A New Deep Q-Network Design for QoS Multicast Routing in Cognitive Radio MANETs

THONG-NHAT TRAN¹, TOAN-VAN NGUYEN¹, KYUSUNG SHIM¹, DANIEL BENEVIDES DA COSTA², and BEONGKU AN³

¹Dept. of Electronics and Computer Engineering in Graduate School, Hongik University, Republic of Korea (e-mails: tranthnhat@gmail.com, vannguyentoan@gmail.com, shimmkyusung@outlook.kr)
²Future Technology Research Center, National Yunlin University of Science and Technology, Douliu, Yunlin 64002, Taiwan, R.O.C., and with the Dept. of Computer Engineering, Federal University of Ceará, Sobral 62010-560, CE, Brazil (e-mail: danielbcosta@ieee.org)
³Dept. of Software and Communications Engineering, Hongik University, Republic of Korea (e-mail: beongku@hongik.ac.kr)

Corresponding author: Beongku An (E-mail: beongku@hongik.ac.kr; Tel. +82-44-860-2243).

Part of this paper has been presented in the International Conference on Electronics, Information, and Communication (ICEIC) 2021, Jeju Shinhwa World, Republic of Korea, Jan. 31 – Feb. 3, 2021 [1], and the International Conference on Artificial Intelligence in Information and Communication (ICAIIIC) 2021, Jeju Island, Republic of Korea, Apr. 13 – Apr. 16, 2021 [2].

ABSTRACT In this paper, we propose a new deep Q-network (DQN) design for quality-of-service (QoS) multicast routing (DQMR) protocol to establish efficient QoS multicast (EQM) trees in cognitive radio mobile ad hoc networks (CR-MANETs). An EQM tree is a shortest-path multicast tree with minimum end-to-end (E2E) cost (a combination of queuing size ratio and link stability) subject to QoS constraints such as queuing size ratio, link stability, number of hops, number of time slots and avoiding the licensed channel of primary users. Particularly, we propose a NP-complete optimization problem such that its feasible solution is an EQM tree. To address this problem, we design a new DQN model and a new game-based model to form EQM tree in real time by offline training instead of online training like previous papers. Moreover, the DQMR protocol is also guaranteed to have high stability, low routing delay, low control overhead, and high packet delivery ratio (PDR). Furthermore, one more new contribution of the paper is that exact closed-form expressions for the E2E queuing delay of a multicast routing tree are also derived assuming random waypoint mobility and the reference point group mobility models to compare with simulation results of routing delay. Simulation results show that the DQMR protocol outperforms multicast ad hoc on-demand distance vector routing protocol in terms of routing delay, control overhead, and PDR.

INDEX TERMS Cognitive mobile ad hoc networks, cross-layer, deep Q-network, game theory, QoS multicast routing.

I. INTRODUCTION

Cognitive radio (CR) technology has been deployed in mobile ad hoc networks (MANETs) which allows mobile devices to cognitively establish dynamic topologies without necessarily relying on any fixed infrastructure [3]. The benefits of CR are bought by enabling the unlicensed mobile nodes operating in an opportunistic with the licensed spectrum bands, thus improving the spectrum utilization in cognitive radio mobile ad hoc networks (CR-MANETs) [4]. Multicast routing protocols in CR-MANETs mainly relied on flooding operation to find the best route to destinations in the whole network, which often consumes considerable resources such as control overhead, spectrum, delay, and energy [5], [6]. Due to the dynamic nature of MANET environments, the routing optimization problem and QoS constraints are always nondeterministic polynomial-time (NP) complete [7], [8].

Reinforcement learning (RL) is an area of machine learning that enables agents to learn in an interactive environment by trial and error using feedback from its own actions and experiences in order to maximize its reward and minimize its penalty. Due to the versatility of RL, it has ability to solve a myriad of problems ranging from computer vision, speech recognition, robotics, and self-driving car, to wireless communications [9]. Moreover, RL technique is suitable for routing problems in distributed networks such
as CR-MANETs since it has the ability to learn automatically the dynamic features of network such as new flow arrivals, queuing behavior, topology changes, bandwidth, link quality, and energy consumption to enhance the system QoS while optimizing available network resources \cite{9-11}.

### A. RELATED WORK

A stable QoS multicast routing protocol was investigated by the authors in \cite{12} to minimize the network resource utilization while satisfying the jitter delay, reliability, and bandwidth constraints. The optimal multicast routing tree (MRT) was also obtained with the optimal allocation of node buffer and link bandwidth. Yang et al. in \cite{13} investigated the non-asymptotic capacity in MANETs with multicast traffic, where two Markov chain theoretical models were developed to feature the fastest packet propagation process at source and the fastest received packet process at the multicast group.

