

Research Article

Reinforcement Learning-Based Routing Protocol to Minimize Channel Switching and Interference for Cognitive Radio Networks

Tauqeer Safdar Malik ¹ and Mohd Hilmi Hasan²

¹Department of Computer Science, Air University Multan Campus, Multan 60000, Pakistan

²Centre for Research in Data Science, Department of Computer and Information Sciences, Universiti Teknologi Petronas, Seri Iskandar 32610, Perak, Malaysia

Correspondence should be addressed to Tauqeer Safdar Malik; tauqeer.safdar@aumc.edu.pk

Received 15 May 2020; Revised 5 July 2020; Accepted 25 July 2020; Published 13 August 2020

Academic Editor: Shuping He

Copyright © 2020 Tauqeer Safdar Malik and Mohd Hilmi Hasan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the existing network-layered architectural stack of Cognitive Radio Ad Hoc Network (CRAHN), channel selection is performed at the Medium Access Control (MAC) layer. However, routing is done on the network layer. Due to this limitation, the Secondary/Unlicensed Users (SUs) need to access the channel information from the MAC layer whenever the channel switching event occurred during the data transmission. This issue delayed the channel selection process during the immediate routing decision for the channel switching event to continue the transmission. In this paper, a protocol is proposed to implement the channel selection decisions at the network layer during the routing process. The decision is based on past and expected future routing decisions of Primary Users (PUs). A learning agent operating in the cross-layer mode of the network-layered architectural stack is implemented in the spectrum mobility manager to pass the channel information to the network layer. This information is originated at the MAC layer. The channel selection is performed on the basis of reinforcement learning algorithms such as No-External Regret Learning, Q-Learning, and Learning Automata. This leads to minimizing the channel switching events and user interferences in the Reinforcement Learning- (RL-) based routing protocol. Simulations are conducted using Cognitive Radio Cognitive Network simulator based on Network Simulator (NS-2). The simulation results showed that the proposed routing protocol performed better than all the other comparative routing protocols in terms of number of channel switching events, average data rate, packet collision, packet loss, and end-to-end delay. The proposed routing protocol implies the improved Quality of Service (QoS) of the delay sensitive and real-time networks such as Cellular and Tele Vision (TV) networks.

1. Introduction

Cognitive Radio (CR) was first coined by Mitola et al. in 2002 [1]. CR technology is yet different from conventional wireless radios since it can opportunistically detect the available channels of wireless spectrum [2]. It is the foundation for CR Network establishment. This is made possible through its network layer capability that controls communication and the spectrum awareness between layers. In this case, the layers are Medium Access Control (MAC) and Network layers. Overall, CR is about providing localized control of radios within one node/user, while CR network functions according to end-to-end controls of network performance. The end-to-end controls are governed at run-

time by the requirements of operators, users and applications, and the available resources. The difference in control from local to end-to-end enables easier operation for CR network across all network protocol stack layers [3]. In CR networks, Primary Users (PUs) are supposed to be the legitimate licensed users while the Secondary Users (SUs) are unlicensed users. We can classify the CR networks on the basis of their architectures such as infrastructure-based and infrastructureless networks. The former networks are developed through a centralized Service Access Point (SAP) while the latter networks are established without the centralized architecture; such networks are also called as CR Ad Hoc Networks (CRAHNs) [4]. In infrastructure-based CR networks, SAP manages the network operations just like a

base station in the cellular networks. On the other hand, SUs in CRAHNs can communicate with each other in a peer to peer fashion [5]. The ultimate goal of the CRAHN is to choose and assign channels to SUs that are currently not being utilized by the incumbent PUs [6].

In CR networks, PUs and SUs have unique rights in terms of channel utilization. PUs are the incumbent users having priority rights to occupy the licensed channels. On the other hand, SUs are less privileged users such that they can only access the licensed channel whenever the PUs are inactive. Therefore, each SU needs to select its transmission parameters based on channel utilization rights. The transmission parameters, for instance, channel availability, transmission rate, and transmission time, are dependent on the time-varying availability of the channel and user type. An SU can utilize a licensed channel in the absence of PUs whenever the PU returns and SU needs to revoke its transmission on that channel. However, it can switch to any other available channel to resume its transmissions. The frequent arrival of PU can lead SU to observe incremental channel-switching events which can seriously degrade the Quality of Service (QoS) during the end-to-end routing process at the network layer. To maintain the QoS during routing, it is very important to manage the time-varying availability of transmission parameters, such as channel availability, type of modulation, channel transmission rate, and transmission time, during the whole communication process of SUs. Therefore, the CRAHN must act as a highly intelligent network so that it can intelligently change its transmission parameters and maintain QoS during SUs' transmissions.

The CRAHN should also have the abilities of self-management and self-awareness so that the routing parameters can change on the basis of the current network requirements in a decentralized way. The routing parameters can be selected by each SU through the spectrum mobility and time-varying availability of a channel, known as Dynamic Spectrum Access (DSA) [7]. Federal Communications Commission (FCC) allowed DSA implementation in 2003 [8]. For DSA in CRAHNs, routing parameters like delay, link length, capacity, throughput, channel availability, and/or user interference, are directly related to the QoS required by the application [9]. User interferences are stated in DSA as the unexpected arrival of the PU on its licensed channel and contention between SUs on channel selection. SU must switch a channel to any other available channel to continue its transmission during the routing process to avoid harmful interference. As more and more unexpected arrivals of PUs occur, more and more channel switching events happen, thus degrading the QoS during end-to-end routing. According to user characteristics, user interferences can be categorized into interflow and intraflow interferences. The inter-flow interference occurs between PU and SU when the unexpected arrival of PU happens. However, the intra-flow interference can occur between SUs themselves due to channel contentions on the MAC layer. So, the property of DSA in terms of spectrum mobility and time-varying availability user interferences management is not only for efficient routing process but also for the better utilization of channels.

The QoS that is observed by the users is supposed to be the overall performance of any network. In this regards,

various quantitative parameters like average data rate, packet loss, and end-to-end delay (EED) are used to measure the QoS observed by the different users. The overall throughput of any network is dependent on the end-to-end data delivery without any packet loss or delay to maintain the QoS. It is very challenging and demanding for SUs to make decisions of end-to-end routing by selecting the appropriate channel for its transmission in the CRAHNs so that QoS can be maintained during the transmission with less channel-switching events. A routing protocol for the CRAHNs must have the ability to select the appropriate channel where less channel-switching events occur due to user interferences.

In this paper, we propose a new channel selection routing protocol for CRAHNs, which is implemented in the network layer with multiple disjoint PUs operating on the same frequency channels. The proposed routing protocol is able to minimize the number of channel switching events by minimizing user interferences through Reinforcement Learning (RL) techniques such as No External Regret learning, Q-learning and Learning Automata. The proposed protocol adds the channel information through the header of Route Request (RREQ) and Route Reply (RREP) as the List of Available Channels (LAC), Channel Assigned (CA), Channel Access Duration (CAD), and Path Identifier (PI). We have analyzed the performance of the proposed protocol in Network Simulator (NS-2) and compared with various routing protocols. Simulation results reveal that our proposed routing protocol outperforms existing routing algorithms in terms of packet loss, number of channel switching events, and end-to-end delay.

The key contributions of this paper are listed as the following:

- (i) We propose a routing protocol for CRAHNs to minimize channel switching events during network transmissions.
- (ii) We have implemented RL techniques to retrieve channel information in the route discovery messages so that SUs can make judicious decisions while making judgements regarding channel-switching at the network layer.
- (iii) We analyze the performance of the proposed routing protocol in terms of packet loss, number of channel switching events, and end-to-end delay and compare the results with existing routing protocols.

The rest of this paper is organized as follows. We summarize the overview of the related works in Section 2 and after then the proposed routing protocol is discussed in detail with its working and implementation. The simulation environment is explained in Section 5 followed by the Results section, and finally we conclude the paper in Section 7.

2. Related Work

Routing issues are considered in the CRAHN implementation to assist in making the route decision and become part of future planning to provide better route on the previous decision's basis. The two major decision planning frameworks applied to CRAHN are Markov Jumping

Systems (MJSs) and game theory. Game theory is also differentiated from optimization theory in their ability to model multiagent decision making scenarios whereby the decisions of each agent affect each other. Meanwhile, MJSs have been applied extensively in communication network which includes a routing framework for single agent in single and multiple states decision and planning [10]. MJSs approach is nonlinear for an “optimal” control problem in which the aim is to select actions that maximize some measure of long-term reward [11]. For MJSs, there exist many results on Kalman filtering, H ∞ filtering, passive filtering, and dissipative filtering [11]. However, it should be noted that most of the developed filters are mode-dependent. This may limit their applications in some complex network environments. One solution is to design asynchronous control filters for a class of Hidden Markov Jumping Systems (HMJSs) [10]. HMJSs have been extensively used in CRAHN for a wide range of problems. They can be used for spectrum prediction, PU detection, signal classification, etc. A potential drawback when using HMJSs is that a training sequence is needed, with the training process being potentially computationally complex in case of routing in CRAHN. Therefore, if the probabilities of MJSs are unknown, then the problem becomes a RL task.

In RL, an agent aims to determine a sequence of actions or policy which maps the state of an unknown stochastic environment to an optimal action plan. We note here that MJSs, on the other hand, address this planning problem for known stochastic environments [12]. Since RL agents work in a stochastic environment, they have to balance two potentially conflicting considerations: on one hand, it needs to explore the feasible actions and their consequences (to ensure that it does not get stuck in a rut), while on the other hand, it needs to exploit the knowledge, attained through past experience, of favorable actions which received the most positive reinforcement. A cross-layer routing approach is proposed to improve the QoS parameters for multimedia applications in CR Networks. However, in this solution, the routing is performed in a centralized way, and hence, it is not applicable for the distributed environment such as CRAHN [5]. Several learning solutions for CRAHN have been proposed to address the load balancing and characterization of channel stability of routing problem [13, 14]. To this end, researchers have proposed many metrics for improving the link quality of the CRAHN such as extensions of the Expected-Transmission-Count (ETX) metric [15]. In [16], spectrum allocation strategy is used to improve the QoS for Cognitive network, which is not suitable for varying the link reliability Spectrum-aware routing, by introducing a new routing metric to locate the available channel through PUs’ activities. In [17], a new routing metric is developed for the link positions by adding extra functionalities of Cat swarm approach to improve energy efficiency. However, the implementation of this approach is not sufficient to support link quality in DSA paradigm due to the limitations of channel movement. In CRAHN, routing protocols have twofold objectives, which are finding a path from source to destination and avoiding channels used for the PUs’ transmissions. This routing solution improves QoS

requirements in the domain of event-driven applications yet not applicable for environment learning applications [18]. We can find a thorough overview of routing metrics based on QoS, and factors influencing the performance of routing protocols are proposed in [19]. Furthermore, the RL-based routing protocols are investigated in [20], which have shown the need for new routing metrics to handle user interferences in the DSA environment of CRAHN. RL can be employed without training data as its objective is to capitalize on the long-term online performance [21]. In the CRAHN, the two most crucial tasks of routing protocols are offering reconfigurability due to channel switching and managing the end-to-end route in the time-varying availability of the channel due to user interferences [22]. Hence, Q-learning can be used as a model-free-based RL approach while implementing CRAHN routing tasks on the basis of reward and penalty [23]. On the other hand, Temporal Difference (TD) learning approach of model-free RL is used to implement any action on the basis of guess and guess again which is updated based on another guess. Q-learning is rather a better choice to be utilized for the purpose of CRAHN routing to decide on the selection of future actions from those with reward or penalty based on explorations of the dynamic environment.

The geographic forwarding routing protocol based on spectrum awareness jointly undertakes path and channel selections so that the regions of PUs’ activities can be avoided during route formation [24]. However, avoidance mostly does not fulfill CRAHN’s exploitation requirement. Hence, a modified version based on spectrum awareness is used to minimize the overall hop count [25]. However, many complexities are exposed during the process of the data transmission such as topological changes, faulty nodes, and link degradation which cannot be handled using the avoidance technique [26]. A stability-oriented routing protocol is presented to find a stable route, which comprises link quality and user interference when PUs become agile [27]. The other strategy presented is based on a probabilistic approach with exact ways by which to locate an efficient path in the networks that are random in nature [28]. The least priced path routing based on DSA for the CRAHN is presented to minimize the EED for the opportunistic transmission of data [24]. This least priced path affects each hop that lies within the routing path and the transmission becomes slower for the overall network.

Two metrics, i.e., Frequency Diversity (or Link Stability) and Channel Stability (or Path Stability), are used to count the lowest and balance spectrum utilization areas of PUs’ activities for path stability in [29]. These metrics are based on the busyness ratio of PUs. When the busyness ratio of PUs increases, channel switching delay increases. In [30], another algorithm uses three metrics to handle user interference between PUs and SUs. However, it creates multiple paths which results in larger routing tables in every SUs. In [31], a routing protocol for CRAHN is presented as a new routing protocol especially for the multipath basis by adding a new metric for channel and path selection at the same time. This protocol ensures the path stability in terms of high connectivity but not offers the best QoS path. Similarly, recently

a routing solution is offered to reduce the channel switching events during the transmission using the mobility pattern of PU [32]. This protocol offers the routing predictions based on the user mobility and previous routing pattern matching, which results in higher channel selection time due to the lower mobility and routing pattern prediction. The limitation of lower routing prediction is managed by proper reinforcement learning mechanism implemented on network layer [33]. However, the problem of channel switching events is still an open challenge due to the limitation of user interference [34]. The routing protocol which manages the channel switching events so that the user interference is managed is still in its infancy. In best route selection, best channel availability is not considered which creates the problem of multiple channel-switching events. Hence, multiple routing paths create frequent channel switching events due to user interference and overall network performance is degraded than the CRAHN's QoS requirements. User interferences are managed during end-to-end routing decisions using the PU activity On-Off model in RL-based routing protocols. However, the effect of user interferences on channel switching has not been addressed in spectrum mobility approaches [33]. The effects of user interferences were not differentiated into interflow and intraflow interferences during routing decisions to minimize packet collisions. The channel switching due to user interferences in routing must be addressed to manage spectrum mobility on the network layer. These issues need to be handled to minimize EED and packet loss. By doing so, the overall throughput of the network could be improved in RL-based routing protocols.

