IMPROVED Q-REINFORCEMENT LEARNING BASED OPTIMAL CHANNEL SELECTION IN COGNITIVE RADIO NETWORKS

Sopan Talekar, Satish Banait and Mithun Patil

[1]Assistant Professor, Department of Computer Engineering, MVPS's KBT College of Engineering, Nashik, Maharashtra, India
[2]Assistant Professor, Department of Computer Engineering, K.K. Wagh Institute of Engineering Education & Research, Nashik, Maharashtra, India
[3]Associate Professor, Department of Computer Science and Engineering, N.K. Orchid College of Engineering & Technology, Solapur, Maharashtra, India

ABSTRACT

Cognitive Radio Networks are an emerging technology in for wireless communication. With increasing number of wireless devices in wireless communication, there is a shortage of spectrum. Also, due to the static allocation of channels in wireless networks, there is a scarcity of spectrum underutilization. For efficient spectrum utilization, secondary users dynamically select the free channel of primary users for the transmission of packets. In this work, the performance of routing in a cognitive radio network is improved by the decision of optimal channel selection. The aim of this work is to maximize the throughput and reduce the end-to-end delay. Therefore, an Improved Q-Reinforcement learning algorithm is proposed for the optimal channel selection during the packet routing between source and destination. The performance of this work is compared with the existing routing protocols. It is simulated in network simulator-2 (NS2) with Cognitive Radio Cognitive Network (CRCN) simulation. After performance evaluation, it is observed that the proposed work performs better than existing work with respect to packet delivery ratio, throughput, delay, jitter, control overhead, call blocking probability, packet dropping ratio, good put and normalized routing overhead.

KEYWORDS

Cognitive Radio Network, Reinforcement Learning, Routing Protocols, Channel selection, Throughput maximization.

1. INTRODUCTION

The Cognitive radio (CR) concept is introduced by Joseph Mitola III [1]. It is based on the software Defined Radio (SDR) [2]. In cognitive radio networks, it is necessary to decide the common control channel for the communication [3]. A multichannel MAC protocol is proposed to solve the problem of multichannel hidden terminal [4]. To perform efficient spectrum utilization, CR plays an essential role in wireless communication. When unlicensed spectrums are not sufficient due to the increasing number of wireless users, CR technology fulfills the needs of the spectrum to the unlicensed users. In this, unlicensed users are also called secondary users (SU) or cognitive users (CU) that utilize the licensed spectrum when license users also called as primary users (PU) not using it. When primary users want a channel for the data transmission then secondary users need to immediately release the channel and switch to another channel for the remaining data transmission. Channel switching will be more due to the irregularity of

DOI: 10.5121/ijcnc.2023.15301

primary users on the channel and hence results in delay for data transmission. Due to this dynamicity of channel availability, it is necessary to improve the performance of routing in cognitive radio networks. Therefore, it is needed to design a channel selection strategy that considers the path with minimum channel switching and interference, data transmission rate, and data transmission delay. In the proposed work, Improved Q-Reinforcement learning-based channel selection (IQRLC) is designed to perform the routing in cognitive radio networks.

The organization of the paper is as follows. Section II includes related work on cognitive radio networks. Section III describes IQRLC design and implementation. Section IV discusses the performance evaluation of the IQRLC routing protocol. Finally, Section V concludes the paper.

2. Related Work

An optimal channel selection plays an important role to improve the performance of routing in CRN. To satisfy this objective, many routing protocols are designed in CRN. Existing MANET routing protocols don’t perform well in the dynamic environment of cognitive radio. Therefore, by considering the environment of CR, researchers have developed routing protocols based on the AODV and DSR routing protocols. Mostly, channel selection strategy is designed during the design of routing protocols for cognitive radio networks. To tackle the dynamic environment of cognitive radio, machine learning-based channel selection strategy, and routing are performed in CRNs.

Many routing protocols are designed in cognitive radio networks without using any machine learning approach. Graph Coloring problem is used for channel selection while designing the routing protocol. This approach is not suitable for the highly dynamic environment of cognitive radio networks [5]. In SAMER [6], the path is determined with higher spectrum availability. But in this, optimal channel selection is not performed. CR-AODV is a modification of the basic AODV routing protocol. In this, the WCETT routing metric is used to perform the routing in cognitive radio networks without using any optimal channel selection approach [7]. In CSRP [8] channel selection is performed based on historical information only that may not be sufficient to select an optimal channel in a dynamic environment. JRCS [9], uses a dynamic programming approach for channel selection. It requires more memory to store the results of sub-problems that may or may not be utilized in the future. Proposed-PSM [10] uses path stability metrics to select the best and most stable path for data transmission. It uses a naive Bayes and decision tree approach to learning the decision about sending the data on the path from the available data such as destination, time slot, neigh boring node, and available channels. From the labeled data, it labels the unlabeled data in the form of Yes or No (whether to decide on data transmission on the path).

RL (Reinforcement Learning) is used to determine the best path. RL is used to learn the selection of the best next node for the data transmission. A Current node represents the state, the selection of the next node represents action and for each action it receives a reward. Mostly the Q-learning approach of RL is explored to determine the optimal path for data transmission. Q-learning is used to improve the routing performance in cognitive radio networks. DRQ uses a dual reinforcement learning approach to select the next hop for the routing [11]. CRQ uses combined metrics such as channel availability, PU interference, and channel quality for the next hop selection during the routing process [12]. WCRQ uses a weight function to trade off the performance of PU and SU [13]. PQR considers the periodic activity of PUs to determine the stable path. It outperforms CRQ in terms of throughput and interference [14].