RL-based routing protocols were extensively studied in wireless ad hoc networks, where the best route was established with low delay, efficient bandwidth, and low energy consumption \cite{9, 14}. The authors in \cite{11} studied a Q-learning reliable routing with a weighting agent approach, where the rewards were given to the agent considering the data transmission latency or network lifetime. In \cite{15}, a Q-learning-based adaptive routing model (QLAR) was developed via RL techniques, which was able to predict the network mobility state information at different times such that each mobile node determined the route with the highest throughput and stability. The authors in \cite{16} studied a QoS-aware Q-routing algorithm in MANETs, where a source node selected its neighbor associated with the optimal Q-value for a destination. By this way, a reactive route with low computational cost and reduced communication overhead was established. To reduce the latency and energy consumption, a Q-learning-based multi-objective optimization routing protocol was proposed in flying ad hoc networks \cite{17}, where the data transmission delay and residual energy of nodes were considered in the reward function for Q-learning.

### B. MOTIVATIONS

Most of previous papers have limitations as follows:

- Q-learning models have not been designed in detail and sufficiently to solve QoS routing optimization and resource allocation problems.
- Since mobile nodes move frequently in MANETs, the Q-learning models must be updated online continuously. Thus, the system spent much time and resources for routing process.
- The channel-time slots allocation issue has received low attention in QoS routing papers, which can decrease the efficiency of data transmission and resource allocation.
- The end-to-end (E2E) queuing delay problem for routing has not been analyzed in previous works, which is essential to estimate the average E2E delay and behavior of the routing protocol.

These unsolved issues motivate us to design a new deep Q-network design for QoS multicast routing leveraging deep Q-network (DQN) and game theory (GT), followed by mathematical analysis of E2E queuing delay (EQD) in this paper. For ease of presentation, Table 1 summarizes the main abbreviations used in this paper.

### C. MAIN CONTRIBUTIONS

In this paper, we study mainly on QoS routing problems in the network layer with information obtained from the physical layer and the data link layer by cross-layer design. The contributions of the paper can be summarized as follows:

- This paper aims to propose a new deep Q-network (DQN) design for quality-of-service (QoS) multicast routing (DQMR) protocol to establish efficient QoS multicast (EQM) trees in cognitive radio mobile ad hoc networks (CR-MANETs). An EQM tree is a shortest-path multicast tree with minimum end-to-end (E2E) cost (a combination of queuing size ratio and link stability) subject to QoS constraints such as queuing size ratio, link stability, number of hops, number of time slots and avoiding the licensed channel of primary users.
- Firstly, we propose a NP-complete optimization problem such that its feasible solution is an EQM tree. Since this problem is too complicated to solve, it is divided into two sub-problems that are minimum E2E cost of multicast tree (MEC) problem and channel-time slot allocation for multicast tree (CTA) problem.
- Secondly, we design a new DQN model, called DQN-MEC model, to address the MEC problem. This model is trained offline to predict optimal link values \((Q^*\text{ values})\), which supports the DQMR protocol to establish minimum E2E cost multicast trees in real time.
- Thirdly, we propose a game-based model to solve the CTA problem, called GT-CTA model. This model supports the DQMR protocol to obtain minimum E2E cost multicast trees with minimum number of time slots for given number of channels, while preventing interference

| TABLE 1: List of Abbreviations. |
|-----------------|-------------------|
| CR              | Cognitive radio   |
| MANET           | Mobile ad hoc network |
| QoS             | Quality-of-service |
| RL              | Reinforcement learning |
| DQN             | Deep Q-network |
| DNN             | Deep neural network |
| GT              | Game theory |
| E2E             | End-to-end |
| EQD-MRT         | E2E queuing delay of a multicast routing tree |
| MEC             | Minimum E2E route cost |
| CTA             | Channel-time slot allocation |
| RWP             | Random waypoint mobility |
| RPGM            | Reference point group mobility |
| DQMR            | DQN-based QoS multicast routing |
| MAODV           | Multicast ad hoc on-demand distance vector |
| EQM             | Efficient QoS multicast tree |
| RREQ            | Route request |
| RREP            | Route reply |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
links and avoiding regions of multiple primary users. Moreover, the design of GT-CTA model is proven mathematically as a convergent potential game.

