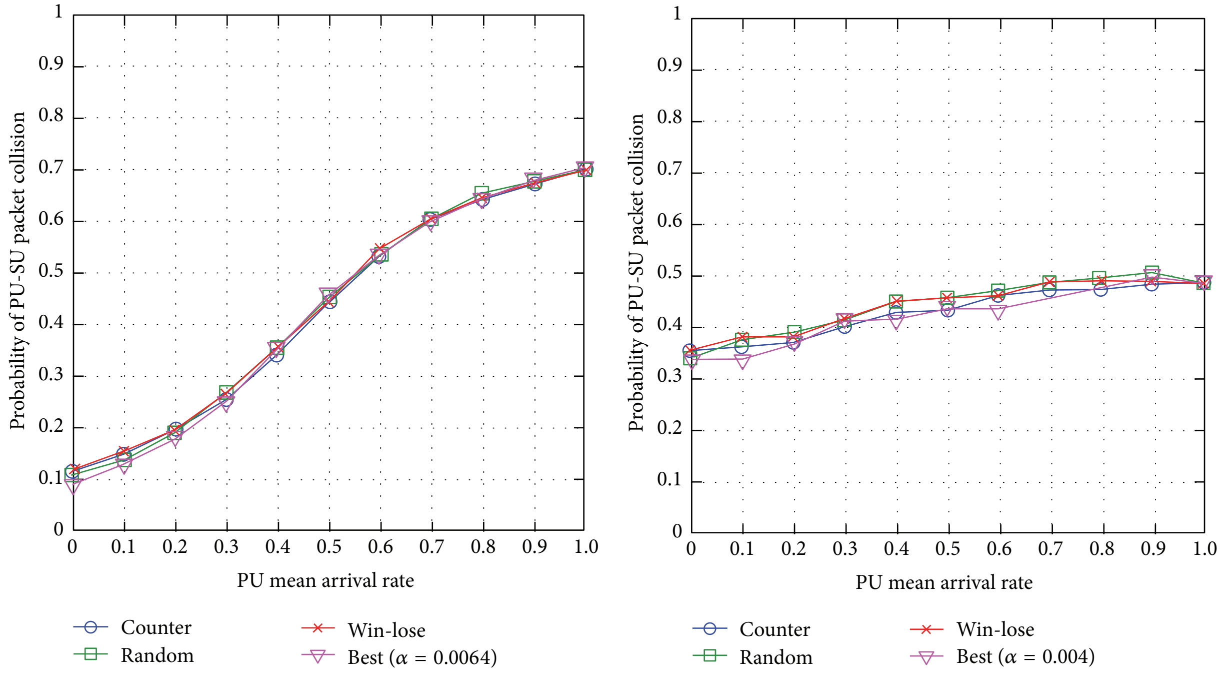
Table 6: Simulation parameters and values for investigating the learning rate adjustment approaches.

Category	Parameter	Value
SU	Best empirical learning rate, α	{0.0064, 0.004}
	Learning rate adjustment factor, f	0.001
	Learning rate range, $[\alpha_{\min}, \alpha_{\max}]$	[0.001, 0.1]
PU	Standard deviation of PUL, σ_{PUL}	{0.2, 0.8}
Channel	Mean PER, μ_{PER}	0
	Standard deviation of PER, σ_{PER}	0

While Table 3 presents the default simulation parameters and values, Table 6 presents the specific simulation parameters and values for this investigation. Generally speaking, we simplify the simulation values in order to focus on the effects of various learning rate adjustment approaches on network performance in the presence of the dynamicity of the channel availability with respect to PUL, so we assume a noiseless channel with $\mu_{\text{PER}} = 0$ and $\sigma_{\text{PER}} = 0$. The simulation results are shown for two conditions, in which the standard deviation of PUL $\sigma_{\text{PUL}} \in \{0.2, 0.8\}$ indicates low and high levels of the unpredictability of channel availability, respectively. The best empirical learning rate is set to $\alpha \in \{0.0064, 0.004\}$, which is estimated by running extensive simulations. The range of the learning rate is $[\alpha_{\min}, \alpha_{\max}] = [0.001, 0.1]$ because network performance degrades significantly when $\alpha > 0.1$. Finally, the learning rate adjustment factor is $f = 0.001$, which is estimated by running extensive simulations with the objective of enhancing SUs' network performance.