3. Methodology of the Proposed Minimization of Channel Switching and User Interferences (MCSUI) Routing Protocol

Our proposed protocol extends the functionality of the existing network-layered stack to accommodate the channel selection decisions in the network layer for minimizing the channel switching and user interference overhead during the end-to-end routing process. There are three modules appended at the network layer named Network Tomography, Minimization of Channel Switching and User Interferences (MCSUI) routing, and QoS and Error Control as shown in Figure 1. The MCSUI routing module is the core of the proposed routing protocol, in which routing tables are created based on the channel selection information passed from the MAC layer through spectrum sensing. The routing tables are updated through the learning agent at the network layer for the channel selection decision. The functionality of the learning agent is correlated within the spectrum mobility manager to manage channel switching and user interferences for resource allocation and event monitoring in a cross-layer fashion.

The learning agent estimates the quality of the routing path based on the available channel list (which is received from the MAC layer through the spectrum mobility manager) in a cross-layer approach. For the routing tables,

various parameters such as the number of channels switching events, channel transmission rate, next-hop, and all available routes from source to destination for each link are included. The choice of multiple paths and channels is saved in the routing table and if a PU suddenly returns to its channel, the SU needs to switch to any other available channel. This sudden arrival of PU is defined as user interference by SU and due to the increment of user interference, the channel switching events occur, and hence, the routing table increases.

The implementation of Artificial Intelligence- (AI)-based RL techniques is the core of the proposed routing protocol to handle and properly manage the user interferences. The user interferences are observed and saved in the learning block so that the appropriate channel is selected for future routing decisions. The decision block coordinates with the learning block for making a decision on channel selection as shown in Figure 2. Further, it correlates with the QoS and Error Control to select the best available channel among the List of Available Channel (LAC) according to its traffic type. The decision of the best available channel depends on the history of channel selection by PU. The channel parameters are also selected with routing parameters in order to improve the QoS in terms of average data rate, packet loss, and End to End Delay (EED). All modules work collaboratively using the learning agent through the spectrum mobility manager in a cross-layer approach. The modifications in the functions of the learning and decision blocks are discussed in the following subsections.

3.1. Learning Block. The learning block learns through exploration and exploitation learning to select the best available channel based on the saved history information of user interferences. The exploration and exploitation of channel selection are tracked using AI-based RL techniques. Three RL techniques including No-External Regret learning, Q-learning and Learning Automata are used for the selection of best available channels for routing. The exploitation learning is based on No-External Regret learning to utilize the previous (past) best channel selected for a successful routing decision. On the other hand, Q-learning is used for the exploration of newly available channels for SUs' transmissions. The implementation of No-External Regret learning is beneficial for updating the saved channel information so that the routing table size can be minimized in case of bad channel selection. Finally, the Learning Automata technique is used to balance between the exploitation and exploration learning for channel selection by maximizing or minimizing the rewards of any channel. Hence, a channel is saved and it remains in the available channel list based on the reward value.

The available channel information such as channel transmission rate and channel ID is passed from the MAC layer to the network layer through the message exchange process between SUs by modifying the existing RREQ, RREP, and Redirecting messages. These messages are maintained for the reward (maximum reward) or penalty (minimum reward) of a channel from Learning Automata

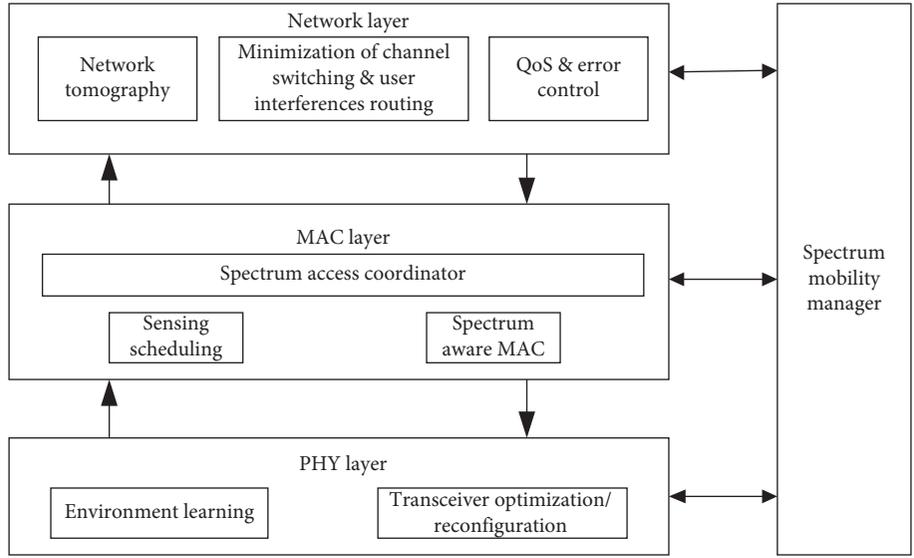


FIGURE 1: Communication model for the proposed routing protocol.

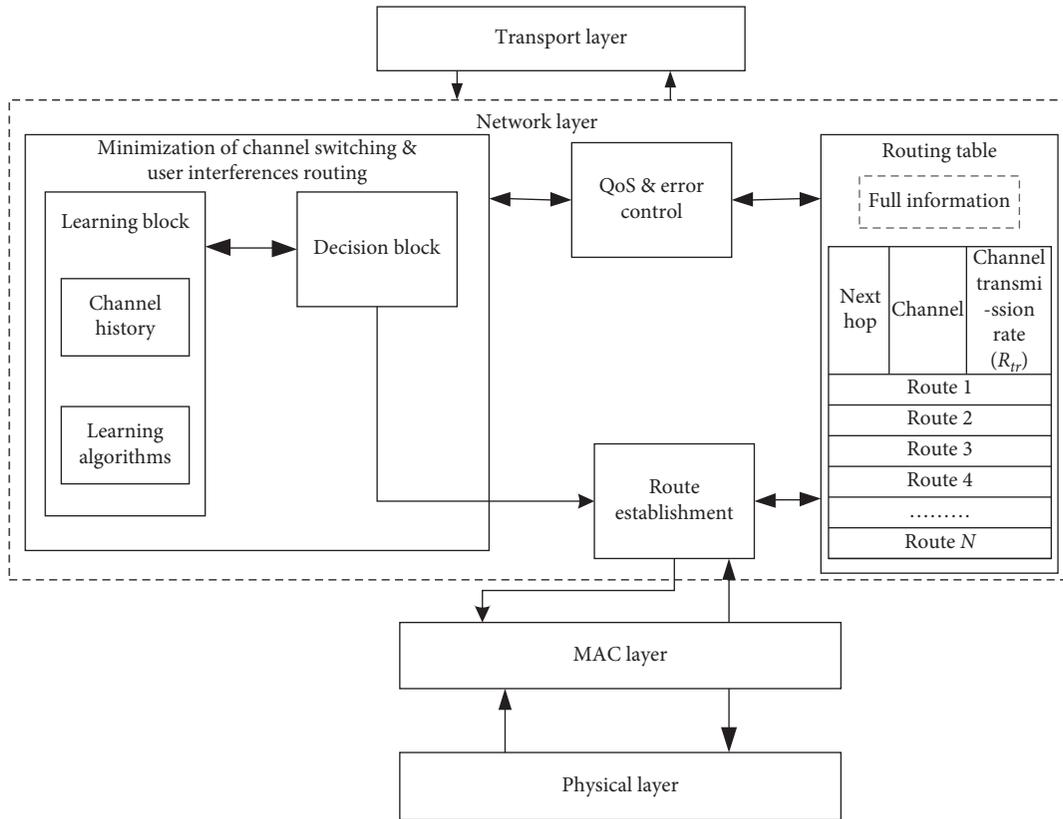


FIGURE 2: Implementation of MCSUI routing protocol at the network layer.

through the hello interval and active_route_timeout (ART) parameters. These two routing parameters, hello interval and ART, in the Ad-hoc On-demand Distance Vector (AODV) [35] routing protocol specify the value of the lifetime for node-to-node connectivity.

The channel is selected through the message exchange process using the learning mechanism during the end-to-

end routing decision. Whenever the SU wants to send a new transmission, it has to send the RREQ message to the intermediate node SU and the neighborhood status is updated in the SU by using the database of the available channel list. It is maintained and updated in the learning block using the No-External Regret learning and Q-learning. The intermediate node then accesses the new channel list and sends the

Redirecting request message to the neighboring intermediate nodes. The Redirecting request message is used to update the LAC through all neighboring nodes (or SUs). In this way, all the SUs have the same available channel information and have no competition in channel selection, and hence, intraflow interference can be minimized, as shown in Figure 3. The destination is selected on the basis of Redirecting Reply messages of different neighboring nodes. In the case of inter-flow interference, the message exchange mechanism works in the same way except that the intermediate node is a PU. Once the Redirecting request is received, the neighboring nodes evaluate its validity in correspondence with the message and update of the ongoing traffic flow using the message exchange process. The neighboring nodes then send the Redirecting Reply message to the intermediate node. Finally, the Route Reply (RREP) message is passed to the source node. The best available channel is selected which is user interference-free on the basis of reward and punishment value of Learning Automata technique.

3.2. Decision Block. The channel is selected on the basis of various pieces of channel information with the help of decision block. For this purpose, the routing may avoid channels which have a high level of PU interference. The decision block co-ordinates with the learning block to select the best available channel according to the QoS and Error Control requirement. Furthermore, the decision block carries out the route establishment after a channel is selected and passes that channel to the routing tables. The selection of the best channel is not only dependent on channel availability but also on the QoS parameter for that channel. The learning block is providing the LAC on the basis of previous (past) and present channel selection decisions. At the channel selection time, the QoS and Error Control parameters such as traffic type, interference, channel bandwidth, transmission time, and Packet Error Rate (PER) are also essential for end-to-end routing. Therefore, QoS requirement of a SU is also incorporated within the learning block using the decision block. The detailed implementation of RL techniques in the proposed routing protocol is discussed in the next section.

4. RL-Based Proposed MCSUI Routing Protocol

Most RL algorithms can be classified into becoming either model-free or model-based. In the model-based approach, the agent builds a model of the environment through interaction with it is typically in the form of an MJS analogous to the approach taken in adaptive optimal control with input time-delays. With a model in hand, given a state and action, the resultant next state and next reward can be predicted. This allows for planning through which a future course of action can be contemplated by considering possible future situations before they are actually experienced. Based on the MJS model in the model-based approach, a planning problem is solved to find the optimal policy function with techniques from the related field of dynamic programming. The commonly used

algorithms to solve MJSs include the celebrated dynamic programming algorithms of online value iteration and online policy iteration. In online value iterating learning techniques, the optimal policy is calculated on the basis of optimal value function. In online policy iterating learning techniques, on the other hand, the learning is directly performed in the policy space. We are using the Q-learning as an online value-iterating model-free technique and learning automata as online policy-iterating technique. In the online model-free approach, on the other hand, the agent aims to directly determine the optimal policy by mapping environmental states to actions without constructing an MJS model of the environment [12].

The proposed MCSUI routing is based on RL techniques using the existing AODV routing protocol mechanism. Therein, the route set-up is in accordance with an expanding ring search mechanism, whereby it uses RREQ and RREP messages of AODV routing. The maintenance of route utilizes Route Error (RERR) packets generated due to SUs' mobility and wireless propagation instability. However, SUs MCSUI routing is capable enough to obtain channel information of licensed channels without causing interference and delay to incumbent PUs. Moreover, SUs should be able to accomplish channel selection from the spectrum mobility provided by the CR environment without causing excessive overhead for route formation. Three RL techniques are used to modify the routing mechanism in the CRAHN, namely, No-External Regret learning, Q-learning, and Learning Automata. Various routes emerge through different channels, and each route is derived through the various channels using exploitation learning. The selected channel must be idle and user interference-free from PUs' activities to successfully make an end-to-end transmission through this route. Therefore, this routing strategy is beneficial in finding user interference-free channel for the whole transmission. The exploration in Q-learning allows SUs to explore various routes through different channels for a transmission to overcome user interference during transmission. The channel is selected either through exploitation or exploration. In Figure 4, a path selection is used as a routing path selection through one of the channel selection decisions either from exploitation or exploration.

One advantage of this strategy is the handling of routing loops through the route maintenance process for handling PUs' activities. The Route Error (RERR) message is used to inform all intermediate nodes of a route that the link has failed; hence, a new route is needed, while the route maintenance process derives an additional type of message to handle the PU activity as PU-Route Error (PU-RERR); that is, the PU-RERR message is utilized to tell neighbor nodes that some PU activities are detected on a specific channel and that a new channel is needed to accomplish the transmission. The routing process is implemented with the help of RREQ, RREP, and PU-RERR messages.