- Fourthly, the DQMR protocol is proposed by using the DQN-MEC and GT-CTA models to establish EQM trees with high stability, low routing delay, low overhead, and high PDR.

- Fifthly, since the routing delay depends on many factors such as different kinds of delay, mobility model, network topology and so on; it cannot be analyzed correctly. Thus, we derive exact closed-form expressions for the E2E queuing delay of a multicast routing tree (EQD-MRT) under the random waypoint mobility (RWP) and the reference point group mobility (RPGM) models, that show an approximation and the same pattern as the simulation result of routing delay, which confirms the correctness of the developed analysis.

- Finally, the simulation results show that the DQMR protocol outperforms multicast ad hoc on-demand distance vector (MAODV)-based routing protocol [18] in terms of routing delay, control overhead, and PDR.

The rest of the paper is arranged as follows. Section II introduces the system model, the basic concept of DQMR protocol. Section III formulates the QoS multicast routing as an optimization problem. Section IV develops the DQN-MEC model. Section V proposes a GT-CTA model. Section VI proposes the DQMR protocol. Section VII provides a solid theoretical analysis for the EQD-MRT. Section VIII presents the performance evaluations. Finally, Section IX concludes the paper.

II. SYSTEM MODEL
We consider a CR-MANET consisting of multiple primary users (PUs) and secondary users (SUs) as shown in Figs. 1 and 2. Each SU can access opportunistically licensed channels which are not occupied by PUs [4]. In two-dimensional space, the SUs can move based on RWP model and RPGM model [19]–[21], while PUs rely on RWP model. We assume that each node can be aware of its location through the global positioning system (GPS) and the location of destinations in the multicast group [21], [22]. Moreover, each node has a fixed radio range and can exchange control packets by using control channels that do not affect the licensed channels of PUs [23].

A. BASIC CONCEPT OF THE PROPOSED DQMR PROTOCOL
In this paper, we use the same multicast group management techniques as MAODV protocol, e.g., join group, leave group, to maintain the multicast tree. The basic concept of the DQMR at a node, as shown in Figs. 1 and 2 can be presented as follows:

Overview:
- Each SU (node) uses cross-layer design in Fig. 2(a) to get parameters from physical, data link, and network layers such as node’s position, node speed, direction, channel, queue, hop count, IP address of source and destination, affected region of PUs, and multicast tree information. In routing process, these parameters will be used for DQN-MEC model in Fig. 2(b) and GT-CTA model in Fig. 2(c) to obtain EQM trees. Particularly, the DQN-MEC model predicts $Q^*$—values to establish minimum E2E cost multicast trees in real time, and the GT-CTA model selects optimal channel-time slot strategies (Nash equilibrium points) for the minimum E2E cost multicast trees.
Multicast tree discovery:
- If a source (src) needs to establish a multicast tree to the multicast group \( D \), it will require the information of neighbors. For every destination \( dst_i \in D \), the src uses the DQN-MEC model to calculate link values \( Q^*_v(src, w) \) for all \( w \) in the set of the src’s neighbors to select the best neighbor \( w^*_i \) associated with the highest value \( Q^*_v(src, w^*_i) \). Then, the src generates a route request (RREQ) packet and broadcasts it to the set of the best neighbors \( \{w^*_i\} \).
- If a node \( w \in \{w^*_i\} \) receives a RREQ, it will record the sender as the previous node in the route table. Node \( w \) calculates the set of best neighbors to re-broadcast the RREQ packet by the same way as the src.
- If a destination (dst) receives a RREQ packet, it will record the sender as the previous node in the route table and unicast a route reply (RREP) packet to the previous node.
- If a node receives a RREP packet, it will append the sender to the set of next hops (NH) in the route table. Next, node \( v \) forwards the RREP to the previous node by using unicast technique. This process is repeated until the source receives all RREPs from all destinations and go to the channel-time slot allocation process.

Channel-time slot allocation process:
- Each multicast tree member (TM) of the EQM tree applies the GT-CTA model to obtain an optimal channel-time slot strategy. Go to data transmission process.

Data transmission process:
- The src and TMs of the EQM tree send data to the multicast group members based on their next hops (NH) and channel-time slot strategies. If the EQM tree is broken, the maintenance process will be activated and DQN-MEC model and GT-CTA model will be used to locally find alternative routes to the multicast group members.