We present simulation results for the four performance metrics in the rest of this section.

6.3.1. SUs' Interference to PUs. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0.2$, most next-hop node (or link) and channel pairs have very similar PU mean arrival rate μ_{PUL}. The effect of learning rate on network performance is minimal and so all the learning rate adjustment approaches achieve almost similar network performance (see Figure 16(a)). When $\sigma_{\text{PUL}} = 0.8$, the link and channel pairs have great difference in the levels of PU mean arrival rate μ_{PUL}, and the Q-values among the channels generally vary greatly. The counterapproach achieves almost similar SUs' interference to PUs compared to the best empirical approach and the lowest interference level compared to the win-lose and random approaches (see Figure 16(b)). This is because when σ_{PUL} is high, the counterapproach chooses a suitable learning rate that reduces fluctuations in Q-values while making routing decisions, while the win-lose and random approaches adjust the SU learning rate very fast resulting



(a) $\sigma_{\text{PUL}} = 0.2$

(b) $\sigma_{\text{PUL}} = 0.8$

Figure 16: SUs' interference to PUs for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

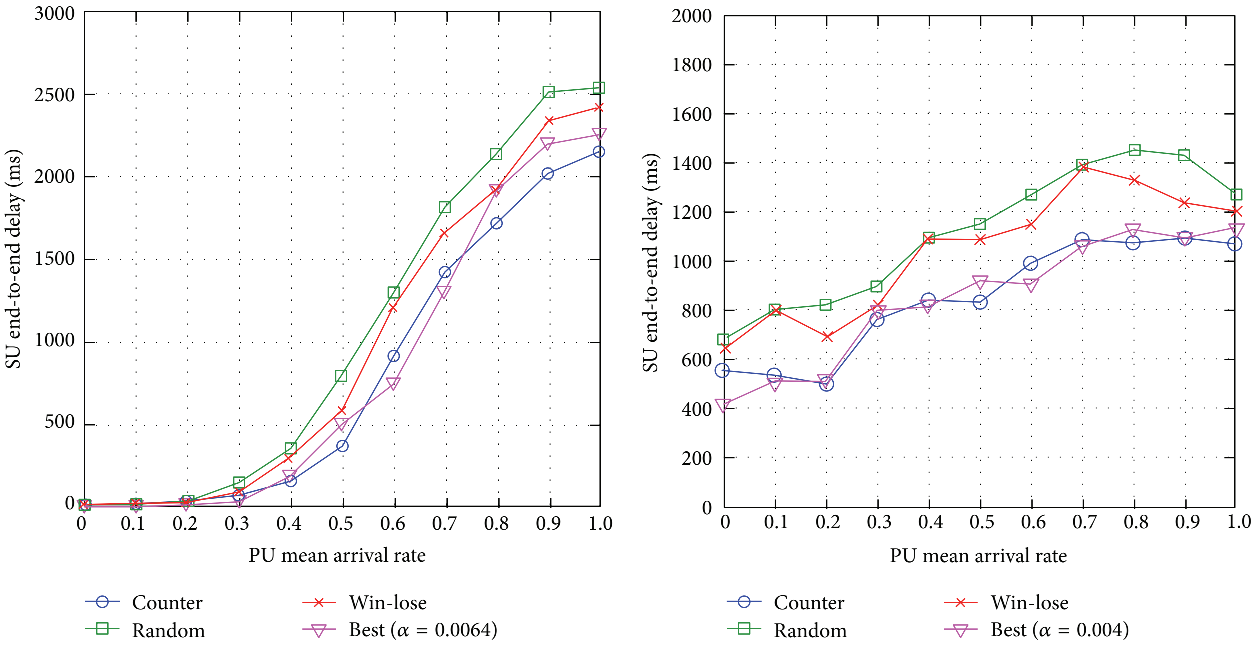
in more frequent changes to SU route decision, and hence higher SUs' interference to PUs.