In this, SMART [15] and Proposed 2017 [16] are designed based on clustering and RL approach to improving the performance of cognitive radio networks. Clusters of SUs are formed based on the minimum common channels available and RL is used to perform routing decisions. Clusters are maintained in terms of splitting and merging in the Proposed 2017 [16]. The Actor-Critic model of Reinforcement Learning is used to improve the routing performance in CRN. In this, the optimal policy is determined by the actor-critic model to perform the channel selection in CRN [17]. In [18], impact of variable packet size on routing performance in cognitive radio networks is evaluated. MCSUI [19] performs joint channel selection and routing which minimizes channel switching and user interferences in cognitive radio networks. In [20], particle swarm optimization is used to optimize the channel allocation when multiple secondary users listen to the channel at the same time, and the best cooperative channel selection is performed based on Q-reinforcement learning. In [21], an Extended Generalized Predictive Channel Selection Algorithm is proposed to increase the throughput and reduce the delay of secondary users. In [22], two methods such as Secure CR and Optimal CR based on multi-criteria decision-making (MCDM) are proposed. From the results, it is observed that/ Secure CR is more efficient than Optimal CR in response to time but Optimal CR gives better results than Secure CR. In [23], the proposed method (RLCLD) improve the performance of routing by 30% more than the conventional AODV routing protocol. This method uses cross -layering with Q- Reinforcement learning to improve the routing performance in the industrial internet of Ad-hoc sensor networks.

3. DESIGN AND IMPLEMENTATION OF THE PROPOSED WORK

An improved Q-Reinforcement Learning in cognitive radio networks (IQRLC) reduces the space of values and takes less time to make decisions than conventional Q-reinforcement learning. During the learning process, IQRLC automatically finds useful sub-goals. It creates abstractions that find out the undesirable regions of state space and focuses on useful regions of state space with a maximum rewarding environment.

The channel selection process of the proposed work IQRLC is shown in Figure 1. All steps of this process are described in detail as follows:

Step: 1 Network model and channel database creation

In the network model primary users (PU), secondary users (SU),and base stations (BS) are created. All are located with their inbuilt GPS technology. Primary users announce their location periodically & accordingly all secondary users determine and update their distance from PUs. The channel database stores information about nodes and available channels. This database continually updates for the selection of channels by the node for the data transmission. Cognitive users perform the spectrum sensing and update the channel list periodically.

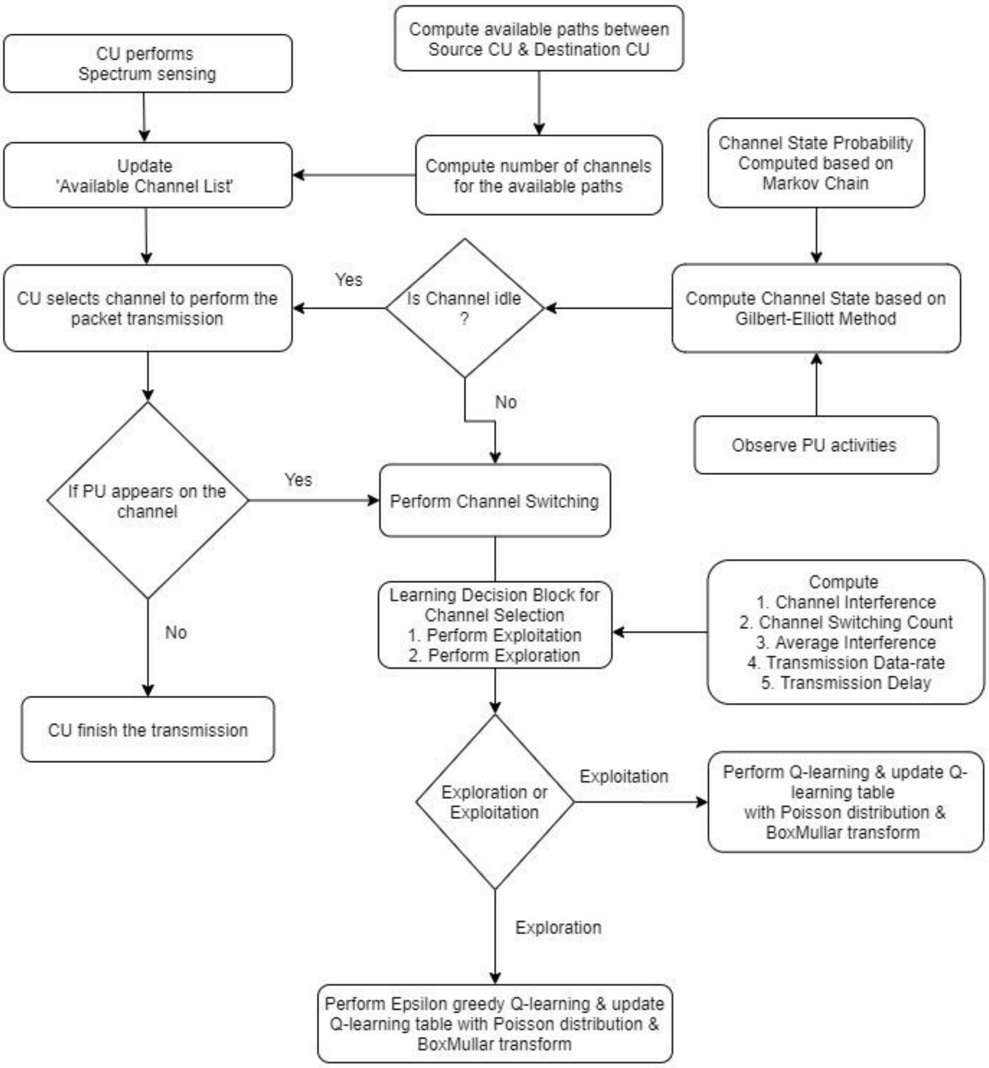
CU performs Spectrum sensing
Compute available paths between Source CU & Destination CU
Update 'Available Channel List'
Compute number of channels for the available paths
Channel State Probability Computed based on Markov Chain
CU selects channel to perform the packet transmission
Yes
Is Channel idle ?
Compute Channel State based on Gilbert-Elliott Method
No
Observe PU activities
If PU appears on the channel
Yes
Perform Channel Switching
Learning Decision Block for Channel Selection 1. Perform Exploitation 2. Perform Exploration
Compute 1. Channel Interference 2. Channel Switching Count 3. Average Interference 4. Transmission Data-rate 5. Transmission Delay
No
CU finish the transmission
Exploration or Exploitation
Exploitation
Perform Q-learning & update Q-learning table with Poisson distribution & BoxMullar transform
Exploration
Perform Epsilon greedy Q-learning & update Q-learning table with Poisson distribution & BoxMullar transform