A CR-MANET is considered as a directed graph \( G = (V, L) \), where \( V \) is a set of SUs, and \( L \) is a set of directed links among nodes. A link between node pairs \((v, w)\) indicates that \( v \) is a sender, \( w \) is a receiver, and \( w \) is within \( v \)'s range and \( v \) is within \( w \)'s range. The set of destinations is referenced to the set of destination’s positions which is denoted as \( D \).

B. QUEUING DELAY MODEL
We assume that a node is a server and number of control packet traffic for routing increases in proportion to the number of links between an intermediate mobile node and its neighbors. Thus, the control packet traffic arrival can be modeled by Poisson process, and the service time is exponentially distributed. Hence, we can employ M/M/1 queuing system for nodes to evaluate and analyze the delay caused by intermediate nodes in routing process, where packets arrive according to Poisson process and the service time is modeled by exponential distribution. The arrival rate and service rate are denoted by \( \lambda \) and \( \mu \), respectively. Based on the Markov chain for M/M/1 system and Little’s theorem \([24]\), each node in the network has a queuing delay model with the following preliminary results: the average time of a packet spending in the system is \( \bar{T} = 1/(\mu - \lambda) \), which includes the queuing delay plus the service time; the average time of a packet spending in queue is \( \bar{W} = \bar{T} - 1/\mu \); the average number of packets in the system is \( \bar{N} = \lambda \bar{T} \); and the average number of packets in the queue is \( \bar{N}_Q = \lambda \bar{W} \).

We define the queue size ratio as a part of the cost function in Eq. (1) which supports the DQN-MEC model to select optimal links with low queuing delay for routing process. The queueing size ratio of a link \((v, w)\) can be expressed as follows:

\[
Q_v(v, w) = \frac{\min\{Q_z(v), Q_z(w)\}}{Q_{z_{\text{max}}}},
\]

where \( Q_z(\cdot) \) is a queue size of a node and \( Q_{z_{\text{max}}} \) denotes the maximum \( Q \) of a node.

C. LINK STABILITY
We use the link stability ratio in \([25]\) as a part of the cost function in Eq. (2) which supports the DQN-MEC model to select optimal links with high stability. The distances between \( v \) and \( w \) at time \( t_i \) and \( t_{i+1} \) are denoted by \( D_{t_i}(v, w) \) and \( D_{t_{i+1}}(v, w) \), respectively. The link stability ratio of a link \( l = (v, w) \) over interval time \( \Delta t = t_{i+1} - t_i \) can be expressed as follows:

\[
LS_{\Delta t}(l) = \begin{cases} 
0, & \text{if } D_{t_{i+1}}(l) \leq D_{t_i}(l), \\
\frac{\Delta D(l)}{2v_{\text{max}}\Delta t}, & \text{otherwise},
\end{cases}
\]

where \( \Delta D(l) = D_{t_{i+1}}(l) - D_{t_i}(l) \) and \( v_{\text{max}} \) is the maximum speed of nodes. Note that the value of \( LS_{\Delta t}(l) \) indicates that the smaller \( LS_{\Delta t}(l) \) is, the higher the stability of the link \( l \).

D. COST FUNCTION
We design a cost function of a link \( l = (v, w) \) as a combination of the queue size ratio in Eq. (1) and the link stability in Eq. (2) which supports the DQN-MEC to select a link with high stability and low queuing delay. Thus, the cost function is used to reduce routing delay and obtain a high stability EQM tree in the routing process, which can be defined as

\[
\text{cost}(l) = \alpha_1 Q_v(l) + \alpha_2 LS_{\Delta t}(l),
\]

where \( \Delta t \) is a period of time and \( \alpha_1 + \alpha_2 = 1 \).

For a source src and a destination dst \( \in D \), we consider a route \( P(src, dst) = \{src = n_0 \rightarrow n_1 \rightarrow \cdots \rightarrow n_{m-1} \rightarrow n_m = dst\} \) of a multicast tree \( T \), where \((n_i, n_{i+1}) \in T, \forall i = 0, \ldots, m - 1 \). The number of hops of route \( P \) is denoted as \( #\text{hops}(P) = m \) and the E2E cost of the route \( P \) can be expressed as

\[
\text{cost}(P) = \sum_{(n_i, n_{i+1}) \in P, \forall i = 0, \ldots, m-1} \text{cost}(n_i, n_{i+1}).
\]
E. CHANNEL MODEL

We present a channel model used for the GT-CTA model, which support the DQMR to establish EQM trees. We assume that there is a set of $L$ licensed channels $\mathcal{C} = \{c_1, \ldots, c_L\}$. In a time slot $t$, each node $v$ only uses either a channel ($ctx^v_t$) to transmit messages or a channel ($crx^v_t$) to receive messages. If node $w$ is a receiver of node $v$, the transmission channel of node $v$ must be the same as the receiving channel of node $w$. The set of receivers of node $v$ in time slot $t$ is denoted as $RCV^v_t$. The set of nodes transmitting on a channel $ch_v$ in time slot $t$ is denoted as $TN^v_t$ and the set of nodes transmitting in time slot $t$ is $TN^t = TN^t_1 \cup \cdots \cup TN^t_L$.