Generally speaking, the counterapproach reduces SUs' interference to PUs. Additionally, the SUs' interference to PUs increases with the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

6.3.2. SU End-to-End Delay. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0.2$, most link and channel pairs have very similar PU mean arrival rate μ_{PUL}. The counterapproach achieves almost similar SU end-to-end delay compared to the best empirical approach and the lowest SU end-to-end delay compared to the win-lose and random approaches, specifically up to 11% compared to win-lose which incurs the highest packet loss (see Figure 17(a)). The random approach has the highest SU end-to-end delay, which is up to 12% higher compared to the best empirical learning rate approach, and this shows the effects of non-optimal learning rate on SU end-to-end delay. When $\sigma_{\text{PUL}} = 0.8$, the link and channel pairs have great difference in the levels of PU mean arrival rate μ_{PUL}, and the Q-values among the channels generally vary greatly. So, the learning rate has greater effects on SU end-to-end delay as a SU needs a suitable learning rate to learn about the available routes while minimizing fluctuations in Q-values. Similarly, the counterapproach achieves almost similar SU end-to-end delay to PUs compared to the best empirical approach, and the lowest SU end-to-end delay compared to the win-lose and random approaches, specifically up to 21% lower compared to win-lose (see Figure 17(b)). This is because when σ_{PUL} is high, the counterapproach chooses a suitable learning rate that reduces fluctuations in Q-values while making routing decisions, while the win-lose and random approaches adjust the SU learning rate very fast resulting in more frequent changes to SU route decision, and hence higher SU end-to-end delay. When PU mean arrival rate is $\mu_{\text{PUL}} = 1$, the win-lose and random approaches reduce SU end-to-end delay since faster adjustment of learning helps to achieve load balancing among the available routes.

Generally speaking, the counterapproach reduces SU end-to-end delay. Additionally, the SU end-to-end delay increases with the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

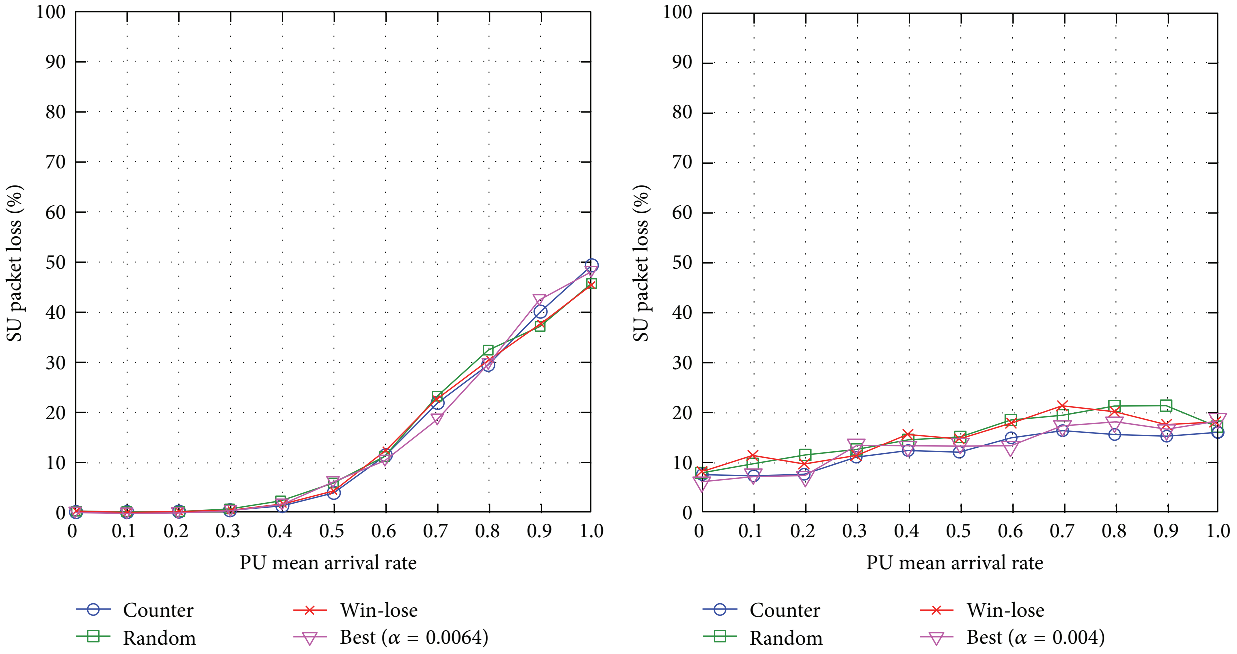
6.3.3. SU Packet Loss. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0.2$, the effect of learning rate is minimal and so all the learning rate adjustment approaches achieve almost similar network performance. When the PU mean arrival rate is $\mu_{\text{PUL}} = [0.8, 1.0]$, the win-lose and random approaches achieve lower packet loss compared to other approaches, specifically up to 8% lower compared to the counterapproach (see Figure 18(a)). This is because the win-lose and random approaches adjust the SU learning rate faster than the counterapproach leading to more frequent changes to routes, and as the routes become more congested, this helps to achieve load balancing among the available routes and reduces SU packet loss. When $\sigma_{\text{PUL}} = 0.8$, the learning rate has slightly greater effects on SU packet loss as a SU needs a suitable learning rate to learn about the available routes while minimizing fluctuations in Q-values. Similarly, the counterapproach achieves almost similar SU