The RREQ message to update the routing table is shown in Figure 5, whereby the channel selection process is started when an intermediate node receives a RREQ message through the available Channel i ; then it sends back a reverse route to source on the same channel. In case of the intermediate node, a valid route can provide the channel

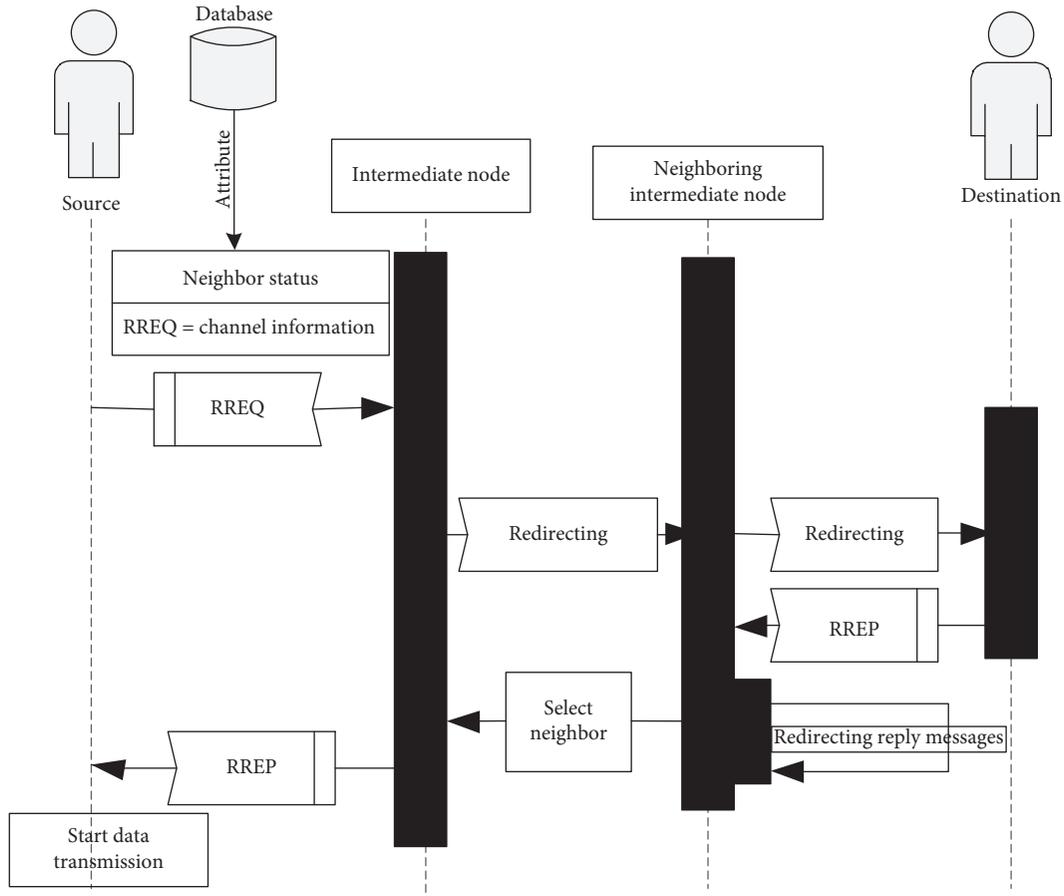


FIGURE 3: Message exchange process using learning block.

information for the desired destination. Finally, a unicast RREP will be sent to the source through the reverse route for the selected channel. On the other hand, if it cannot provide a valid route, it re-broadcasts the received RREQ message on the same channel to all other neighboring nodes. If any additional RREQ message is received for the same source and destination by the same intermediate node on the same (or different) channels, the received RREQ message is then compared against all available routes stored in the routing table. If the reverse route or the received RREQ message contains a better route, then it is selected for the transmission and stored in the routing table as a newer route or better reverse route. It is simply discarded otherwise. The different routes available through the various available channels are stored in the routing table and updated in a reactive manner. The various channels are selected on the basis of exploitation and exploration learning and available on the network layer for the routing purpose. If any channel is unavailable for a stored route, it is referred to as 'regret' and the route is discarded from the routing table using the exploitation learning technique of No-External Regret. To minimize regrets, exploration learning (Q-learning) is used to find options for new routes through the available channels for routing. At this stage, Learning Automata helps to select the best available channel among all available channel lists maintained in learning block for a specific route.

The process of RREP is shown in Figure 6. According to the process, when the first RREP message is received by an intermediate node from an available channel selected through the learning process, it sends a forward route to the destination on the same channel. It also forwards the RREP message with the reverse route available on the same channel stored in its routing table. If an extra RREP message is received for the same source and destination by any intermediate node, then it is compared against the stored reverse route on the same or different channels. If the forward route is better than the stored one, then it is processed and updated in the routing table; otherwise, it is discarded.

The unexpected arrival of the PU is handled by providing the available channel list on the network layer through route maintenance during routing as shown in Figure 7. When any PU activity is detected through the Poisson process, the mean value will be assigned using the Box-Muller transform method on the selected channel through exploitation or exploration. The SU abolishes all the routing entries available in the routing table from that channel using a PU-RERR message. The SU also informs all the neighbor nodes that the channel is currently unavailable. All other SUs that receive the PU-RERR message also abolish the routes from all channels which involve the channel of the source of the PU-RERR message. In this way, the proposed routing protocol minimizes the switching delay and manages channel

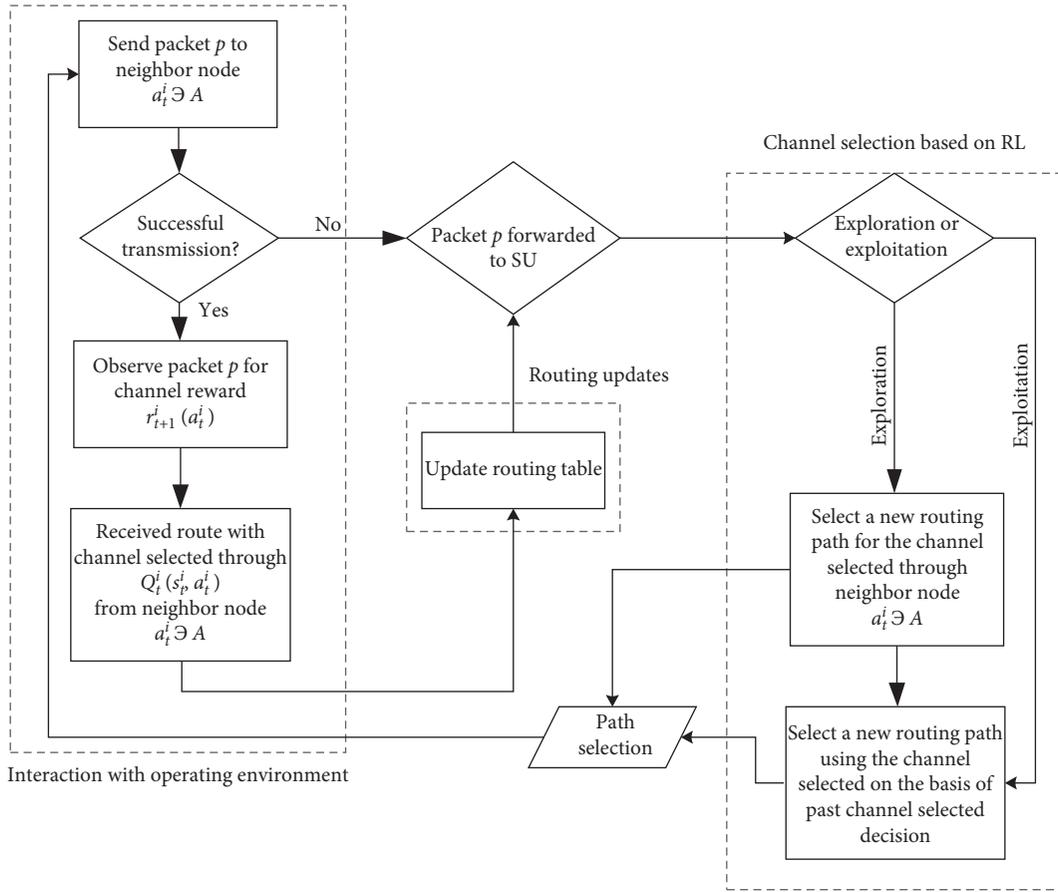


FIGURE 4: Path selection process in the proposed MCSUI protocol.

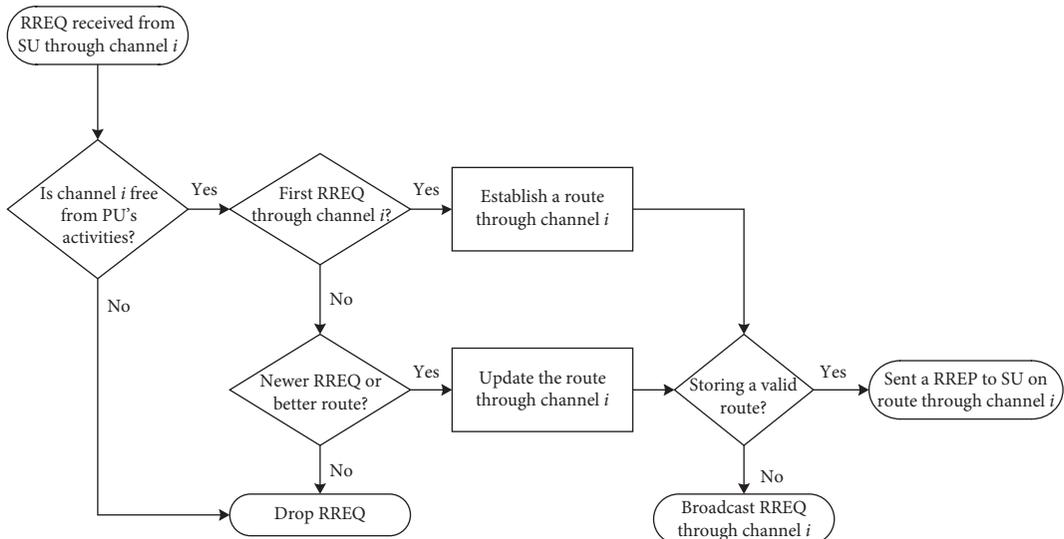


FIGURE 5: RREQ mechanism in proposed MCSUI protocol.

switching events due to user interferences, and therefore, EED is minimized with the improved average data rate. The PU-RERR messages also enable the MCSUI routing protocol to have the spectrum mobility and DSA functionality during routing on the network layer.

The RL-based routing is improved using the PU-RERR message because whenever the SU receives a PU-RERR message, the availability of extra routes is checked in the routing table from other channels for a specific destination. If so, the SU can continue the transmission through other

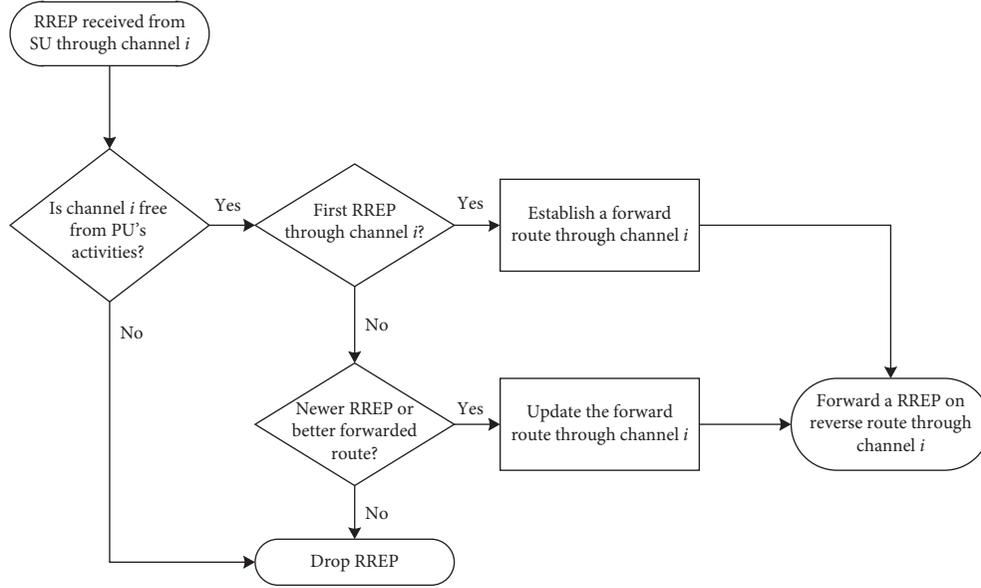


FIGURE 6: RREP mechanism in proposed MCSUI protocol.

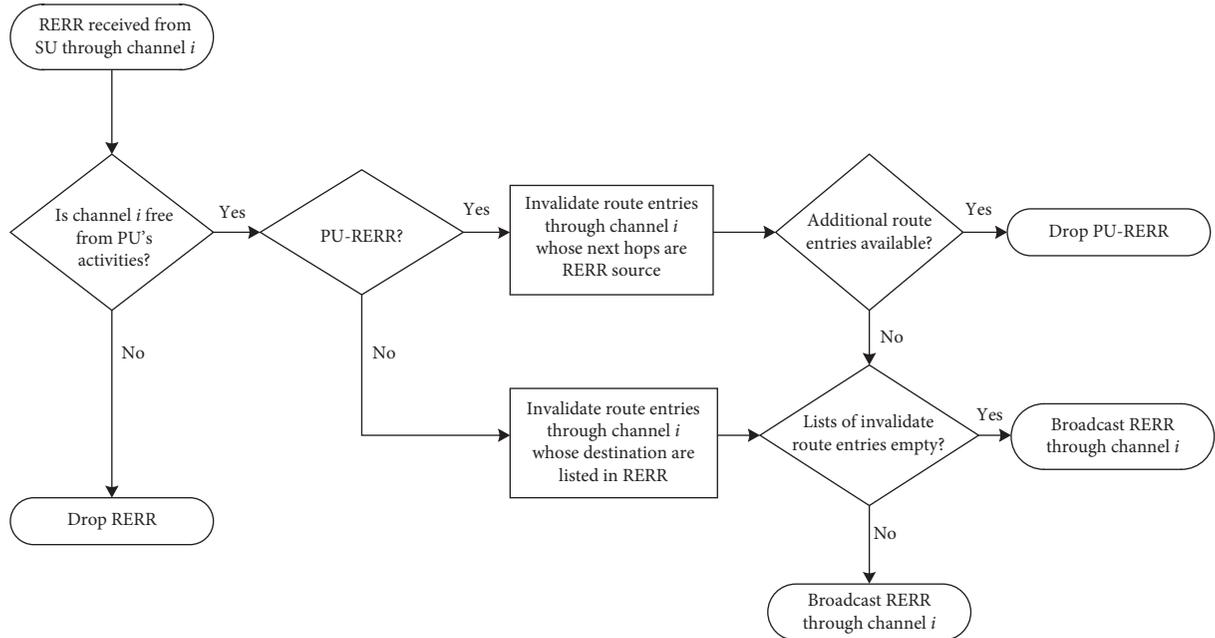


FIGURE 7: PU-RERR mechanism in proposed MCSUI protocol.

available routes from other channels. Else, a new route discovery process is initiated using the traditional RERR message. In order to minimize the EED, the MCSUI routing protocol updates different routes through different channels to minimize the channel switching delay. For this purpose, every SU first identifies the shortest routes using Dijkstra's algorithm from all available routes through various available channels. Secondly, it starts transmission on these shortest paths. The various available channels are selected in such a way that the spectrum mobility allows the MCSUI routing protocol to implement DSA in CRAHNS.