1) The Channel-Time Slot Condition for Preventing Interference

In a time slot $t$, a set of multicast links $ML^v_t = \{(v, w); \forall w \in RCV^v_t\}$ is satisfied for the channel-time slot condition for preventing interference if and only if

$$crx^v_t = ctx^v_t, \forall w \in RCV^v_t, \quad (5)$$

$$NB_{RCV^v_t} \cap TN^v_t = \{\emptyset\}, \quad (6)$$

$$w \notin TN^v_t, \forall w \in RCV^v_t. \quad (7)$$

In a time slot $t$, the condition (5) implies that all receiving channels $crx^v_t$ of all nodes $w \in RCV^v_t$ are the same as the transmission channel $ctx^v_t$ of node $v$, the condition (6) means that only node $v$ can transmit to all nodes $w \in RCV^v_t$ on channel $ctx^v_t$ at time slot $t$ (a node cannot receive from more than one transmitter at the same time) and the condition (7) indicates that when node $v$ transmits to $RCV^v_t$ on channel $ctx^v_t$, all nodes $w \in RCV^v_t$ do not transmit over all channels (a node cannot receive and transmit at the same time).

III. PROBLEM FORMULATION

To support the proposed DQMR protocol to establish EQM trees in routing process, we propose an optimization problem such that its feasible solution is an EQM tree. We consider a tree $T$ as a set of routes from a source to multiple destinations

$$T = \{P_1 = P QDir, dst_1), \ldots, P_M = P QDir, dst_M)\}, \quad (8)$$

where $src$ is the source, $dst_t$ is a destination belonging to multicast group $D_t$ and $M$ is the number of destinations. The E2E cost of the tree $T$ can be represented as $cost(T) = (cost(P_1), \ldots, cost(P_M))$. A tree $T^*$ is a minimum E2E cost tree if every route $P^*_t \in T^*$ has a minimum cost($P^*_t$). We have that $T^* = \arg \min T cost(T)$, where $\mathcal{T}$ is the set of trees from a source to a multicast group and $\min T cost(T) = \{\min cost(P_1), \ldots, \min cost(P_M)\}$. $T^*$

We define a set of time slots as $TS = \{t_{s_1}, \ldots, t_{s_M}\}$. A node $v$ has a channel-time slot strategy which is defined as $CT_v = (t_{s_1}^x, \ldots, t_{s_M}^x, ctx_v^{t_{s_1}}, \ldots, ctx_v^{t_{s_M}})$, where $t_{s_1}^x, \ldots, t_{s_M}^x \in TS$, $ctx_v^{t_{s_1}}, \ldots, ctx_v^{t_{s_M}} \in \mathcal{C} = \{c_1, \ldots, c_L\}$ and $t_{s_1}^x \neq t_{s_2}^x$. A channel-time slot strategy of a tree $T$ is defined as $CT_T = \{CT_v : \forall v \in T\}$. The number of time slots of a route $P$ is defined as $TS(CT_T) = \max_{v \in P} \{t_{s_1}^x\}$, and the number of time slots of the multicast tree $T$ is defined as

$$TS(CT_T) = \max_{v \in T} \{t_{s_1}^x\} = \max_{P \in \mathcal{T}} \{TS(CT_T)\}. \quad (9)$$

The problem can be formulated as follows:

$$\begin{align*}
(P) : & \quad \min_{T \in \mathcal{T}} cost(T) \quad \text{and} \quad \min_{CT_T \in \mathcal{C}} TS(CT_T) \\
& \text{s. t.} \quad \forall t \in T, \quad Qr_P(v, w) \leq Qr_th, \quad \forall P, \quad \forall v, \quad \forall w \quad (10a) \\
& \quad \forall t \in T, \quad LS_P(v, w) \leq LS_th, \quad \forall P, \quad \forall v, \quad \forall w \quad (10b) \\
& \quad \forall t \in T, \quad \#hops_P(v, w) \leq \#hops_th, \quad \forall P, \quad \forall v, \quad \forall w \quad (10c) \\
& \quad CT_T \text{ satisfies the PI condition,} \quad (10d) \\
& \quad CT_T \text{ satisfies the TT condition,} \quad (10e) \\
& \quad CT_T \text{ does not affect PUs,} \quad (10f)
\end{align*}$$

where $Qr(P) = \max_{(v, w) \in P} \{Qr(v, w)\}$, $LS(P) = \max_{(v, w) \in P} \{LS(v, w)\}$.