(a) $\sigma_{\text{PUL}} = 0.2$

(b) $\sigma_{\text{PUL}} = 0.8$

FIGURE 17: SU end-to-end delay for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.



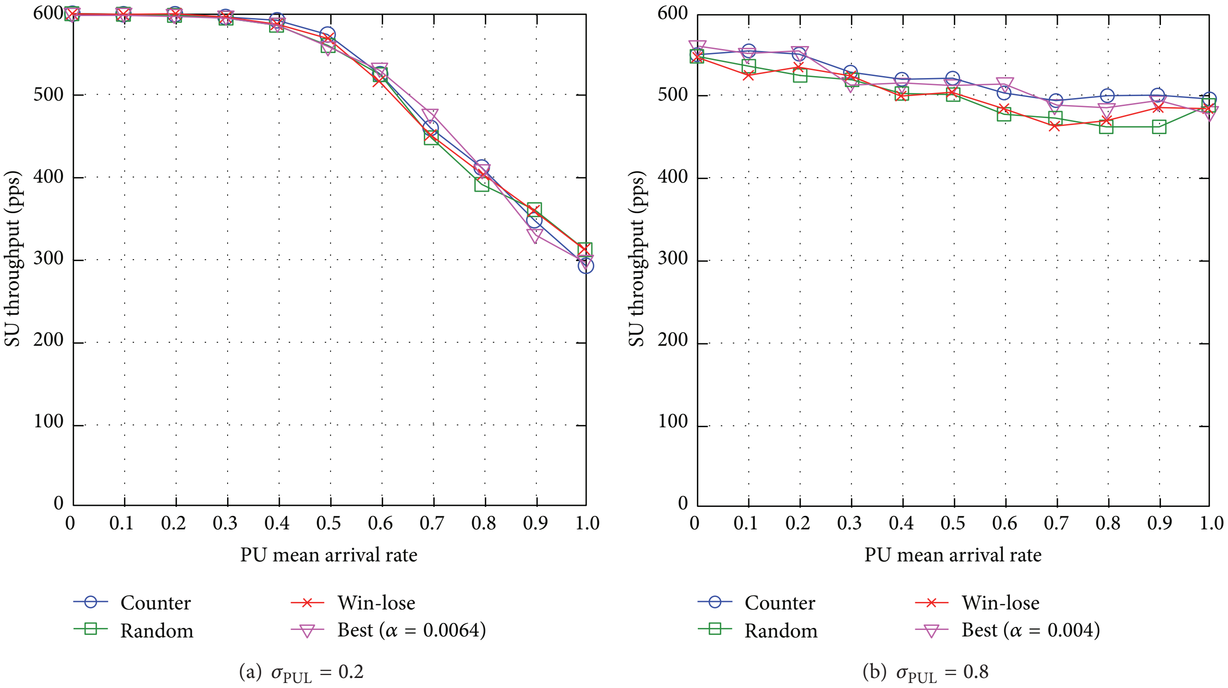
(a) $\sigma_{\text{PUL}} = 0.2$

(b) $\sigma_{\text{PUL}} = 0.8$

FIGURE 18: SU packet loss for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

packet loss compared to the best empirical approach, and the lowest SU packet loss compared to the win-lose and random approaches, specifically up to 23% lower compared to win-lose (see Figure 18(b)). This is because when σ_{PUL} is high, the counterapproach chooses a suitable learning rate that reduces fluctuations in Q-values while making routing decisions.