4.1. Channel Selection in MCSUI Protocol. We used the multiagent Q-learning based channel selection model of the network layer in our proposed MCSUI Protocol. The update rule for the Q-learning values for the first agent is as follows:

$$\text{absolute growth of } Q_{a_i} = Q_{a_i}(t+1) - Q_{a_i}(t) = \alpha(r_{a_i}(t+1) - Q_{a_i}(t)), \quad (1)$$

where Q_{a_i} represents the Q-value of agent α for action i at time $t+1$ and time t , respectively, and r_{a_i} accounts for the reward at time $t+1$ by subtracting the previous Q-value of the agent α . This difference indicates the absolute growth in

Q_{a_i} between time t and time $t + 1$. The approximate growth of Q_{a_i} for the small amount of time (for the continuous time version of $\Delta t \in [0, 1]$) is given by

$$\begin{aligned} \text{apro growth of } Q_{a_i} &= Q_{a_i}(t + \Delta t) - Q_{a_i}(t) \approx \Delta t \\ &\times \alpha(r_{a_i}(t + \Delta t) - Q_{a_i}(t)). \end{aligned} \quad (2)$$

When $\Delta t = 0$ and $\Delta t = 0$, equation (2) becomes an identity equation. The linear approximation can be achieved by equation (2) for the continuous time version between 0 and 1 ($0 < \Delta t < 1$). Hence, the approximation equation for the continuous time version of equation (1) is achieved by dividing Δt and taking the limit for $\Delta \rightarrow 0$ as in [36], given by

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} \frac{Q_{a_i}(t + \Delta t) - Q_{a_i}(t)}{\Delta t} &\approx \alpha(r_{a_i}(t) - Q_{a_i}(t)), \\ \frac{dQ_{a_i}(t)}{dt} &\approx \alpha(r_{a_i}(t) - Q_{a_i}(t)), \end{aligned} \quad (3)$$

which is achieved by applying integration as follows:

$$Q_{a_i}(t) = Ce^{-\alpha t} + r_{a_i}, \quad (4)$$

where C is the integration constant, e^{-x} is the monotonic function, and $\lim_{\Delta t \rightarrow 0} e^{-x}$. Hence, the reward achieved by the Q -values through applying the limit to Equation 4 when $t \rightarrow \infty$ is given by

$$\lim_{t \rightarrow \infty} Q_{a_i}(t) = \lim_{t \rightarrow \infty} (Ce^{-\alpha t} + r_{a_i}) = r_{a_i}. \quad (5)$$

For channel selection, the first agent learns through the learning process and the other user uses the previous learned states by utilizing the exploitation as a reward. The users are user interference-free since the same reward will be generated for the first agent to take a channel selection action, and the channel will be added to the List of Available Channel (LAC). For this case, equation (5) assures the monotonically increasing (or decreasing) of initial Q -values. The reward is monotonically increasing if $Q_{a_i}(0) < r_{a_i}$ and is otherwise monotonically decreasing if $Q_{a_i}(0) > r_{a_i}$. When a SU wants to transmit, it sees the availability time of each channel and if it meets the channel transmission rate and time, the channel is then added to the LAC. The user can use the exploration to find a new strategy for channel selection decisions. If a SU is using exploration, the game is then played repeatedly in such a way that the rewards can be replaced as

$$E[r_{a_i}]E[r_{y_j}] = \sum_j a_i r_j, \quad (6)$$

where $E[r_{a_i}]$ represents the expected reward for the first user and y_j is the strategy for the second user. It is very important that the Nash Equilibrium Point (NEP) is the specific point of the strategy of any user, in which the probability 1 is given to one of the channel selection actions. After this, equations (3) and (4) become

$$\frac{dQ_{a_i}(t)}{dt} \approx \alpha(E[r_{a_i}](t) - Q_{a_i}(t)), \quad (7)$$

which is achieved by applying integration as follows:

$$Q_{a_i}(t) = Ce^{-\alpha t} + E[r_{a_i}]. \quad (8)$$

Hence, if the user is not learning any more for new channel selection decisions in case of exploitation, then the Q -values are selected as an expected reward $E[r_{a_i}]$ in a monotonic function where they are either never decreasing or never increasing. On the other hand, the learning process is used to find new Q -values for channel selection in the case of exploration, which is a complex task, and the expected reward is possibly changed over time. Exploration learning can change the probability which consequently changes the expected reward. The expected reward modifies the associated direction field using equation (7) and so NEP is changed. If the expected reward changes every time, then a new channel selection direction is generated. Both the limit and direction of Q -values are changed by this modification as in equation (8). This mechanism is also responsible for unifying the deal between exploitation and exploration, so that the user can reinforce the evaluation of actions that already known to be good while also exploring new actions. For this purpose, Q -greedy exploration is used to select a random action with Q -probability and the best action which has the highest Q -value with the probability of $1 - Q$. The probability is updated by the Q -greedy mechanism whenever it finds a new action with the highest Q -value. The overall behavior of a user depends on the assembly of these crossing points which define the Q -values.

An important note is that equation (7) cannot be solved in the same way as equation (3) when changes occur in the expected rewards over time although the initial Q -values can be derived from the early direction paths. Another aspect is the updating speed of Q -values that depends on the learning rate. In the learning process, actions hold different probabilities depending on convergence to the NEP. This speed is selected as a constant learning rate for the stochastic random selection problem as $\alpha = 0.1$. The message exchange process is used to update the Q -tables for the learning block to exploit and explore the channel information using the learning mechanism. Learning Automata is used to identify an action as a reward (or punishment) on the basis of its opponent's utility function. So, the average channel reward information can be updated in Learning Automata. The Q -values can be evaluated on the basis of the action's success. The Q -value is marked as a reward when that value gives a successful transmission during which there is no user interference and no channel switching occurs. By contrast, the Q -value is marked as a punishment/penalty for an unsuccessful transmission due to channel switching caused.

The reward value of actions is calculated using the two values of Q , i.e., Q_{penalty} and Q_{reward} . Q_{penalty} resulted from the node taking one of the two actions: decreasing hello_interval and active_route_timeout. In case of decrements in hello_interval and active_route_timeout values, the channel will no longer be available for transmission and the routing choice will not be available through that channel. Connectivity information may be provided by a node through broadcasting the local hello messages. However, this must

only be used if the node is part of an active route. For every `hello_interval`, the node verifies whether a broadcast RREQ has been sent or not in the last `hello_interval` so that it can update the channel selection choices of its opponents. In the case whereby sending has not taken place, it may broadcast a RREP with Time-To-Live (TTL), $TTL = 1$, which is called a hello message with the RREP message. This lifetime value is equal to `hello_interval` multiplied by `allowed-hello-loss` (an integer). Their default values are 1 second and 2 seconds. To manage the network status of instability, exploration is utilized to identify the new action for channel selection using the learning mechanism to reduce the chances of punishments/penalties. Q_{reward} presents the stability status of the network and the node performs actions such as increasing the value of `hello_interval` and `active_route_timeout`. The increments in `hello_interval` and `active_route_timeout` indicate the stability of the route for transmission and also the reward achieved has the highest Q -value. The Q -learning-based calculation of Q_{penalty} and Q_{reward} can be found in [25] as follows:

$$Q_{\text{penalty}} = (1 - \alpha)Q(s, a)_{\text{penalty}} + \alpha Q(s', a'), \quad (9)$$

$$Q_{\text{reward}} = (1 - \alpha)Q(s, a)_{\text{reward}} + \alpha Q(s', a')_{\text{reward}}, \quad (10)$$

which is embedded in each SU to make interference-free routing decisions for channel selection with the support of the learning mechanism. The learning process is accomplished in three stages named state, action, and reward. The state denotes the decision-making factor for channel selection while the reward shows the negative (penalty/punishment) or positive (reward) effect as a result of an action being taken in a state. A positive action is calculated as a reward and negative action as a penalty. A SU i is considered for the reward r selected from the actions $A_i = \{1, 2, \dots, J\}$ through $S = \{1, 2, \dots, N\}$ number of states to show the proposed routing process to reach destination n . The state $s_i \in S$ is the channel selection state of SU i at time t for achieving the reward r_{a_i} through the action $a_i \in A_i$.

Whenever SU i sends packet to the destination at time t , then SU updates the Q -value at time $t + 1$ as a reward for the destination node through the next hop node j in its routing table as follows:

$$Q_{t+1}^i(s_t^i, j) \leftarrow (1 - \alpha)Q_t^i(s_t^i, j) + \alpha \left(r_{t+1}^i(j) + \max_{k \in A^j} Q_t^j(s_t^j, k) \right), \quad (11)$$

where $0 \leq \alpha \leq 1$ is the learning rate, node $k \in A^j$ is an upstream node (opponent node), and j is the next-hop node. The reward $r_i(j)$ shows the successful channel selection for SU i to transmit to the neighbor node j . The Q -value $Q_t^j(s_t^j, k)$ collectively represents the channel transmission rate through $k \in A^j$. This Q -value calculating model is used by SUs for routing decisions to a destination through learning about the available channels from multiple paths for its reward. The multiple paths are explored from the available channels which are affected by the different levels of the PU's utilization. Hence, higher utilization by a PU of a channel

lowers the Q -value for that channel due to higher user interference which results in higher channel switching events and delay for the transmission. For the transmission by a SU i , the action is selected to adopt a policy $\pi_{t+1}^i(s_t^i)$ that selects a SU next-hop node holding the maximum Q -value as

$$\pi_{t+1}^i(s_t^i) = \arg \max_{a \in A^i} (Q_t^i(s_t^i, a^i)). \quad (12)$$

Algorithm 1 shows channel selection through the three learning processes based on RL, which is initialized with the Q -value as 0 at time t and selects a default channel to check the availability of various users. If a packet is received successfully through that channel, the action of transmission is awarded (incremented), and the channel remains same otherwise. This condition is checked for every channel available on the spectrum and average reward is calculated in case of unavailability of a free channel on the spectrum in both ways. Finally, at timestamp t , the Q -value of user i for the strategy is updated based on the average reward of the channel availability. This is due to the channel is selected for the transmission on the basis of action probability calculated through Q -greedy exploration from all the available channels on the spectrum. The reward action is calculated using the RL algorithms and updates the action strategies according to equation (11). These equations are derived through the RL algorithms, No-External Regret Learning, Q -Learning, and Learning Automata. The learning agent is capable of this channel selection mechanism, which is implemented in cross layer fashion of CRAHNs architecture.

4.2. Network Co-Ordination. Suppose that N SUs in a CRAHN opportunistically access M orthogonal licensed channels. Common Hopping will take effect, whereby time-slotting procedure upon the channels is carried out and SUs are synchronously communicating among each other. If no packet requires transmission, all SUs carry out transmission using channels according to the same number of channel sequence. For instance, the sequences of channels as $1, 2, \dots, M$. In this regard, β denotes the time slot length (i.e., the time on each channel transmission). During a transmission attempt, Request-to-Send (RTS), and Clear-to-Send (CTS) packets are firstly exchanged by a pair of SUs during a time slot. When the CTS packet is received by SU transmitter, the channel switching is paused. Moreover, the particular SU transmitter will remain on the same channel for the transmissions of data. However, nontransmitting SUs continue channel switching otherwise. Once the data packet is successfully transmitted, the SU pair can rejoin the channel if required.

In spectrum mobility, different sets of SUs may utilize diverse channels for exchanging control information and constructing several links simultaneously. For instance, this type of channel switching is shown in Figure 8, in which SUs A, B and C, D are two transmitting pairs that intend to initiate new transmissions at the same time. Each SU generates a distinct pseudorandom channel sequence number for its transmission instead of using the same channel number for all the SUs. The channel sequence number for

SU A is 2-4-1-3 and for SU B is 3-2-1-4. The default sequence of channel for transmitting on the channels is followed by a SU when it is in idle condition. When a SU wants to perform data sending to a receiver, it will temporarily tune to the ongoing receiver channel and an RTS packet will be sent during a time slot. In the event that the receiver sends a reply containing CTS packet, channel switching will be stopped by the transmitter and receiver. Then data transmission will begin using the same channel. In the event that the data transmission completes, the default sequence of channel will be resumed by them. It is assumed that strict time synchronization among SUs for the purpose of channel transmission may be accomplished even if the exchange of control messages on a Common Control Channel (CCC) does not occur. It is considered that a synchronization scheme in each SU by including a time stamp for each of the packets it sends. After that, the specific SU receiver's clock information is obtained by the SU transmitter. This is performed using two actions—listen to the corresponding channel as well as estimation of clock drift rate to produce time synchronization. SU transmits the data packet and stops the transmission at the start and end of a time slot, respectively. Consequently, the multiple time slots will represent the SU data packet length, denoted by σ .

4.3. Network Implementation. The activity of every licensed channel is learned in the form of ON/OFF operation to maintain the LAC for the routing purpose. As shown in Figure 9, a PU ON period or data packet on a channel is represented with the gray rectangle while the OFF or idle period is denoted by the white space. The gray rectangle length designates the PU data packet length. Hence, a channel can only be utilized by a SU if there is no PU carrying out the transmission simultaneously. A SU starts channel learning for its availability as t_0 which represents the transmission time of a PU. Hence, at any time in the future t ($t > t_0$), the channel status is represented by $N_i(t)$ for the i -th channel. The $N_i(t)$ notation denotes a binary random variable, representing the idle and busy states with values 0 and 1, respectively. For the packet arrival process, each PU is following the Poisson distribution process with the MAR λ_i and an arbitrary probability density function (pdf) $f_{L_i}(l)$ is followed by the data packet length. We assumed each SU of

two radios. The first radio manages data and controls traffic, and it is known as the transmitting radio. Meanwhile, the second radio, named the scanning radio, is dedicated to scanning the whole spectrum in order to gain the information of channel occupancy. The scanning radio has two functions: (1) monitoring channel transmission time and storing channel information in memory so that channel availability in learning block can be retrieved in the future; and (2) confirming whether the channel that is just selected is idle or not for transmitting SU.