Figure 1. Channel selection process for routing in IQRLC

Step: 2 Compute the number of available channels for the available paths

Source CU determines available paths to the destination. For the available paths, it computes the number of available channels and accordingly updates the channel list.

Step: 3 Compute channel state

Gilbert Elliot's method is used to estimate the channel state from the activities of PUs. Channel state probability is computed using the Markov chain method. After channel state computation, the following two cases discussed further action.

Case: 1

If the channel state is idle then CU selects channel for the packet transmission

Else

Perform channel switching

Case: 2

If PUs appears on the selected channel then perform the channel switching

Else

Finish the transmission

Step: 4 Perform channel switching

When PU appears on the channel or the selected channel is busy then it is needed to perform the channel switching. CU switches their communication to another channel that is selected from the exploitation or exploration. Channel quality parameters such as channel interference, channel switching count, average interference, transmission data rate, and transmission delay are considered while updating the Q-learning table after performing the exploitation and exploration. Initially, CU performs the channel selection with reward 1 based on the minimum channel switching count; otherwise it explores the channel selection using the epsilon greedy Q-learning algorithm. Epsilon greedy Q-learning algorithm for channel selection is discussed below.

a. Epsilon Greedy Q learning algorithm for channel selection

In the epsilon greedy Q learning method, an agent performs exploitation to take benefit of existing knowledge. Based on the existing knowledge, it selects channel to perform the routing in cognitive radio networks. In addition, it performs exploration to derive new knowledge about the channel selection. This approach selects the channel with the maximum reward. Therefore, it is needed to trade off between exploration and exploitation. With a small value of epsilon, it performs the exploration and with a large value of epsilon it performs the exploitation.

Table 1. Algorithm 1: Epsilon Greedy Q learning algorithm for channel selection

```
Input:  Learning parameters = $\alpha$, $\gamma$;
S = State, A = Action and R = Reward;
# for each step in each period determine and update Q value
For each episode do
Initialize state S;
For each step in episode do
A = Action-Channel Selection (Q, S, $\epsilon$)
$Q (S, A) = Q (S, A) + \alpha [R + \gamma Max_a Q(S', a) - Q (S, A)]$
# Update Q learning table
S = S';
While S is not terminal
End
End
```

Table 2. Algorithm 2: Epsilon Greedy Action-Channel Selection Function

```
Input:  Generation of Q table
          Є = small number
            S = State
            N = uniform random number (0, 1)
Function: Action-Channel Selection (Q, S, Є)
            If N < Є then
A = select channel randomly
Else
A = select channel with Max Q value
End
Return Action-Channel Selection (A)
End
```

b. Computation of Channel Quality Parameters

The Q-learning table is updated with Q-learning values, learning state, learning activities, and some channel quality parameters such as interference, transmission data rate ,and transmission delay after performing the epsilon greedy Q-learning algorithm. Poisson distribution and BoxMullar transform are used to determine and update the learning values (LV).
Transmission data rate and delay are computed based on the channel interference. It is computed as below.

If (Average interference – Minimum interference) < Interference range then

- Interference = (Average interference – Minimum interference)
- Transmission Data rate = (Channel Bandwidth-Bandwidth_Node) * (1- Interference)

Transmission Delay = Channel Bandwidth/ Transmission data rate * 8

c. Perform Q-learning

Learning state & learning action value are computed below.

- Learning State = Transmission Data rate / (Transmission Delay + Interference + 1);
- Learning Action = (reward == 1)?1:((penalty == 1)?2:0);

Update Q _Learning as per action. Channel reward and penalty updated as per the action.

d. Compute the Poisson and Box-Muller transforms:

- Poisson = Expo (-Learning State) * Pow (Learning State, Learning Action) / Fact (Learning Action)
- BoxMullar = (Learning State)^2 + (Poisson)^2
- Reward Value = (Learning Action == 1)? Random:: Uniform (0, 1):Random Uniform (-1, 0)

e. Compute Q_Final Value for the learning:

- $Q_{a\_}$= Initialization of q value
- Q_a = α * (Reward Value – $Q_{a\_}$)

- Q_{ai}= Transmission Delay * α * (Reward Value * (Learn Interval + Transmission Delay) - Q_a)
- Q_{at}= C_1 * Exp (-α * Learn Interval) + Reward Value
- Sum_Value = ∑Learning Action * Reward Value
- Explore Value = Sum_Value
- Q_{at}= C_1 * Exp (-α * Learn Interval) + Explore Value
- CS_Action = α * (Explore Value * Learn Interval - Q_{at})
- Q = (1- α)* CS_Action + α * Q_{ai}
- Q_2 = α * CS_Action + (1- α) * Q_{ai}

Channel State is computed. Channel parameters are computed as per channel states using Q learning

- Pi_t = Max (Max (Max($Q_{a\_}$, Q_a), Q_{ai}), Q_{at})
- Q_Final = (1- α) * Q + α * Pi_t
- Li = $Q_{a\_}$
- Lp = (Li + Exp (-Learning State) / Learning State) * Q_Final
- $\boldsymbol{\gamma}$ = random:: uniform (0, 1)
- Pi_t = Max (Pi_t, Explore Value * Pow ($\boldsymbol{\gamma}$, Learn Interval -1) * Reward Value

If the current channel state is Busy then R = 0 else R =1. Based on the status of the current state, update learning values (Li).