Generally speaking, the counterapproach reduces SU packet loss. Additionally, the SU packet loss increases with



(a) $\sigma_{\text{PUL}} = 0.2$

(b) $\sigma_{\text{PUL}} = 0.8$

FIGURE 19: SU throughput for varying PU mean arrival rate μ_{PUL} for different levels of standard deviation of PUL σ_{PUL}.

the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

6.3.4. SU Throughput. When the standard deviation of PUL is low $\sigma_{\text{PUL}} = 0.2$, the effect of learning rate is minimal and so all the learning rate adjustment approaches achieve almost similar network performance (see Figure 19(a)), and the reasons are similar to those observed in the investigation of SU packet loss (see Section 6.3.3). When $\sigma_{\text{PUL}} = 0.8$, the counterapproach achieves almost similar SU throughput compared to the best empirical approach, and the highest SU throughput compared to the win-lose and random approaches, specifically up to 7% higher compared to win-lose (see Figure 19(b)), and the reasons are similar to those observed in the investigation of SU packet loss (see Section 6.3.3).

Generally speaking, the SU throughput decreases with the PU mean arrival rate μ_{PUL} and reduces with the standard deviation of PUL σ_{PUL}.

6.3.5. Section Summary. We summarize simulation outcomes for the comparison of win-lose, counter and random approaches as follows.

(i) Network performance degrades as the dynamicity of PUs' activities μ_{PUL} increases.

(ii) The effects of learning rate on SUs' network performance increase with the unpredictability of the PUs' activities σ_{PUL}.

(iii) When the PU mean arrival rate is high (i.e., $\mu_{\text{PUL}} = [0.8, 1.0]$), where route congestion is higher, the win-lose and random approaches improve SUs' network performance. This is because these approaches regulate the learning rate at higher speed, and this helps to achieve load balancing among the available routes, and subsequently enhances SUs' network performance.

(iv) The counterapproach achieves almost similar SU network performance to the best empirical learning rate approach and outperforms the win-lose and random approaches in most cases, particularly when the unpredictability of the PUs' activities σ_{PUL} becomes higher. This is because the counterapproach chooses a suitable learning rate to reduce fluctuations in Q-values which start to vary at higher σ_{PUL} values.

7. Conclusions

Through simulation, this paper investigates the effects of reinforcement learning (RL) parameters on network performance for routing scheme in the presence of the dynamicity and unpredictability of the channel availability in cognitive radio ad hoc networks. We present WCRQ-routing that incorporates a weight factor ω in the reward representation to adjust the tradeoff between PUs' and SUs' network performances, as well as to further improve the overall network performance of SUs. Higher weight factor ω increases PUs' network performance, while lower ω increases SUs' network performance, and so a balanced value of ω helps to achieve the best SUs' network performance, particularly lower packet loss and higher throughput. The SUs' network performance can be further enhanced by regulating exploration probability and learning rate. We present a simple and pragmatic exploration approach called *dynamic softmax* to regulate the exploration temperature. Dynamic softmax learns a near-optimal exploration temperature dynamically according to the dynamicity and unpredictability of the channel availability, and it achieves better network performance in most cases compared to the traditional exploration approaches, namely ε-greedy and softmax. We present a simple and pragmatic learning rate adjustment approach called the *control* approach to regulate the learning rate dynamically based on the historical Q-values. The counterapproach achieves better SUs' network performance compared to the traditional win-lose approach, and it achieves almost similar SUs' network performance to the best empirical learning rate approach by learning a near-optimal learning rate dynamically based on the dynamicity and unpredictability of the operating environment.