An SU can learn the channel availability before starting the transmission so that the channel switching delay could be minimized. Based on that learning, SU will make a decision from three possibilities: (1) staying in the current channel; (2) switching to a new channel; (3) ending the current transmission according to the history of a channel. Our proposed protocol determines whether a channel switching should follow or not based on the following two criteria: (1) the learning probability that the current channel and the potential channel that could be chosen to continue the ongoing transmission of data (which we called as candidate channel is either busy or idle; (2) the expected transmission of the channel idle duration. The traffic activity of PU user on channel i is shown in Figure 9. In the figure, X_i denotes time of interarrival, while T_i denotes time of arrival. Both times refer to i -th packet.

By following the assumption that the arrival of PU packets is based on Poisson distribution pattern, X_i is exponentially distributed with the MAR λ_i packets per second and the PU packet length follows the pdf $f_{L_i}(l)$. According to Figure 9, for any future time t , the learning probability (L_P) that the i -th channel is busy or idle can be written as follows:

$$\begin{aligned} L_P(N_i(t) = 1) & \text{ if } T_i^k < t \text{ and } T_i^k + L_i^k \geq t, \quad k \geq 1, \\ L_P(N_i(t) = 0) & \text{ if } T_i^k + L_i^k < t \text{ and } T_i^{k+1} \geq t, \quad k \geq 1, \\ & T_i^{k+1} \geq t, \quad k = 0, \end{aligned} \quad (13)$$

where L_k denotes the length of the k -th PU transmission on channel i . Hence, the learning probability (L_P) that channel i is idle at any future time t can be obtained as

$$\begin{aligned} L_P(N_i(t) = 0) &= \int_0^\infty \left[\sum_{k=1}^\infty L_P\left(T_i^k + L_i < \frac{t}{k}\right) L_P\left(T_i^{k+1} \geq \frac{t}{k}\right) L_P(k) + L_P(T_i^1 \geq t) L_P(k=0) \right] f_{L_i}(l) dl \\ &= \int_0^\infty \left\{ \sum_{k=1}^\infty \left[\frac{\lambda_i (t - L_i)^k}{k!} e^{-\lambda_i (t - L_i)} \right] \left(\frac{(\lambda_i t)^k}{k!} e^{-\lambda_i t} \right) \frac{(\lambda_i t)^k}{k!} e^{-\lambda_i t} + e^{-2\lambda_i t} \right\} f_{L_i}(l) dl. \end{aligned} \quad (14)$$

Let t_{off} denote the OFF period duration. The following equation defines the cumulative distribution function (CDF) of OFF period duration for real valued variable t , for the i -th channel:

$$\begin{aligned} L_P(t_{off} < x) &= \int_0^\infty \int_0^{l+x} \lambda_i e^{-\lambda_i t} f_{L_i}(l) dt dl \\ &= \int_0^\infty (1 - e^{-\lambda_i (l+x)}) f_{L_i}(l) dl. \end{aligned} \quad (15)$$

```

(1) Initialize  $Q(Sa_i) \leftarrow 0$ ;
(2) Start with default channel selection;
(3) Transmit packet using multiple access scheme;
(4) while channel  $< C$  do
(5)   if packet received is "yes" then
(6)     Utility of channel is calculated from the arrived packet rate;
(7)   else
(8)     Get channel utility from the ACK packet;
(9)   end if
(10)  Calculate average utility reward using equation (11);
(11)  Update  $Q(s^j)$  using equation (12);
(12)  channel  $\leftarrow$  channel + 1;
(13) end while
(14) Assign channel using the probability reward of Q-greedy exploration;
(15) End of session;

```

ALGORITHM 1: Channel Selection of the proposed MCSUI routing protocol in SUs.

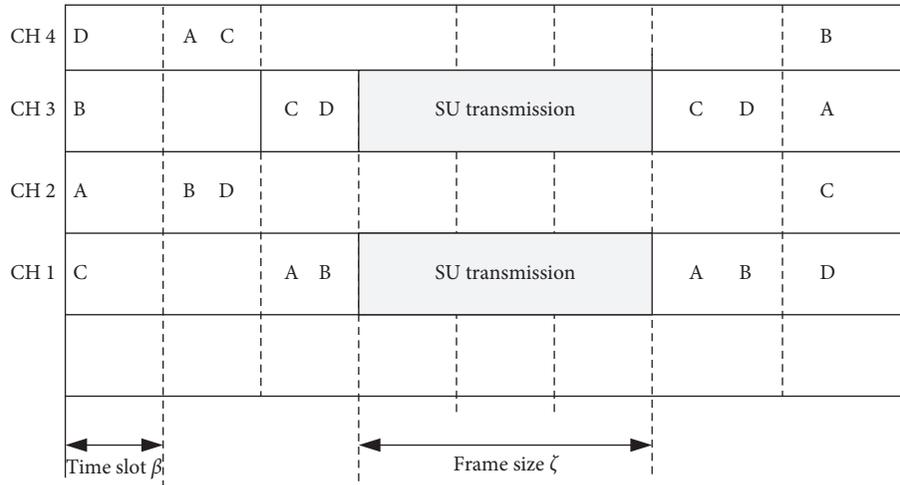
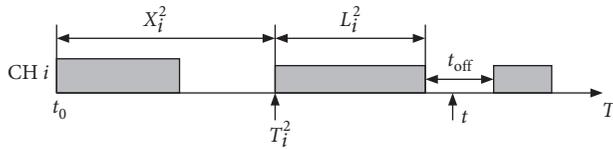


FIGURE 8: An example of the channel switching events by proposed protocol.

FIGURE 9: PU traffic activity on channel i .

The decision which requires a SU to make a switching to a new channel (based on the above learning probability) is as follows:

$$L_P(N_i(t) = 0) < \tau_L, \quad (16)$$

where τ_L accounts for the threshold value of a channel. If τ_L is less than learning probability, then the channel is assumed as being busy and so SU needs to carry out a channel switching event. This means that the channel is not being assumed as idle until the end of the current transmission. Additionally, the decision that a channel j at time t is available simply depends on the following:

$$\begin{aligned}
L_P(N_j(t) = 0) &\geq \tau_H, \\
L_P(t_{j,\text{off}} > \eta) &\geq \Theta,
\end{aligned} \quad (17)$$

where τ_H is the learning probability threshold for a channel H , which is considered as idle at the end of the current transmission, whereby η is the length of transmission plus a time slot (i.e., $\eta = \zeta + \beta$), and θ is the learning probability threshold for a channel to be considered as idle for the next transmission period. This would be helpful to note that the probability of learning that the idleness time of j -th channel is more than transmission time must be higher than or equal to θ in order to support at least one transmission.

For data transmission, SUs first perform the procedure that senses whether the existing operating channel is available or not. Toward this end, our proposed protocol MCSUI assumes that each SU has to wait on the chosen target channel until it becomes idle. An example is shown in Figure 10 to explain the minimization of channel switching delay when channel switching occurs during transmission. Therein, HLU is High-priority Licensed Users (i.e., PUs) while LPU is Low-priority

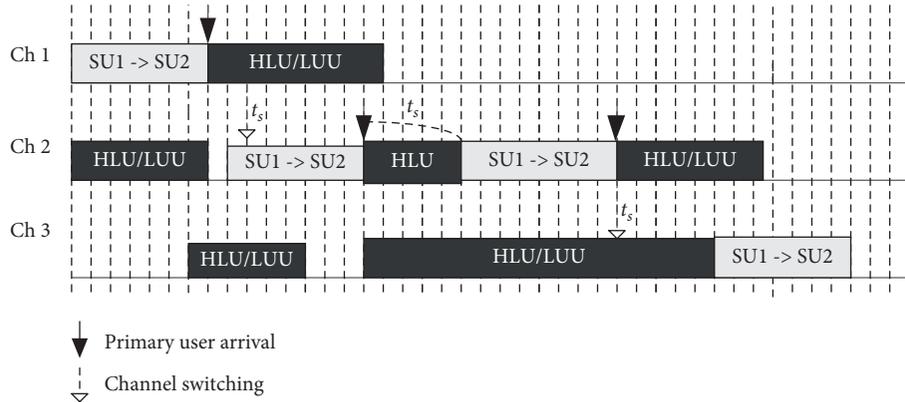


FIGURE 10: Example scenario of transmission process with channel switching.

Unlicensed Users (i.e., SUs). We see channel Ch1 becomes SU1's default channel. Initially, SU1 performs transmission to the matching receiver SU2. The channel switching process is described as the following. Channel Ch1 is changed to the idle channel Ch2 by SU1 during the first interruption. The channel switching time, t_s , represents the channel switching delay. Then, SU1 remains on the existing channel Ch2 during the second interruption. The channel can only be accessed by SU2 after the transmissions are completed by HLU of Ch2. With respect to this, channel switching delay refers to the busy duration produced from PUs of Ch2. SU then performs a change to Ch3 during the third interruption. This is due to the fact that Ch3 is busy, SU1 will only be served after all other users in the ongoing Ch3 queue finish being served. Therefore, switching delay refers to the total of t_s and the waiting time occurred in Ch3. Overall, SU1 transmission finally finishes on Ch3. The total service time refers to the period between the instance of transmission beginning and the instance of transmission completion. Moreover, channel switching delay refers to the duration from the instant of pausing transmission until the instant of resuming the unfinished transmission.

The proposed protocol consists of two parts. The first part describes how an SU pair initiates a new transmission regardless of the channel selection mechanism used during channel switching. If a data packet arrives at a SU, the SU predicts the availability of the next transmitting channel (or the channel of the receiver) at the starting of the subsequent time slot. By referring to the learning results, when the learning probability is satisfied by the channel as in equation (17) for the transmissions of data, an RTS packet will be sent by the transmitter to the receiver using the same channel at the starting of the subsequent time slot. Upon receiving the RTS packet, the intended SU receiver replies a CTS packet in the same time slot. Then, if the CTS packet is successfully received by the SU transmitter, the two SUs pause the channel switching and start the data transmission on the same channel to minimize the channel switching delay. This part is effectively minimized the overall End-to-End Delay (EED) by minimizing the switching delay so that the overall throughput is improved. The second part is based on the proactive channel switching events during the transmission of a SU to determine whether or not the SU transmitting pair has to perform channel switching

to a new channel at the end of a transmission. The decision of channel switching event during the transmission is performed according to the Algorithm 2.

The proposed protocol is able to avoid interference between the SU transmitting pair and PUs using Algorithm 2. It is based on the observed channel transmission time information of an SU, which checks the channel switching policy, as in equation (16), for the current channel by learning the list of available channels (LAC) at the end of the transmission. If the policy is not satisfied at a moment, this means that the current channel is still available for the next transmission. This is shown in Algorithm 2 as next available channel (NAC). Then, the SU transmitting pair does not perform a channel switching and keeps staying on the same channel. However, if the policy is satisfied, the Channel-Switching Event (CSE) is set to 1 as shown in Algorithm 2 online 6; that is the current channel is considered to be busy during the next transmission time and the SUs need to perform a channel switching by the end of the transmission to avoid user interference to a PU who may use the current channel. After the CSE is set, the two SUs rejoin the channel in the next time slot after the previous transmission.

The channel selection during switching is proposed according to Algorithm 2, in which the SUs should update the available channel information to the rest of SUs, so that the SUs must have channel information of neighboring SUs before transmitting at the same channel. Hence, when the CSE is incremented, the ongoing transmission will be paused by those SUs that have to carry out channel switching. The channel will be resumed by them using the identical sequence number, in order to ensure that the same channel is used for the transmission. Nevertheless, each SU follows a default channel sequence which may not be the same as other's channel sequence numbers. To gain the ability to exchange information of channel availability among SUs on the same channel, SUs have to use the same channel sequence number only in the event that they are carrying out channel switching. Meanwhile, the criterion in equation (17) is checked by the SU transmitter for available channels in the spectrum. When there is no available channel, the ongoing

```

(1) Initialization:  $CSE \rightarrow 0$ ;  $DSF \rightarrow 0$ ;  $NAC \rightarrow 0$ ;  $LAC \rightarrow \theta$ ;
(2) for  $j \leftarrow 0$  to  $j \leftarrow M$  do
(3) Learning  $L_P(N_j(t)=0)$ ,  $L_P(t_{j,off} > 0)$ ;
(4) end for
(5) if  $L_P(N_i(t)=0) < \tau_L$  and  $DAT=1$  then
(6)  $CSE \leftarrow 1$ ;
(7) end if
(8) if  $CSE=1$  then
(9) for  $k \leftarrow 0$  to  $k \leq M$  do
(10) if  $L_P(N_j(t)=0)$   $\tau_H \geq$  And  $L_P(t_{j,off}) \geq \theta$  then
(11)  $NAC \leftarrow NAC + 1$ ;
(12)  $LAC(NAC) \leftarrow k$ ;
(13) end if
(14) end for
(15) end if
(16) if  $CSE \leftarrow \theta$  then
(17) Stop transmission and goto Line 1;
(18) else if  $LAC = \theta$  then
(19) Start scanning radio;
(20) Launch channel selection of Algorithm 1 for LAC;
(21) Send CSR packet;
(22) end if
(23) if CSA packet is received then
(24) Switch to selected channel and start scanning radio;
(25) end if
(26) if channel is busy then
(27) Stop transmission and goto Line 1;
(28) else
(29)  $DSF \leftarrow 1$ ;  $CSE \leftarrow 0$ ;
(30) end if
(31) if  $DSF \leftarrow 1$  then
(32)  $DSF \leftarrow 0$ ;
(33) Transmit a DATA packet;
(34)  $DAT \leftarrow 0$  when transmission ends;
(35) end if

```

Note. CSE–Channel Switching Event, DSF–Data Sending for Current Channel, NAC–Next Available Channel, LAC–List of Available Channel and DAT–Data Sending for next channel.

ALGORITHM 2: Channel switching control in SUs during transmission.

transmission will stop immediately. Both SUs switch to the subsequent channel for another time slot, and the channel availability at the starting of the subsequent time slot will be checked by them using equation (17) criteria. However, if the LAC is not empty, Algorithm 2 will be triggered by the SU transmitter, and the sending of a Channel-Switching-Request (CSR) packet comprising information of the newly selected channel in the subsequent time slot will take place. When the CSR packet is received, a Channel-Switching-Acknowledgement (CSA) packet will be responded by the SU receiver. Then, if the SU transmitter successfully receives the CSA packet, the establishment of channel switching agreement between the two SU nodes will occur. Thus, both SU nodes switch to the selected channel and start the data transmission. The switching delay of a channel switching is defined as “the duration from the time an SU vacates the current channel to the time it resumes the transmission”. It is possible that inaccurate prediction is produced and there exists another PU on the channel that the SUs switch to.