- Li = Explore Value * Pow (R + $\boldsymbol{\gamma}$ * Max (Max (Max ($Q_{a\_}$, Q_a), Q_{ai}), Q_{at}) – Q), 2)
- Li_2 = Explore Value * Pow (R + $\boldsymbol{\gamma}$ * Max (Max (Max ($Q_{a\_}$, Q_a), Q_{ai}), Q_{at}) – Q_2), 2)
- delLi_1 = Explore Value * (R + $\boldsymbol{\gamma}$ * Max (Max (Max ($Q_{a\_}$, Q_a), Q_{ai}), Q_{at}) – Q) * delLi * Li
- delLi_2 = Explore Value * (R + $\boldsymbol{\gamma}$ * Max (Max (Max ($Q_{a\_}$, Q_a), Q_{ai}), Q_{at}) – Q_2) * delLi * Li_2
- $\boldsymbol{\theta}_1$ = Q + β * delLi_1
- $\boldsymbol{\theta}_2$ = Q_2 + β * delLi_2

f. Computation of channel action values based on epsilon greedy Q-learning algorithm

Channel action values are computed based on an epsilon greedy Q-learning algorithm. It is computed as below.

Table 3. Computation of channel action value

```
Let initialize a = 0 & g = 0, Є = random ∷ uniform (0, 1) and Є2 = 0
If random ∷ uniform (0, 1) < Є then
        a = Choose action via exploration. Explore all states to select action.
        # Select random action to explore.
 Else
        a = Max(Max(Max($Q_{a\_}$, $Q_a$), $Q_{ai}$), $Q_{at}$)
        #Choose action via exploitation.
        #Select the action with Max value (Future Reward)

Let compute d1 = a * $\boldsymbol{\theta}_1$;    d2 = a * $\boldsymbol{\theta}_2$;
e = (d1+d2)/(Max(d1, d2)* Explore Value)
R = R + e
```

```
If random:: uniform (0, 1) < Є2
    g = choose random action (explore)
    # Random::uniform(states[scount-1]);
Else
    g = Max(Max(Max(Max(Max($Q_{a\_}$, $Q_a$), $Q_{ai}$), $Q_{at}$), R), Max($\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$))
```

g. Update Q learning Table:

After computing channel action values a & g, the Q-learning table will be updated with channel id, interference, channel data rate, transmission delay, learning state, learning activities and learning values (LV). Box-Muller and Poisson's distributions are used to update LV. It is computed as below.

- BoxMullar Distributions (LV) = Min (Max (a, g), Max (BoxMullar, Min (Q_final, LP)))

Poisson (LV) = Min (Max (a, g), Max (Poisson, Min (Q_final, LP)))

4. SIMULATION AND PERFORMANCE EVALUATION

The performance of the proposed routing protocol (IQRLC) is evaluated by different performance metrics such as throughput, packet delivery ratio, delay, control overhead, good put, normalized routing overhead, call blocking probability, jitter, and packet dropping ratio. Network Simulator NS2 with CRCN patch is used to evaluate the performance of the proposed routing protocol. The performance analysis of the proposed routing protocol IQRLC is compared with the existing routing protocol CSRC [17], LCRP [15] ,and CRP and it is discussed below.

4.1. Simulation Parameters

Simulation parameters for the proposed work are shown in Table4.

Table 4. Simulation Parameters

Parameters	Values
Routing Protocols	IQRLC, CSRC, LCRP, CRP
Medium Access Control	802.11
Simulation area	600m x500m
Number of nodes	50
Number of PU	15
Mobility Model	Random Way Point
Traffic model	Constant Bit Rate (CBR)
Pathloss model	Two Ray
Energy Model	Initial Energy:100 Joules
Packet Size in bytes	1000

4.2. Performance Evaluation

The proposed IQRLC protocol is evaluated with different routing metrics and it is compared with CSRC[20], LCRP and CRP routing protocols [17]. In LCRP, the optimal policy for channels is derived based on the policy iteration method. It selects the maximum policy value with the maximum reward. In CRP, it is modified based on basic AODV routing protocol [24]. It selects the channel with a long idle period for data transmission. IQRLC is proposed to overcome the limitation of scalability and trade-off between exploitation & exploration. It explores all the states for an optimal channel selection in a cognitive radio network to make a balance between exploitation and exploration with an epsilon greedy approach.