Conflict of Interests

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

Acknowledgment

This work was supported by the Malaysian Ministry of Science, Technology and Innovation (MOSTI) under Science Fund 01-02-16-SF0027.

References

[1] H. A. A. Al-Rawi and K.-L. A. Yau, "Routing in distributed cognitive radio networks: a survey," *Wireless Personal Communications*, vol. 69, no. 4, pp. 1983–2020, 2013.

[2] H. A. A. Al-Rawi, M. A. Ng, and K.-L. A. Yau, " Application of reinforcement learning to routing in distributed wireless netowrks: a review ," *Artificial Intelligence Review*, 2013.

[3] P. Derakhshan-Barjoei, G. Dadashzadeh, F. Razzazi, and S. M. Razavizadeh, "Power and time slot allocation in cogitive relay networks using particle swarm optimization," *The Scientific World Journal*, vol. 2013, Article ID 424162, 9 pages, 2013.

[4] J. Zhao and J. Yuan, "An improved centralized cognitive radio network spectrum allocation algorithm based on the allocation sequence," *nternational Journal of Distributed Sensor Networks*, vol. 2013, Article ID 875342, 13 pages, 2013.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, Mass, USA, 1998.

[6] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artificial Intelligence*, vol. 136, no. 2, pp. 215–250, 2002.

[7] Q. Guan, F. R. Yu, S. Jiang, and G. Wei, "Prediction-based topology control and routing in cognitive radio mobile ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4443–4452, 2010.

[8] K. R. Chowdhury and I. F. Akyildiz, "CRP: a routing protocol for cognitive radio ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 794–804, 2011.

[9] Q. Zhu, Z. Yuan, J. B. Song, Z. Han, and T. Başar, "Dynamic interference minimization routing game for on-demand cognitive pilot channel," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, pp. 1–6, Miami, Fla, USA, December 2010.

[10] M. Xie, W. Zhang, and K. Wong, "A geometric approach to improve spectrum efficiency for cognitive relay networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 268–281, 2010.

[11] B. Xia, M. H. Wahab, Y. Yang, Z. Fan, and M. Sooriyabandara, "Reinforcement learning based spectrum-aware routing in multi-hop cognitive radio networks," in *Proceedings of the 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM '09)*, pp. 1–5, Hannover, Germany, June 2009.

[12] D. Oužecki and D. Jevtić, "Reinforcement learning as adaptive network routing of mobile agents," in *Proceedings of the 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO '10)*, pp. 479–484, Opatija, Croatia, May 2010.

[13] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, "Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, vol. 35, no. 3, pp. 360–372, 2005.

[14] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: a reinforcement learning approach," *Advances in Neural Infomation Processing Systems*, vol. 6, pp. 671–678, 1994.

[15] Y. Zhang and M. Fromherz, "Constrained flooding: a robust and efficient routing framework for wireless sensor networks,," in *Proceedings of 20th IEEE International Conference on Advanced Information Networking and Applications (AINA '20)*, IEEE, Vienna, Austria, 2006.

[16] H. A. Al-Rawi, K.-L. A. Yau, H. Mohamad, N. Ramli, and W. Hashim, "A reinforcement learning–based routing scheme for cognitive radio ad hoc networks," in *Proceedings of the 7th IFIP Wireless and Mobile Networking Conference (WMNC '14)*, May 2014.

[17] G. E. Box and M. E. Mullerm, "A note on the generation of random normal deviates," *The Annals of Mathematical Statistics (Institute of Mathematical Statistics)*, vol. 29, no. 2, pp. 610–611, 1958.

[18] A. Förster and A. L. Murphys, "FROMS: feedback routing for optimizing multiple sinks in WSN with reinforcement learning," in *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '07)*, pp. 371–376, IEEE, Melbourne, Australia, December 2007.

[19] A. McAuley, K. Sinkar, L. Kant, C. Graff, and M. Patel, "Tuning of reinforcement learning parameters applied to OLSR using a cognitive network design tool," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '12)*, pp. 2786–2791, IEEE, Shanghai, China, April 2012.

[20] P. Nurmi, "Reinforcement learning for routing in ad hoc networks," in *Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt '07)*, pp. 1–8, Limassol, Cyprus, April 2007.