Hence, at the beginning of the transmission, the SU transmitting pair restarts the scanning radio to confirm that the selected channel is idle. If the channel is sensed busy, the two SUs immediately resume the channel switching and launch Algorithm 2. The number of available channels for data transmission is maintained in a list named Next Available Channel (NAC). The DAT and DSF denote data transmission requesting data sending for Channel i (current channel) and j (next channel), respectively. The proposed MCSUI routing protocol is not only aimed at minimizing the switching delay but also the number of channel switching events using the learning mechanism incorporated in the learning block for future routing decisions.

5. Simulation Environment

This section presents details of the simulation environment regarding the implementation of the proposed routing protocol. The simulation environment includes

the network model and implementation set-up to present the channel selection for the RL-based routing protocol. The implementation set-up is carried out through the system model, simulation parameters, and assumptions for the implementation of the proposed MCSUI routing protocol. The performance of the proposed MCSUI routing protocol is compared with that of the existing Ad Hoc On-Demand Distance Vector (AODV) [35], Opportunistic Spectrum Access (OSA) [33], and the Coolest Path (CP) [34] routing protocols. MCSUI is compared with the AODV since it is the most frequently used reactive routing protocol for real-world solutions. The CP protocol is used as a benchmark because it is the first routing protocol that identifies the issue of user interference in CRAHN. In addition, the proposed MCSUI routing protocol is evaluated based on the learning mechanism used for routing decisions. For this purpose, the OSA is chosen as the routing protocol to compare the implementation of the learning algorithms.

5.1. Network Model. To simulate the proposed MCSUI routing protocol, a network model of the CRAHNs is implemented with mobile SUs which can dynamically access any available licensed channel. The PUs are implemented as fixed users to utilize their licensed channels. Whenever there are free channels, SUs can gain access for data packets transmissions. The network is modeled in a two-dimensional Cartesian scenario in which the availability of PUs' channels is unknown to SUs. The LAC is collected on the network layer using the learning agent through spectrum sensing of the MAC layer in each SU. The SU uses one of the available channels for its transmission. However, SU switches to any other available channel with the unexpected arrival of a PU. This issue is referred to as channel switching due to user interference at the network layer during the transmission of data. To reduce the effect of PU activity on routing by minimizing channel-switching events during transmission, we consider a network that consists of four SUs with two PUs as shown in Figure 11. The implementations of exploitation and exploration learning are shown in Figures 11(a) and 11(b), respectively. The proposed routing protocol is implemented with four SUs (denoted by SU_A , SU_B , SU_C , and SU_D) within the transmission ranges of four PUs (denoted by PU_1 , PU_2 , PU_3 , and PU_4). SU_A can communicate with SU_D using the routes of $SU_A \rightarrow SU_B$ on Channel 1 and $SU_B \rightarrow SU_D$ using Channel 2. In this scenario, a SU can dynamically switch a channel during routing using exploitation and exploration learning of channel selection on the network layer. Another option for route and channel selection is available through the routes of $SU_A \rightarrow SU_C$ on Channel 1 and $SU_C \rightarrow SU_D$ using Channel 2 depending on the activity of PUs. In this scenario, both the interflow and intraflow interferences can be minimized by PU and/or SU, respectively. This scenario also enables properties of spectrum mobility and dynamic spectrum access during the routing process since the channel and route both can be selected at the network layer for the routing purpose to improve the RL-based routing.

The proposed MCSUI routing protocol jointly learns by exploiting and exploring the route and channel during the routing for end-to-end transmission. This characteristic allows SUs to dynamically select any other available channel and route which are user interference-free to reduce the number of channel switching events. The channel selection is based on RL which not only makes it dynamic but also reduces the user interference. The implementation of these features is carried out through the RREQ, RREP, and PU-RERR messages of the routing protocol.

5.2. Simulation Setup. The implementation setup is carried out using the CR Cognitive Network (CRCN) simulator, which is an extension of the famous Network Simulator (NS-2). The CRCN simulator supports the three layers of the CRAHN architectural stack, namely, the network, MAC, and Physical (PHY) layers. The network layer maintains the neighboring node list and the available channel list for routing purposes. The channel availability information is received from spectrum sensing by the MAC layer. The PHY layer maintains the information, such as transmission power, Signal-to-Interference-Noise ratio (SINR), and the propagation model. All layers share this information with each other through the spectrum mobility manager which is already available in the cross-layer network architecture of CRAHNs. Currently, the CRCN simulator is not modeling the activity of PU to observe the effect of PU's activity on SU. For the proposed MCSUI routing protocol, the PU activity on the channel is modeled as a Poisson process based on an expected mean and a standard deviation with the mean value determined using the Box-Muller transform [36]. The PU's arrival rate is fixed and the mean for the discrete data is used in the implementation setup to calculate the user interferences for the best available channel. The CogMAC protocol is used for spectrum sensing at MAC layer to find the channel availability information, while the SINR is used on the PHY layer.

The mobility parameters such as pause time and speed are selected to describe the changed behavior of speed and direction for a node. The Random WayPoint (RWP) model correlates the changed behavior of speed and direction with the time between two events. The detailed parameters selection is given in Table 1. An adjustable input parameter of the model is not used; it depends otherwise on the speed of the nodes, size, and shape of the area of the network. A higher mobile node speed results in a higher frequency of direction changes of a node for a given area. Therefore, the area selected for the RWP mobility model is 500 meters square (m^2), and the analytical expression of its Probability Density Function (PDF) is used for the speed of the node [35]. The distance and time between two consecutive way points are analyzed for the transmissions of SUs in the RWP mobility model. These way points represent the starting and ending points of a user movement period and are uniformly distributed per transmission. The system model to implement these properties is described in the next subsection.

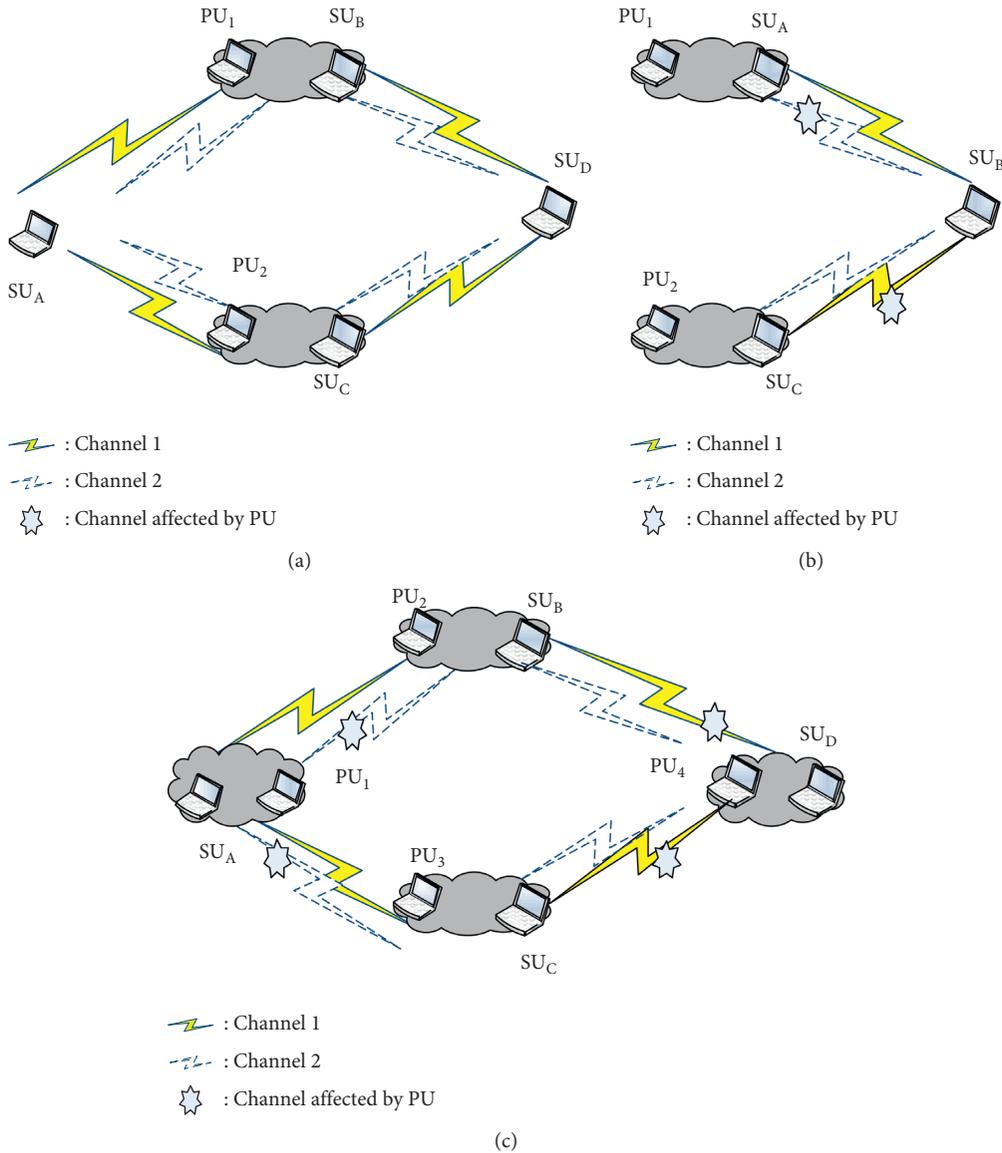


FIGURE 11: Network model of proposed routing protocol.

5.3. *System Model.* The simulation network is defined in a two-dimensional Cartesian scenario of $2000\text{ m} \times 2000\text{ m}$ with 100 mobile SUs and 7 fixed PUs. Simulation results are averaged over 50 runs. Each simulation run lasts for 700 seconds. For spectrum mobility, 10 channels are used with a channel capacity of 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 each to support the wideband spectrum-sensing technique with interference-based detection. This is used to sense over a large spectral bandwidth and to select the channel according to the user's requirement. The models of traffic used for collecting simulation results consist of Constant Bit-Rate (CBR) video conferencing and File Transfer Protocol (FTP) application traffic profiles. In the simulation, each SU changes its location within the network based on the RWP mobility model. According to this model, a node randomly selects a destination, moves toward that destination at a speed not exceeding the maximum speed, and then get

pauses. The interval of pause is known as pause-time. Pause-time ranges from 0 to 240 seconds of duration so as to observe the impact of high mobility on the protocol.

Since the mobile nodes are constantly moving during the simulation, a pause-time of 0 second signifies the worst-case scenario regarding high topological instability. The SU moves using the RWP mobility model in which nodes can move randomly and freely without restrictions. To be more specific, the destination, speed, and direction are all chosen randomly and independently of other nodes. Each user starts by pausing for a fixed number of seconds. The user then selects a random destination in the simulation area and a random speed between 0 m/s and the maximum mobility speed of 15 m/s with random speed selected uniformly. The node moves to the destination and again pauses for a fixed period before selecting another random speed and location. This behavior is repeated for the length of the simulation.

TABLE 1: Default simulation and structural parameters.

Parameter	Value
Simulation area	2000 × 2000 m ²
Number of SUs	100 users/nodes
Number of PUs	18 maximum
Model for PUs	Uniform model
No. of channels	10
Propagation model	Two-way
Round channel bandwidth	2, 4, 6, 8, 10, 12, 14, 16, 18, 20 MHz
Channel mean packet error rate (PER)	0.05 packet/ms
Standard deviation of PER (σ PER)	0.025
Mobility model for SUs	Random waypoint
Mobility speed for SUs	0–15 m/s (uniform)
Pause time	0, 60, 120, 180, 240 seconds
Data type	Best effort
Packet payload	8184 bits
MAC header	272 bits
Mean arrival rate (MAR)	[0.0, 1.0]
Transmission delay by SU (d_{trSU})	1.0 ms
Processing delay by SU (d_{prSU})	1.0 ms
Packets generation at MAR (λ_{SU})	0.6 packets/ms
Learning rate (α)	0.1
Routing weight factor (ω)	1.2 ms
Transmission delay by PU (d_{trPU})	120 bits
Standard deviation of MAR (σ MAR)	0.0, 0.4, 0.8
Mobile user traffic	5 sources
Fixed destinations	2 static and 3 mobile
Packet size	15 kb
Transmission rate	10 packets/sec
Application profile	Low-resolution video

The simulation reporting interval is 1 second, which shows that an average value is calculated for the results at each second of the simulation running time.

Each training episode for the simulation starts from the beginning of the simulation to 30 seconds. Each node initially conducts channel selection at random using the learning agent based on decreasing or increasing the Active-Route-Time out (ART) and hello-interval parameters.

The learning agent uses a 0.1 learning rate according to the randomly distributed radio environment of the CRAHN. During the simulations, the proposed MCSUI routing protocol selects the path using the ART parameter between 3 and 10 seconds and the hello-interval parameter between 1 and 10 seconds. The channel quality parameters are used to evaluate the performance of routing protocols for different packet error rate (PER) and levels of PU's activity. The results are evaluated on the standard deviation of PER (σ PER) and Mean Arrival Rate (MAR) of PU. The MAR for the activity of the PU indicates the channel utilization, channel availability information, channel transmission rate, and channel transmission time. On the other hand, the standard deviation represents the channel access probability of PU. Similar representations are used for the PER as Mean PER and standard deviation of PER but with fixed values for all data channels for the sake of simplicity of the system. The activity of a PU is modeled as a Poisson process with a mean which is assigned using the Box-Muller transform [36] for an expected mean arrival rate of [0, 1] and standard deviation of [0.0, 0.4, 0.8] as given in Table 1 at the SU, and packets are

generated by using the Poisson process with a MAR of 0.6 packets/ms in accordance with the Box-Muller transform. Further, PUs are distributed through Poisson Process in a stochastic network environment between the mean arrival rate of [0, 1]. The mean distribution is checked for the standard error with 0.0 as a low, 0.4 as a medium, and 0.8 as a high arrival rate of PU using the standard deviation of the mean.