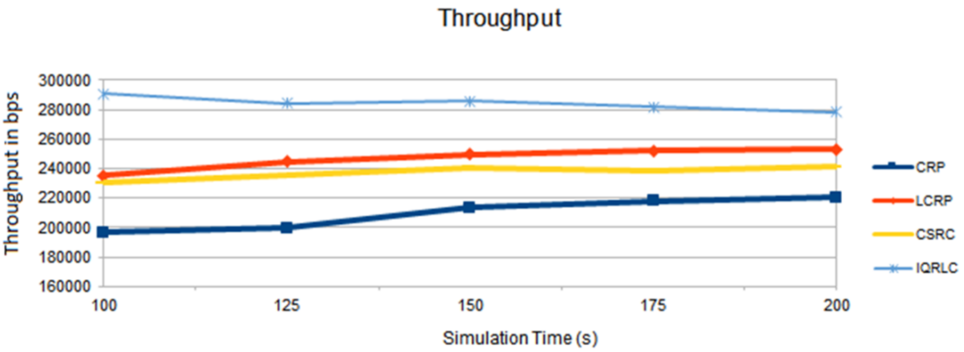

Figure 2. Simulation Time Vs. Throughput

The simulation result for throughput is shown in Figure 2. The throughput of proposed work IQRLC is better than LCRP and CRP routing protocols. IQRLC has a better channel selection strategy than LCRP and CRP. Throughput of LCRP is better than CRP as it uses a reinforcement learning-based channel selection approach for routing in CRNs. The average throughput in IQRLC is 16.45% more than CSRC, 13.07 % more than LCRP, and 26.22 % more than CRP

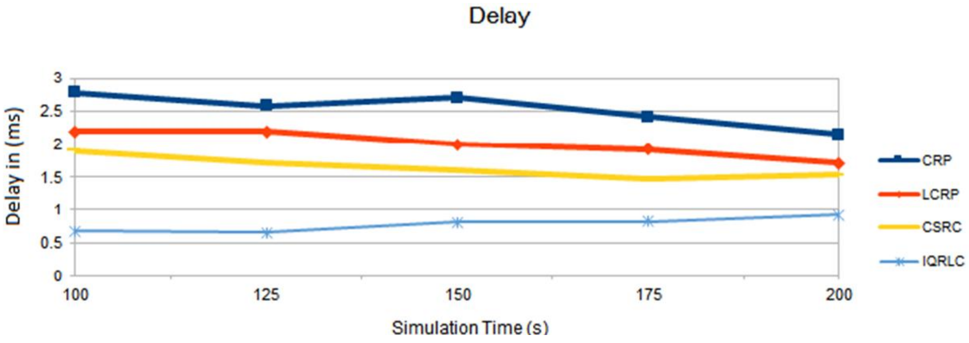

Figure 3. Simulation Time Vs. Delay

The simulation result for the delay is shown in Figure 3. It shows that /the delay in IQRLC is less than LCRP and CRP. In IQRLC, it selects the path with minimum channel switching which results in less delay to transmit the packets. The channel switching parameter is not taken into consideration while determining the path in both LCRP and CRP. Therefore, the delay is more in both LCRP and CRP routing protocols compared to IQRLC routing protocol. Delay in IQRLC is 51.57 % less than CSRC, 60.04% less than LCRP, and 68.32% less than CRP.

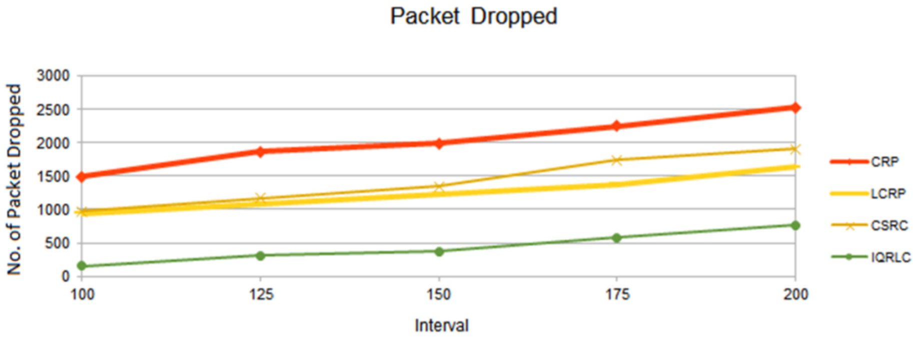

Figure 4. Simulation Time Vs. Packet dropping ratio

Packet dropping ratio for the proposed routing protocol IQRLC is shown in Figure 4. It is observed that CRP has a higher packet- dropping ratio compared to LCRP and IQRLC. Average packet-dropping ratio in IQRLC is 15.00% less than CSRC, 12.50% less than LCRP, and 24.51% fewer than CRP. IQRLC selects less interference channels with an epsilon greedy approach that results in less packet drop during transmission.

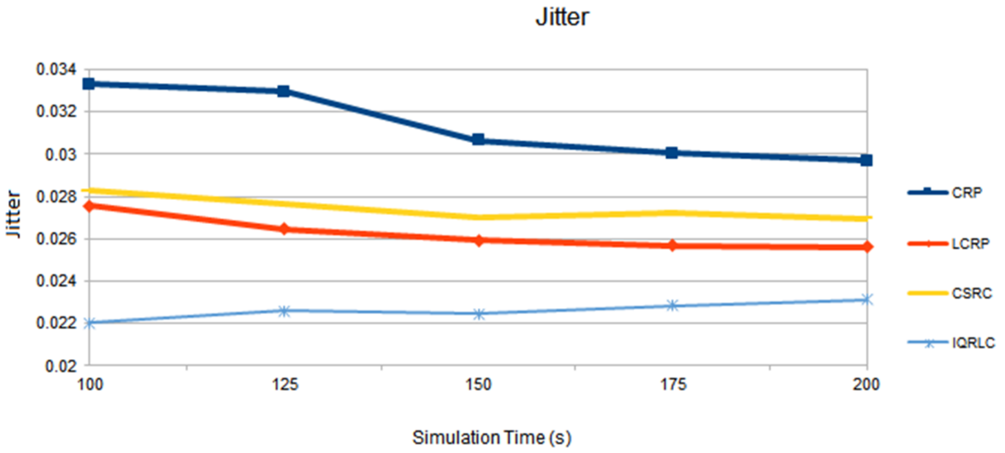

Figure 5. Simulation Time Vs. Jitter

Also, less jitter is observed in the proposed IQRLC compared to LCRP and CRP. The simulation result for the jitter is shown in Figure 5. Average jitter in IQRLC is 17.50 % less than CSRC, 13.77% less than LCRP ,and 27.62% less than CRP.