$\mathcal{C}_{\mathcal{T}}$ denotes a set of channel-time slot strategies ($\mathcal{C}_{\mathcal{T}}$) and constraints (10e) to (10g) are defined as follows:

- The channel-time slot strategy $CT_T$ satisfies the preventing interference (PI) condition (10e) if all sets $ML^v_t, \forall v \in T$ satisfy the conditions (5), (6), (7) defined in Section II-E1.
- The channel-time slot strategy $CT_T$ satisfies the tree-based time slots (TT) condition (10f) if the time slot $t_{s_1}^x$ must be greater than $t_{s_2}^x$ where $w$ is the parent of $v$.
- The channel-time slot strategy $CT_T$ does not affect PUs (10g) if all sets $ML^v_t, \forall v \in T$ does not affect the affected region of PUs.

The problem (P) is a NP-complete problem, and it is a new problem that has not been solved before. To address this problem, we divide it into two sub-problems that are minimum E2E cost of multicast tree (MEC) problem and channel-time slot allocation (CTA) for multicast tree problem.

The MEC problem is formulated to find a shortest-path multicast tree such that each route from a source to a destination of the multicast tree has a minimum E2E cost subject to QoS constraints. The MEC problem can be formulated as

$$\begin{align*}
\text{MEC : } & \quad \min_{T \in \mathcal{T}} cost(T) \\
& \text{s. t.} \quad (10b), (10c), (10d)
\end{align*}$$

The CTA problem is formulated to find an optimal channel-time slot strategy of a tree $T$ with minimum number of time slots, while preventing interference links and avoiding the affected regions of multiple PUs. The CTA problem can be formulated as

$$\begin{align*}
\text{CTA : } & \quad \min_{CT_T \in \mathcal{C}_{\mathcal{T}}} TS(CT_T) \\
& \text{s. t.} \quad (10e), (10f), (10g)
\end{align*}$$
IV. PROPOSED DQN MODEL FOR THE MEC PROBLEM: DQN-MEC MODEL

The DQN-MEC model with offline training in Fig. 2 is designed to predict the optimal $Q^*$-values which are used to select the best neighbors towards the respective destinations in routing process. This neighbors selection process supports select the best neighbors towards the respective destinations in the routing process. This neighbors selection process supports

The DQN-MEC model is run offline once based on a realistic simulation environment on a computer to get a DNN model. Each node is equipped a program which can read realistic simulation environment on a computer to get a DNN scenario with one source and multiple destinations. DQN-MEC model can be efficiently extended to the general scenario of one source and one destination. Then, the obtained DQN-MEC model can be efficiently extended to the general scenario with one source and multiple destinations.

The DQN-MEC model is run offline once based on a realistic simulation environment on a computer to get a DNN model. Each node is equipped a program which can read realistic simulation environment on a computer to get a DNN scenario with one source and multiple destinations. DQN-MEC model can be efficiently extended to the general scenario of one source and one destination. Then, the obtained DQN-MEC model can be efficiently extended to the general scenario with one source and multiple destinations.

The DQN-MEC model is run offline once based on a realistic simulation environment on a computer to get a DNN model. Each node is equipped a program which can read realistic simulation environment on a computer to get a DNN scenario with one source and multiple destinations. DQN-MEC model can be efficiently extended to the general scenario of one source and one destination. Then, the obtained DQN-MEC model can be efficiently extended to the general scenario with one source and multiple destinations.

A. Q-LEARNING MODEL

Q-learning model is designed to make the DQN applicable to the DQMR protocol.

- **Agent**: We consider a node holding a packet or a pair of (packet, node) as an agent which wants to find a route from a source to the destination. Particularly, the packet starts at the source and find the route to a destination which is an optimal solution of MEC problem.

- **State**: The agent has a set of states $S$ which is considered as the set of nodes $V$. At a certain time, if the agent is at node $v \in V$, its state is denoted as $s_v$.