6. Results and Discussion

The proposed MCSUI routing protocol is aimed at minimizing channel-switching events, packet collisions, EED, and packet loss. EED is calculated in milliseconds by considering delays in transmission, queuing, processing, and back-off process along with the path from source to destination. The transmission delay of SU is shorter than that of the PU since PUs have higher priority rights than SUs. The queuing delay is assumed as a finite 1000 packets queue size in each SU with 1.0 ms being the fixed processing delay. The Q-values are initialized as zero to inspire exploration at the start of the simulation with a learning rate as 0.1. We have compared the network performance of the proposed routing protocol with the traditional AODV, recent routing protocol Opportunistic Spectrum Access (OSA) based on reinforcement learning, and Collest Path (CP) routing protocols. For a good comparison, the CP routing protocol was chosen in this simulation study since it emerges as the optimal approach for minimizing the SU's

interference to the PU. The route having the minimum accumulated amount of PU's activities will be selected by CP routing; that means, the least number of PUs is encountered by the particular route. So, MAR of PUs along the route may be the lowest. However, OSA routing is based on RL and implemented through a centralized approach that requires network-wide information on the MAR for each link and channel.

6.1. Normalization of Channel Switching Events. The performance of the proposed MCSUI routing protocol is analyzed for the number of channel switching events on the basis of three different standard deviation values of mean arrival rate of PU as shown in Figures 12–14. The results are compared with the AODV, CP, and OSA routing protocols of the CRAHN, and it shows that there is an increment in number of SUs as well as the number of channels with the simulation time passage. In general, users focus on two measurements regarding the spectrum selection. The first relates to the integrated power measurement across the assigned channel, normally being known as the occupied bandwidth (W), power-in-band (or channel power). In this measurement, power is integrated across the channel from the start to assigned channel frequency. In addition to measuring the power in the channel, there is also a need to ensure that transmissions are not leaking into channels assigned to other users, especially those on either side of the licensed channel. A common approach is by filling up the occupied channel with a test signal. This is to measure or compare the integrated power against frequency in the channels that are adjacent to the occupied channel. The PUs are fixed on a channel with activity time on frequency of $1/\lambda = 200$ seconds with the bandwidth of W . The performance of the MCSUI protocol is analyzed in terms of Channel Switching Events against the simulation time in a relatively large network of 100 SUs. As shown in Figure 12, it is observed that the MCSUI protocol initially has a higher number of channel switching events compared to other routing protocols. This occurs due to the Q -value being initialized with zero, and there is no channel available for channel selection at the start of the simulation. Nevertheless, the MCSUI routing protocol converges to a stable state using exploitation as well as exploration learnings as time passes. The learning rate is a decreasing function of time and so in learning algorithms, it works inversely proportional to the time. In the stable state, users have very few channel-switching events. This behavior can be justified because the SUs are distributed among multipath routes on different channels in the MCSUI routing protocol. Therefore, less channel and user contention occurred. This happens as the exploration learning initially started with action 0 for the 0.0 standard deviation of the mean arrival rate of PU, once it reaches to the Nash Equilibrium Point (NEP) through the exploration learning. The NEP works according to the strategies of its opponents and selects the best available channel using the learning mechanisms. As shown in Figure 13, when the standard deviation of MAR is 0.4, the NEP is not achieved due to the

increment in interference of PUs. Hence, the number of channel switching events is increasing due to user activities. The network will utilize the whole learning process for channel selection, and this will require more time. As shown in Figure 14, the number of channel switching events is decreasing. This is happening due to the increment of the participating number of users as the standard deviation of MAR for PU is increased to 0.8 for the transmission. Hence, the MCSUI routing protocol performs better in the reduction of channel switching events compared to the other well-known routing protocols of the CRAHN. The proposed protocol is reducing the number of channel switching events up to 65%, 29%, and 41% than that of the AODV, CP, and OSA routing protocols, respectively.

6.2. Reduction of User Interferences in Terms of Packet Collisions. Each PU activity in a channel is modeled as a Poisson process to observe the effect of user interferences. We consider the Mean Arrival Rate (MAR) of PU and its' standard deviation (sd) of PU arrival using Box-Muller transform, which is based on an expected MAR between $[0, 1]$. The standard deviation for the activity of PU is $\in \{0.0, 0.4, 0.8\}$ that shows the low, medium, and high levels of availability of the PUs, respectively. User interferences are observed for these different values of the availability of PUs according to the stochastic environment property of the CRAHN. The effect of user interferences on network performance is calculated in terms of packet collisions for the MAR of the PUs. It is assumed that the channels have a low level of noise with a fixed Packet Error Rate (PER) as 0.05 and a mean PER (σ PER) as 0.025. The packets are generated in SU to transmit with a fixed MAR (λ_{SU}) of 0.6 packets/ms. The effect of user interferences is observed for each of the three levels of PUs' availability in terms of packet collisions on a channel.

We observed that if the PU's standard deviation with MAR is low such as 0 (user/ms), most next-hop users or nodes can select the same channel pairs, having the same MAR. Hence, all the routing protocols achieve a similar probability of packet collisions by PU and SU across a CRAHN (see Figure 15). This happens due to the unavailability of PU on a channel. So, no activity is detected on any channel. In case of medium (0.4 (user/ms)) PU's standard deviation with MAR, the channel and user pairs have a difference in user interferences of the SU to the PU and the SU to the SU in the MCSUI routing protocol (Figure 16). This reduces the packet collisions with PUs up to 30% compared to the AODV routing. The CP routing reduces collisions with PUs up to 19% and 14% for OSA routing. Similarly, a high standard deviation level (0.8 (user/ms)) of MAR of the PU also shows similar trends to the number of packet collisions with increment in activities of PUs (Figure 17). This happens due to the increment in duration of channel availability and also the number of available channels. It is noted that the MCSUI routing protocol is more appropriate in minimizing user interferences than the other routing protocols, since it uses an additional type of control packet (PU-RERR) to improve the route efficiency.

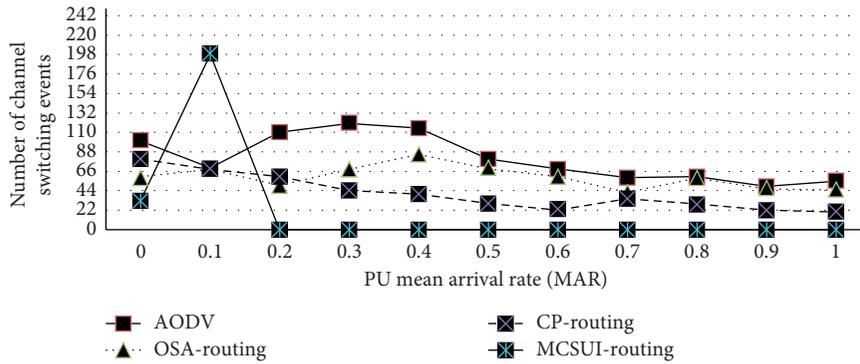


FIGURE 12: Channel switching events of SU at PU's sd of MAR as 0.0 (user/ms).

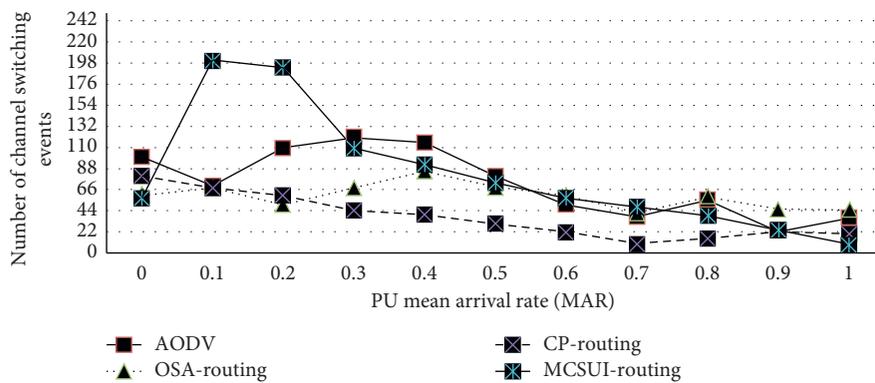


FIGURE 13: Channel switching events of SU at PU's sd of MAR as 0.4 (user/ms).

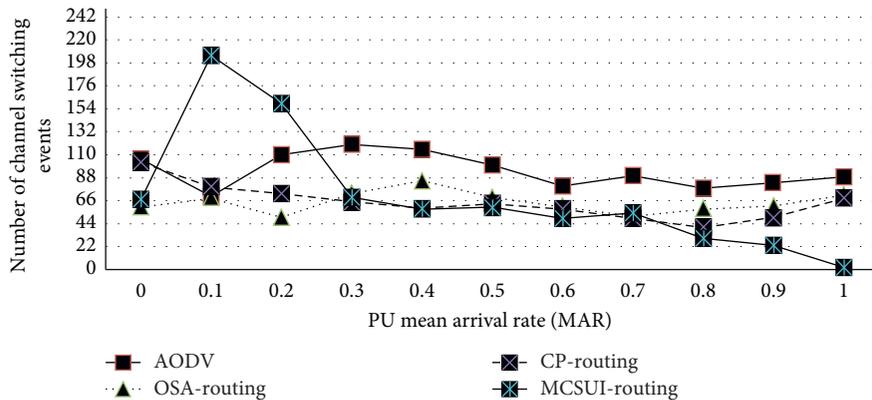


FIGURE 14: Channel switching events of SU at PU's sd of MAR as 0.8 (user/ms).

6.3. *Minimizing in End-To-End Delay (EED)*. The number of channel switching events is reduced due to minimized user interferences, and this effect also helps in minimizing EED. It is observed that the EED of the SU increases with the increase in PU's activities (see Figure 18). The EED consists of switching, transmission, and queuing and back-off delays. When the standard deviation of MAR for the activity of PU is low, the EED is observed as high due to less routing choices available for SUs. The availability of channels and routing paths choices are increases with increment in the

standard deviation of MAR of PUs. When the PU's availability level of MAR increases from 0.4 (user/ms) (Figure 19) to 0.8 (user/ms) (Figure 20), the EED decreases proportionally. The MCSUI routing protocol selects routes that reduces the number of channel switching events caused by PU-SU user interferences contributing to minimizing EED. The MCSUI routing achieves a minimized EED for the SU up to 89% in compared to that of the other routing protocols. This happens due to the increment in the standard deviation of MAR of PUs, which ultimately

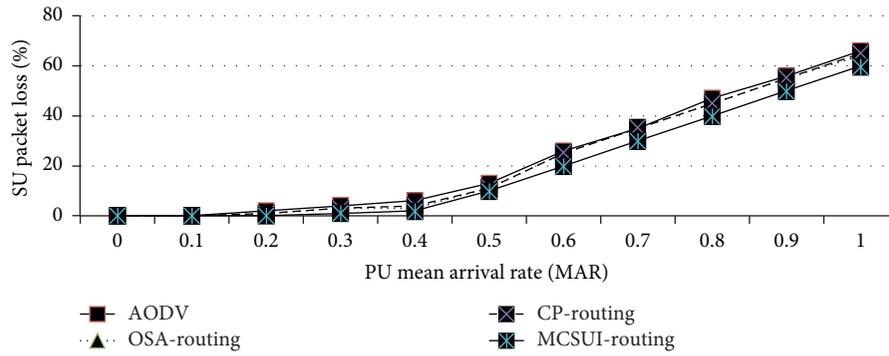


FIGURE 15: Packet loss of SU at PU's sd of MAR as 0.0 (user/ms).

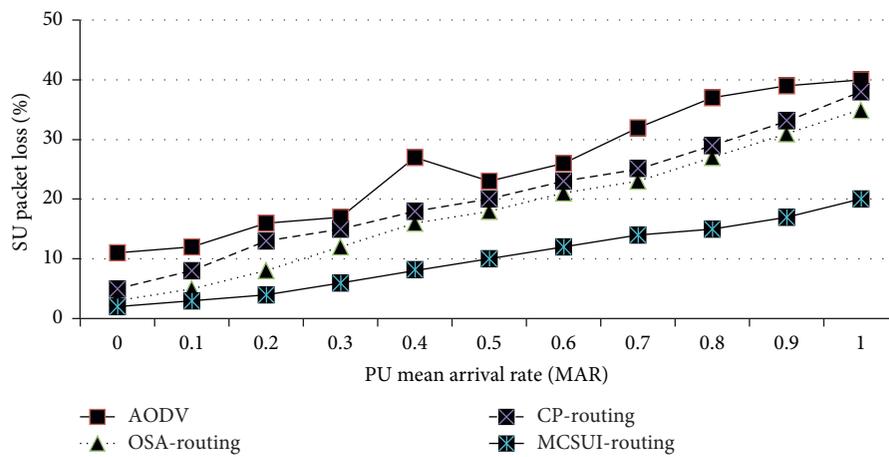


FIGURE 16: Packet loss of SU at PU's sd of MAR = 0.4 (user/ms).

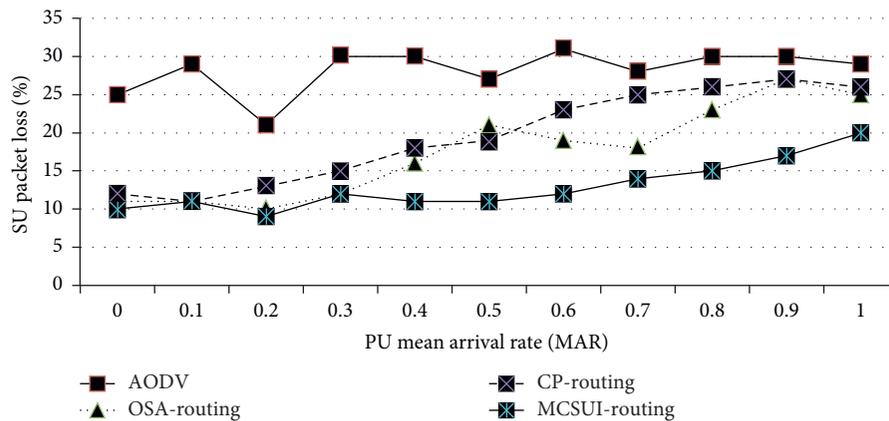


FIGURE 17: Packet loss of SU at PU's sd of MAR = 0.8 (user/ms).

creates routes with more available channel and route choices.