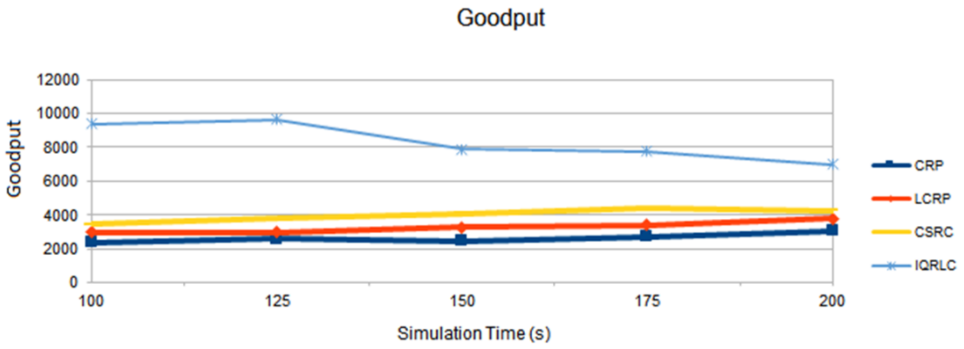

Figure 6. Simulation Time Vs. Goodput

Due to the minimum number of channel switching and optimal channel selection approach, good put in IQRLC is better than LCRP and CRP. It is shown in Figure 6. IQRLC keeps all the available paths between the source and destination. Therefore, it immediately switches to another available path in the situation of route failure. The average good put in IQRLC is 50.91% more than CSRC, 59.63% more than LCRP, and 67.54% more than CRP.

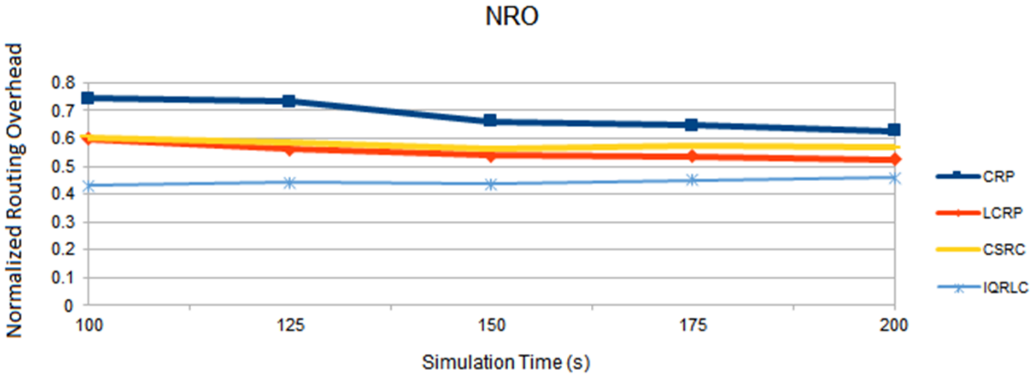

Figure 7. Simulation Time Vs. NRO

The simulation result for normalized routing overhead (NRO) is shown in Figure 7. It shows that IQRLC has less NRO than existing LCRP and CRP routing protocols. It is 19.40% less than LCRP, 23.23% less than CSRC and 34.61% less than CRP. Secondary users (SU) in IQRLC work with multiple paths and select the best channel for the packet transmission that would not be or less interference by the primary users (PU).

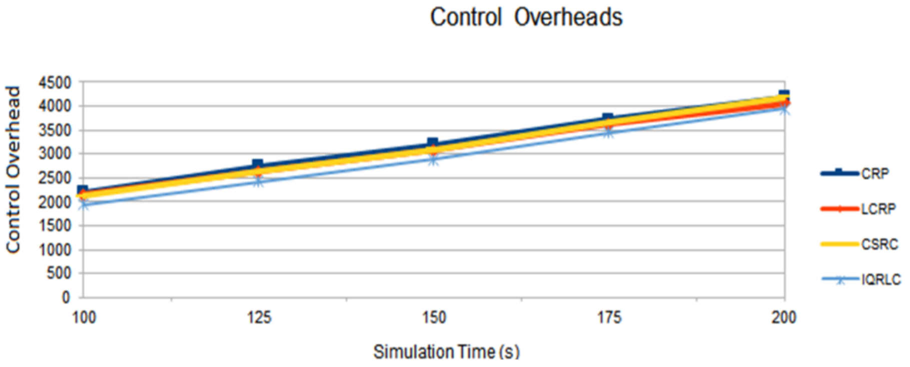

Figure 8. Simulation Time Vs. Control overhead

Similarly, control overhead in IQRLC is less than in LCRP and CRP. It is shown in Figure 8. Average control overhead in IQRLC is 6.8% less than CSRC, 6.38% less than LCRP, and 9.26% less than CRP.