- **Action**: At a certain time, the agent at state $s_v$ has a set of neighbors $NB_v$, which is considered as a set of actions $A_v$ of the agent, i.e., the agent can move to any neighbor in $NB_v$. We denote a node $w \in A_v$ as an action $a_w$ of the agent at state $v$.

- **Environment**: At a certain time, the agent at state $s_v$ has an environment which includes the position, speed and direction information of all node $v$’s neighbors.

- **Reward function**: At state $s_v$, if the agent selects an action $a_w \in A_v$, the reward function of a link $l$ is defined as

$$RW(l) = \begin{cases} -\alpha_c \text{cost}(l) - \alpha_h \text{Wgt}_{\text{hop}}, & \text{if } l \text{ satisfies the QoS conditions,} \\ RW_{\text{min}}, & \text{otherwise}, \end{cases}$$

where $l = (s_v, a_w)$, $\text{Wgt}_{\text{hop}} \in (0, 1)$ denotes a weight of one hop (a connected link between two nodes), $\alpha_c$ and $\alpha_h$ are the weights in $(0, 1)$ such that $\alpha_c + \alpha_h = 1$, and the QoS conditions are

(a) $w \in A_v$, (b) $|q(l)| \leq Q_{rth}$, (c) $LS(l) \leq L_{Sth}$,

The conditions (15a) – (15c) imply the QoS constraints (10b) – (10c) of the MEC problem. For the cost of route (13) and the number of hops constraint (10d), they can only be known after that the route is established. Thus, the metric cost and $\#$ hops are included in the reward function to guarantee that a minimum cost route will be found and a long route will not be formed. The objective function (12) and the constraint (10d) are used to formulate the reward (14), where the values of $\alpha_c$ and $\alpha_h$ are adjusted to obtain the best reward value in the training process. Particularly, when the src obtains the best route to the destination, i.e., the DQN-MEC is converged and there exists a best neighbor $w^* = \pi^*(s_{\text{src}})$, if the number of hops is greater than the constraint $\text{hop}_{\text{th}}$, the weights $\alpha_c$ and $\alpha_h$ of the reward function are adjusted by $\alpha_c = \alpha_c - \varepsilon$ and $\alpha_h = \alpha_h + \varepsilon$ and the DQN-MEC model is repeated until obtaining the best route satisfying the number of hops constraint or exceeding time.

- **Quality function (Q-function)**: At the state $s_v$, the agent takes an action $a_w \in A_v$ to obtain the $Q$-function which is presented as follows:

$$Q(s_v, a_w) := (1 - \alpha)Q(s_v, a_w)$$

$$+ \alpha \left( RW(s_v, a_w) + \gamma \max_{a \in A_v} Q(s_{\text{dst}}, a) > 0 \right),$$

where $\alpha$ and $\gamma$ is the learning rate and discount factor, respectively. We set $\max_{a \in A_v} Q(s_{\text{dst}}, a) = 0$ in (16) to guarantee that the $Q$-values updating process will stop at the destination.

- **Policy**: When the $Q$-values converge to $Q^*$-values, a policy is a function $\pi^*$ that takes state $s_v$ as input and returns the action to be taken by the agent. The policy $\pi^*$ can be expressed as $\pi^*(s_v) = w^* = \arg\max_{a \in A_v} Q^*(s_v, a_w)$. The policy is applied to the DQMR protocol to select the best neighbors in the multicast-tree discovery process.

B. EXPERIENCE REPLAY

Different from regular Q-learning, when the network is complex and frequently changes its topology, experience replay is developed for deep Q-network to learn $Q^*$-values instead of taking much time for re-training. In particular, experience replay is a replay memory technique which is used to store the agent’s experiences at each time-step $e_{t,v} = (s_{t,v}, a_{t,w}, r_t(v, w), s_{t+1,v}, q_t(v, w))$, in a dataset $D = \{e_1, \ldots, e_N\}$, where $r_t(v, w) = RW_t(v, w)$ and $q_t(v, w) = Q_t^*(v, w)$. The experience replay of the DQN-MEC model can be described as follows:

- Based on the simulation time of 1,000 seconds and the section time of 5 seconds in Section VIII, we generate randomly a set of 1,000/5 = 200 environments.
V. PROPOSED GAME-BASED MODEL FOR THE CTA PROBLEM: GT-CTA MODEL

The GT-CTA model is modeled to assist the DQMR protocol to obtain EQM trees with minimum number of time slots for given number of channels, while preventing interference and avoiding regions of multiple primary users. For a given number of channels, while preventing interference which may conflict with its children. Then, its child nodes will update their own strategies to eliminate these conflicts with parent nodes.