Moreover, we have the following two key observations. Firstly, the fluctuations in EED of the SU can be observed because the routes of AODV and CP routing protocols are static and unaware of the unpredictability of the PU, while the routes of the MCSUI routing protocol are aware of the

channel availability during the routing decision. Secondly, the CP routing protocol and the AODV routing protocol lead to deterioration in the network's routing performance with increments in PUs' activities. Hence, when the MAR of the PU increases, the CP and AODV routing protocols select the longer routes resulting in maximizing of EED for the SU. In contrast, the MCSUI routing protocol minimizes

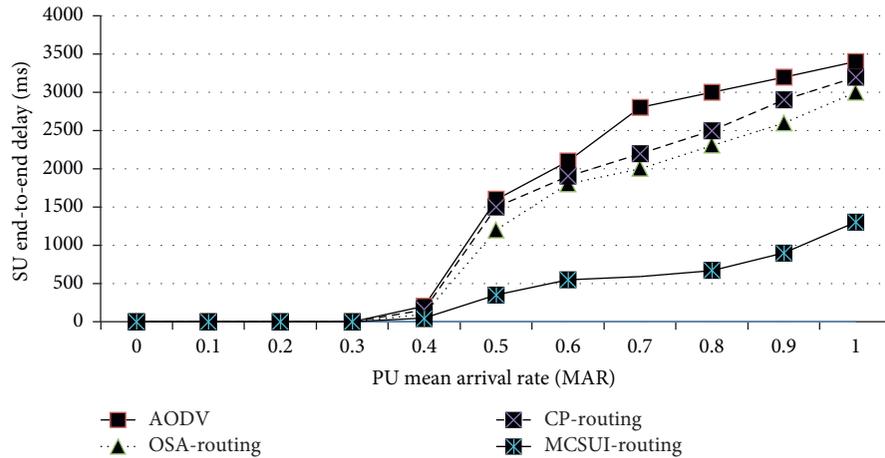


FIGURE 18: EED of SU for PU's sd of MAR as 0.0 (user/ms).

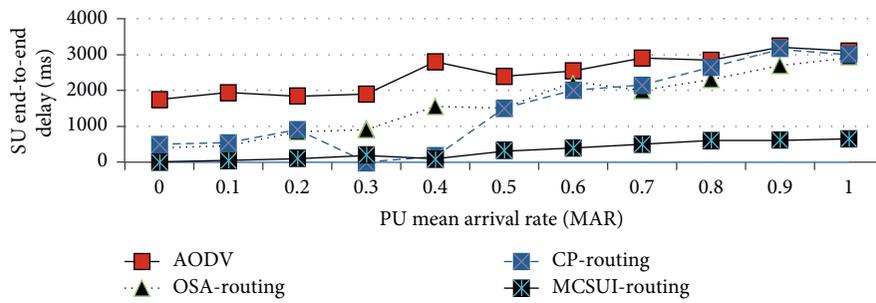


FIGURE 19: EED of SU for PU's sd of MAR as 0.4 (user/ms).

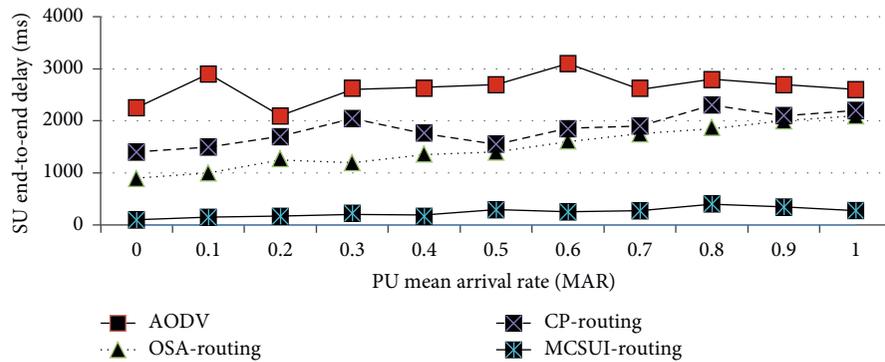


FIGURE 20: EED of SU for PU's sd of MAR as 0.8 (user/ms).

channel-switching events due to user interferences and, hence, selects the shortest routes using the available channel list on the network layer. Generally, the MCSUI routing minimizes the overall EED of SUs in the CRAHN as compared to the other routing protocols.

7. Conclusion

We have enabled the proposed routing protocol to minimize the number of channel switching events, packet collisions due to user interferences, and end-to-end delay

during the transmission. Therein, various Reinforcement Learning- (RL-) based techniques called No-External Regret learning, Q-learning, and Learning Automata are used to Minimize Channel Switching and User Interferences. The overall Quality of Service (QoS) of CRAHN is improved through iterative network state observation of the traditional AODV routing protocol. The user interferences are categorized according to the user characteristics so that the upcoming routing decisions can be based on the channel selection history of PU or SUs over channel selection. Hence, the intraflow interference is minimized by

the implementation of No-External Regret Learning and interflow interference through the Q-Learning. The simulations are carried out with NS-2 environment. Several RL-based routing parameters are applied in the implementation to investigate the performance of the proposed routing protocol. We evaluate the performance of the proposed routing protocol against the various existing AODV, OSA, and CP protocols. We observe that our proposed routing protocol outperforms the existing protocols and achieves good results in terms of the number of channel switching events, packet loss due to user interferences, and end-to-end delay. In future, the efficiency of the proposed routing protocol can be improved using the recent machine learning techniques and the effect of mobile PUs. It is very important to observe the impact of data-aggregation mechanism in conjunction with RL-based routing on energy-efficiency and its effect on performance of proposed MCSUI routing protocol.

Data Availability

All data generated or analysed during this study are included in this published article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Yayasan Universiti Teknologi Petronas, Malaysia, Grant no. 015LC0-029.

References

- [1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *Journal of IEEE on Personal Communications*, vol. 6, no. 4, pp. 13–18, 2002.
- [2] H. Afzal, M. Mufti, A. Raza, and A. Hassan, "Performance analysis of QoS in IoT based cognitive radio Ad Hoc network," *Concurrency and Computation: Practice and Experience*, vol. 32, Article ID e5853, 2020.
- [3] T. D. Le and C. Seong-Gon, "Connectivity analysis of cognitive radio ad-hoc networks with multi-pair primary networks," *Sensors*, vol. 19, no. 3, p. 565, 2019.
- [4] K. Sehgal and S. Markande, "Comparative study of routing protocols in cognitive radio networks," in *Proceedings of the IEEE International Conference on Pervasive Computing (ICPC)*, vol. 6, pp. 1–5, Pune, India, January 2015.
- [5] A. Ali, S. Tariq, M. Iqbal et al., "Adaptive bitrate video transmission over cognitive radio networks using cross layer routing approach," *IEEE Transactions on Cognitive Communications and Networking*, 2020.
- [6] G. Arsany, K. Mohammed, H. Karim et al., "Cooperation-based multi-hop routing protocol for cognitive radio network," *Journal of Network and Computer Applications*, vol. 110, no. 5, pp. 27–42, 2018.
- [7] M. Cesana, F. Cuomo, and E. Ekici, "Routing in cognitive radio networks: Challenges and solutions," *Ad Hoc Networks*, vol. 9, no. 3, pp. 228–248, 2011.
- [8] FCC (Federal Communications Commission), "Spectrum policy task force," Federal Communications Commission, Washington, DC, USA, 2020, https://transition.fcc.gov/sptf/files/SEWGFinalReport_1.pdf Report of the Spectrum Efficiency Working Group: Technical Report.
- [9] R. Kumbhkar, T. Kuber, N. B. Mandayam, and I. Seskar, "Opportunistic spectrum allocation for max-min rate in NC-OFDMA," in *Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 385–391, Stockholm, Sweden, September 2015.
- [10] P. Cheng and S. He, "Observer-based finite-time asynchronous control for a class of hidden Markov jumping systems with conic-type non-linearities," *IET Control Theory & Applications*, vol. 14, no. 2, pp. 244–252, 2020.
- [11] P. Cheng, J. Wang, S. He, X. Luan, and F. Liu, "Observer-based asynchronous fault detection for conic-type nonlinear jumping systems and its application to separately excited DC motor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 3, pp. 951–962, 2020.
- [12] P. Cheng, S. He, J. Cheng, X. Luan, and F. Liu, "Asynchronous output feedback control for a class of conic-type nonlinear hidden Markov jump systems within a finite-time interval," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, 2020.
- [13] L. Duan, J. Huang, and B. Shou, "Duopoly competition in dynamic spectrum leasing and pricing," *IEEE Transactions on Mobile Computing*, vol. 11, no. 11, pp. 1706–1719, 2012.
- [14] Z. Mahdi, A. K. M. M. Islam, B. Sabariah et al., "Medium access control protocols for cognitive radio ad hoc networks: a survey," *Sensors*, vol. 17, no. 9, p. 2136, 2017.
- [15] S. S. Mahsa, E. S. Mohammad, and S. Masoud, "A reinforcement learning based routing in cognitive radio networks for primary users with multi-stage periodicity," *Journal of Wireless Personal Communications*, vol. 101, no. 1, pp. 465–490, 2018.
- [16] R. Li and P. Zhu, "Spectrum allocation strategies based on QoS in cognitive vehicle networks," *IEEE Access*, vol. 8, pp. 99922–99933, 2020.
- [17] L. Singh and N. Dutta, "A novel approach for better QoS in cognitive radio ad hoc networks using Cat optimization," *Data Management, Analytics and Innovation*, Springer, vol. 1042 Berlin, Germany, 2020.
- [18] G. Jakkari, S. Eidenbenz, N. Hengartner, S. V. Krishnamurthy, and M. Faloutsos, "Link positions matter: a noncommutative routing metric for wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 61–72, 2012.
- [19] R. E. Tuggle, "Cognitive multipath routing for mission critical multi-hop wireless networks," in *Proceedings of the IEEE 42nd Southeastern Symposium on System Theory (SSST)*, Tyler, TX, USA, March 2010.
- [20] M. Radi, B. Dezfouli, K. Abu Bakar, S. Abd Razak, and M. A. Nematbakhsh, "Interference-aware multipath routing protocol for QoS improvement in event-driven wireless sensor networks," *Tsinghua Science and Technology*, vol. 16, no. 5, pp. 475–490, 2011.
- [21] A. S. Cacciapuoti, M. Caleffi, and L. Paura, "Reactive routing for mobile ad hoc networks," *Ad Hoc Networks*, vol. 10, no. 5, pp. 803–815, 2012.
- [22] L. Indhumathi and R. Vadivel, "Adaptive delay tolerant routing protocol (ADTRP) for cognitive radio mobile ad hoc networks," *International Journal of Computer Applications*, vol. 128, no. 6, pp. 19–24, 2015.
- [23] A. L. Nisar, A. B. Altaf, G. M. Mir, and R. A. Simnani, "Quality of service provisioning in cognitive radio network," *Orient*

- Journal of Computer Science & Technology*, vol. 10, no. 4, pp. 780–787, 2017.
- [24] W. Y. Lee and I. F. Akyildiz, “Optimal spectrum sensing framework for cognitive radio networks,” *Journal of IEEE Transactions on Wireless Communication*, vol. 7, no. 10, pp. 3845–3857, 2008.
- [25] D. Niyato, E. Hossain, and P. Wang, “Optimal channel access management with QoS support for cognitive vehicular networks,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 573–591, 2011.
- [26] X. Jin, R. Zhang, J. Sun, and Y. Zhang, “TIGHT: a geographic routing protocol for cognitive radio mobile ad hoc networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4670–4681, 2014.
- [27] F. Z. Benidris, B. Benmammam, and F. T. Bendimerad, “An efficient spectrum allocation mechanism for cognitive radio networks,” *Journal of IEEE Transactions on Communications*, vol. 13, no. 7, pp. 534–547, 2014.
- [28] L. Gul, X. Zhong, and S. Zhou, “Traffic assignment algorithm for multi-path routing in cognitive radio ad hoc networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, April 2013.
- [29] J. Chen, H. Li, and J. Wu, “WHAT: A novel routing metric for multi-hop cognitive wireless networks,” in *Proceedings of the 19th IEEE Annual Wireless and Optical Communications Conference*, Shanghai, China, May 2010.
- [30] J. Jia, J. Zhang, and Q. Zhang, “Cooperative relay for cognitive radio networks,” in *Proceedings of the IEEE INFOCOM*, pp. 2304–2312, Turin, Italy, April 2009.
- [31] R. Biswas and J. Wu, “Minimizing the number of channel switches of mobile users in cognitive radio ad-hoc networks,” *Journal of Sensor and Actuator Networks*, vol. 9, no. 2, p. 23, 2020.
- [32] S. AlQahtani and A. Alotaibi, “A route stability-based multipath QoS routing protocol in cognitive radio ad hoc networks,” *Wireless Networks*, vol. 25, no. 5, pp. 2931–2951, 2019.
- [33] Y. Yao, A. Popescu, and A. Popescu, “On prioritised opportunistic spectrum access in cognitive radio cellular networks,” *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 2, pp. 294–310, 2016.
- [34] X. Huang, D. Lu, P. Li, and Y. Fang, “Coolest path: spectrum mobility aware routing metrics in cognitive ad hoc networks,” in *Proceeding of the 31st International Conference of IEEE on Distributed Computing Systems (ICDCS)*, Minneapolis, MN, USA, June 2011.
- [35] C. Perkins and E. Belding-Royer, “Ad hoc on-demand distance vector (AODV) routing protocol,” The Internet Society, Reston, VA, USA, 2003, <http://www.ietf.org/rfc/rfc3561.txt> Request for Comments 3561 in Internet Engineering Task Force.
- [36] G. E. Box and M. E. Muller, “Multi-agent learning using a variable learning rate,” *Artificial Intelligence*, vol. 136, no. 2, pp. 215–250, 2002.