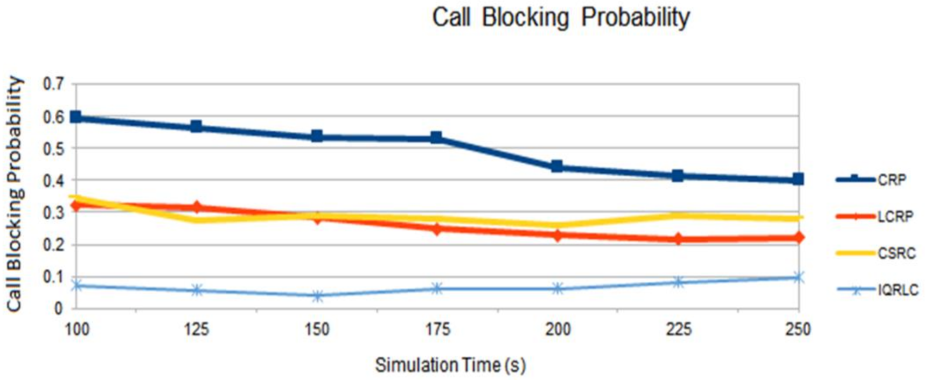

Figure 9. Simulation Time Vs. CBP

Call blocking probability (CBP) is the ratio of a number of blocked users to the total number of users. The Simulation result for call blocking probability (CBP) is shown in Figure 9. It is observed that/ CBP in IQRLC is less than LCRP and CRP. The Average CBP in IQRLC is 73.22% less than LCRP, 76.60% less than CSRC and 85.81 less than CRP. Comparison of the performance of proposed work IQRLC with the existing routing protocols shown in table 5.

Table 5. Comparison of IQRLC performance with existing routing protocols

Performance Parameters	**Observations**
Throughput	IQRLC throughput is 16.45% more than CSRC, 13.07 % more than LCRP and 26.22 % more than CRP.
Delay	IQRLC delay is 51.57 % less than CSRC, 60.04% less than LCRP and 68.32% less than CRP.
Packet Dropped	IQRLC packet dropped is 15.00% less than CSRC, 12.50% less than LCRP and 24.51% less than CRP.
Jitter	IQRLC jitter is 17.50 % less than CSRC, 13.77% less than LCRP and 27.62% less than CRP.
Goodput	IQRLC goodput is 50.91% more than CSRC, 59.63% more than LCRP and 67.54% more than CRP.
Normalized Routing Overhead (NRO)	IQRLC NRO is 19.40% less than LCRP, 23.23% less than CSRC and 34.61% less than CRP.
Control Overhead	IQRLC control overhead is 6.8% less than CSRC, 6.38% less than LCRP and 9.26% less than CRP.
Control Blocking Probability (CBP)	IQRLC CBP is 73.22% less than LCRP, 76.60% less than CSRC and 85.81 less than CRP.

5. CONCLUSIONS

The proposed work for channel selection and routing is designed based on improved Q-Reinforcement learning. This method uses an epsilon greedy approach to exploit and explore the channel based on Q-learning and reward values. An optimal channel is selected based on different channel quality parameters such as transmission data rate, delay, channel switching count, and average interference. It is observed that the proposed method IQRLC performs better in the dynamic environment of cognitive radio networks. It improves the routing performance in terms of different routing metrics such as throughput, delay, packet delivery ratio, jitter, etc. From the simulation results, it is shown that IQRLCoutper forms existing routing protocols

CSRC, LCRP ,and CRP. Finally, it is concluded that optimal channel selection plays an essential role to improve the performance of routing in cognitive radio networks.

As a future scope, deep Q- Reinforcement learning based channel selection will be performed to improve the performance of cognitive radio networks. In this, simulation results for the different routing performance parameters against various parameters such as time, number of nodes ,and speed of nodes will be performed.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] J. Mitola and G. Maguire, “Cognitive radio: making software radios more personal”, Personal Communication, IEEE, vol. 6, no. 4, pp. 13-18, August 1999.

[2] SDRF Cognitive Radio Definitions, SDRF-06-R-0011-V1.0.0, Approved Nov 2007. [Online].Available: http://www.sdrforum.org.

[3] Ijjeh, Abdallah. “Multi-Hop Distributed Coordination in Cognitive Radio Networks,” International journal of Computer Networks & Communications. 7. 137-145. 10.5121/ijcnc.2015.7211.

[4] Kamruzzaman, Sikder. “CR-MAC: A multichannel MAC protocol for cognitive radio ad hoc networks,” International journal of Computer Networks & Communications. 2. 10.5121/ijcnc.2010.2501.

[5] A. Sampath, L. Yang, L. Cao, H. Zheng and B. Zhao, “High throughput spectrum-aware routing for cognitive radio networks,” Proceedings of IEEE international conference on cognitive radio oriented wireless networks (CrownCom); 2007.

[6] I.Pefkianakis, S.H.Y.Wong and S. Lu, “SAMER:Spectrum aware mesh routing in cognitive radio network”, in Proc. of IEEE DySPAN, USA, pp. 766–770, October 2008.

[7] A. Chehata, W. Ajib and H. Elbiaze, “An on-demand routing protocol for multi-hop multi-radio multi-channel cognitive radio networks”, in Proceedings of the IFIP Wireless Days Conference 2011, Niagara Falls, Canada, pp. 1-5, October 2011.

[8] Y. Wang, G. Zheng, H. Ma, Y. Li and J. Li, “A joint channel selection and routing protocol for cognitive radio network”, Wireless Communication and Mobile Computing, vol. 2018, Article ID 6848641, 7 pages, March 2018.

[9] Mumey B, Tang J, Judson IR, Stevens D. On routing and channel selection in cognitive radiomesh networks. IEEE Trans VehTechnol 2012;61(9):4118–28.

[10] ZilongJin, Donghai Guan, Jinsung Cho and Ben Lee, “A Routing Algorithm based on Semi-supervised Learning for Cognitive Radio Sensor Networks” International Conference on Sensor Networks(SENSORNETS 2014), pp.188-194.

[11] Xia, Bing &Wahab, M.H. & Yang, Yang & Fan, Zhong&Sooriyabandara, Mahesh. (2009). Reinforcement learning based spectrum-aware routing in multi-hop cognitive radio networks. Proceedings of the 4th International Conference on CROWNCOM. 1 - 5. doi:10.1109/CROWNCOM.2009.5189189.

[12] H. A. A. Al-Rawi, K. A. Yau, H. Mohamad, N. Ramli and W. Hashim, "A reinforcement learning-based routing scheme for cognitive radio ad hoc networks," 2014 7th IFIP Wireless and Mobile Networking Conference (WMNC), Vilamoura, 2014, pp. 1-8. doi:10.1109/WMNC.2014.6878881

[13] Hasan A. A. Al-Rawi, Kok-Lim Alvin Yau, Hafizal Mohamad, NordinRamli, and WahidahHashim, Reinforcement Learning for Routing in Cognitive Radio Ad Hoc Networks, The Scientific World Journal Volume 2014, Article ID 960584, 22 pages http://dx.doi.org/10.1155/2014/960584

[14] MahsaSoheilShamaee, Mohammad Ebrahim Shiri, MasoudSabaei, A Reinforcement Learning Based Routing in Cognitive Radio Networks for Primary Users with Multi-stage Periodicity, Wireless PersCommun (2018) 101, pp. 465-490 https://doi.org/10.1007/s11277-018-5700-y

[15] Saleem, Yasir and Yau, Alvin Kok-Lim and Hafizal Mohamad, and Nordin, Ramli and Rehmani, Mubashir Husain (2015) SMART: A SpectruM-Aware clusteR-based rouTing scheme for distributed cognitive radio networks. Computer Networks, 91. pp. 196-224. ISSN 1389-1286

[16] Saleem, Yasir&Yau, Kok-Lim & Mohamad, Hafizal&Ramli, Nordin&Rehmani, Mubashir Husain & Ni, Qiang. (2017). Clustering and Reinforcement-Learning-Based Routing for Cognitive Radio Networks. IEEE Wireless Communications. 24. 146-151. 10.1109/MWC.2017.1600117.

[17] S. Talekar and S. Terdal. Reinforcement Learning Based Channel Selection for Design of Routing Protocol in Cognitive Radio Network, 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), Bengaluru, India, 2019, pp. 1-6, doi: 10.1109/CSITSS47250.2019.9031024.

[18] Sopan Talekar and Sujatha Terdal. "Impact of Packet Size on the Performance Evaluation of Routing Protocols in Cognitive Radio Networks," International Journal of Computer Applications 176(36):28-32, July 2020

[19] TauqeerSafdar Malik and MohdHilmi Hasan. Reinforcement Learning-Based Routing Protocol to Minimize Channel Switching and Interference for Cognitive Radio Networks. Complexity, 2020:1–24, 2020.

[20] Talekar Sopan A. and Sujatha P. Terdal. Cooperative Channel Selection With Q-Reinforcement Learning and Power Distribution in Cognitive Radio Networks. IJACI vol.12, no.4 2021: pp.22-42. http://doi.org/10.4018/IJACI.2021100102

[21] Tlouyamma, J., Velempini, M. Channel Selection Algorithm Optimized for Improved Performance in Cognitive Radio Networks. Wireless PersCommun 119, 3161–3178 (2021). https://doi.org/10.1007/s11277-021-08392-5

[22] Asma Amraoui. On a secured channel selection in cognitive radio networks. International Journal of Information and Computer Security, 2022 Vol.18 No.3/4, pp.262 – 27710.1504/IJICS.2022.10050307

[23] Singhal, Chetna& V, Thanikaiselvan. (2022). Cross Layering using Reinforcement Learning in Cognitive Radio-based Industrial Internet of Ad-hoc Sensor Network. International journal of Computer Networks & Communications. 14. 1-17. 10.5121/ijcnc.2022.14401.

[24] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, 90–100. https://doi.org/10.1109/mcsa.1999.749281

AUTHORS

Dr. Sopan A. Talekar received his Doctor of Philosophy (Ph.D.) and M.Tech from Visvesvaraya Technological University (VTU), Belagavi, Karnataka (India). He is currently working as an Assistant Professor in Computer Engineering, MVS's KBT College of Engineering, Nasik, Maharashtra (India) from the last 17 years. He has published & presented papers in international journals & conferences. He is an author of 2 books of Artificial Intelligence, Techmax Publication. He received the prestigious Promising Engineer Award from the Institution of Engineers (India), Local center Nashik, in 2013. His research interests are Cognitive Radio Networks, Wireless Communication, Mobile Ad hoc Network, Artificial Intelligence and Machine Learning.

Dr. Satish S. Banait received his Doctor of Philosophy (Ph. D.) from Savitri baiPhule Pune University, Pune. Post-graduation from Government Engineering College Aurangabad, BAMU University Maharashtra. Presently he is working at K.K.Wagh Institute of Engineering Education & Research Nashik, Maharashtra, India as an Assistant professor from last 17 Years .He has presented papers at National and International conferences and also published paper in international journals on various aspect of the computer engineering and Data mining, Big Data, Machine Learning, Software Engineering, IOT and Network Security.

Dr.Mithun B.Patil received his Doctor of Philosophy (Ph.D.) and M.Tech from Visvesvaraya Technological University (VTU), Belagavi, Karnataka (India). Currently working as Associate Professor in Dept. of Computer Science &Engg of N K Orchid College of Engg& Tech Solapur. He has published more than 13 research papers in renowned Journal and Conferences. He has 9 National and international Patents. His areas of Research Interest are Wireless network, Computer Network, Artificial Intelligence Machine learning