• **Payoff:** The payoff of a node \( v \) for taking a strategy \( s_v \in S_v \) is defined as
  \[
  RW_v(s_v, s_{-v}) = \begin{cases} 
  -t_{sx}^v, & \text{if } s_v \text{ satisfies (SS) rules}, \\
  -\infty, & \text{otherwise}.
  \end{cases}
  \]

• **Best Response:** The best-response of a node \( v \) can be expressed as
  \[
  \pi_v(s_{-v}) = s_v^* = \underset{s_v \in S_v}{\arg \max}RW_v(s_v, s_{-v}).
  \]

• **Potential function:** The potential function of the game can be defined as
  \[
  \phi: \mathcal{E}T \rightarrow \mathbb{R} \\
  s_T \mapsto \phi(s_T) = \min_{v \in T} RW_v(s_v, s_{-v}).
  \]

Proof. The proof of the theorem is divided into two parts as follows:

**The first part:** Based on the strategy selection rules, if a node \( v \) chooses a strategy \( s_v = CT_v = (t_{sx}^v, c_{tx}^{v^o}, t_{sx}^w, c_{rx}^{w^o}) \), the strategy \( s_v \) satisfies the rule SS-(iv), i.e., \( s_v \) does not conflict with strategies of all neighbors except for \( n \) of children of node \( v \). We have:

**Case 1:** The strategy \( s_v \) does not conflict with the strategies of node \( v \)’s children.

- If for all \( s_v \in S_v \), \( \phi(s_v, s_{-v}) = RW(s_v, s_{-w}) \) with \( w \neq v \), we have \( \arg \max_{s_v \in S_v} \phi(s_v, s_{-v}) = S_v \). Thus, the condition (20) is satisfied.

\[ \phi(s_v, s_{-v}) = RW(s_v, s_{-v}) \] and \( \max_{s_v \in S_v} \phi(s_v, s_{-v}) = S_v \).
will decrease after each iteration. Thus, the best-response of state \( s_i \) of node \( v \) conflicts with a strategy of a child \( w \) of node \( v \). It means that the function \( \Phi \) always takes \(-\infty\) and \( \arg \max \Phi(s_i, s_{-i}) = S_i \).

Thus, the condition \( 20 \) is satisfied.

**Case 2:** The strategy \( s_i \) of node \( v \) conflicts with a strategy of a child \( w \) of node \( v \). The maximum number of time slots that needs to transmit data without interference is \( M \). Hence, the maximum number of time slots that needs to transmit data from a source to multicast group is \( 1 + M \times (N_{hop} - 1) \).

Agents will obtain new better strategies after each iteration, i.e., the number of time slots of the multicast tree will decrease after each iteration. Thus, the best-response of the game will converge to a Nash equilibrium point within \( 1 + M \times (N_{hop} - 1) \) iterations at most.

Finally, the algorithm of the GT-CTA model at a node \( v \) can be presented as follows:

**Step 1.** Node \( v \) requires the information of strategies \( s_{w} \) for all neighbors \( w \in NB_v \).

**Step 2.** Node \( v \) calculates the set of available strategies \( S_v \) based on strategy selection rules. Next, node \( v \) chooses a best-response \( a_v = \pi_v(a_{-v}) \) in \( 18 \) as a current strategy.

**Step 3.** Steps 1 and 2 are repeated until node \( v \) cannot find a better strategy, i.e., the sum of the payoffs converges.

**VI. THE PROPOSED DQN-BASED QOS MULTICAST ROUTING PROTOCOL: DQMR PROTOCOL**

In this section, we present the DQMR protocol that uses the DQN-MEC and GT-CTA models to establish EQM trees which are a shortest-path multicast tree with minimum E2E cost subject to QoS constraints, preventing interference links and avoiding regions of primary users. Moreover, the DQMR protocol has high stability, low routing delay, low control overhead and high packet delivery ratio (PDR). In practical MANETs, the mobile nodes can move based on different mobility models, as shown in Fig. 3. In particular, nodes 1 to 11 can move according to the RWP model while other nodes can move according to the RPGM model with different groups such as nodes 12, 13, 14 in the first group, nodes 15 to 18 in the second group, and nodes 19 to 22 in the third group.

Thus, the DQMR protocol is tailored to work well in both mobility models. In the given CR-MANET with a source node \( (src) \) and the multicast group \( D \), the DQMR protocol, as shown in Figs. 3 and 4, can be presented as follows:

**Initialization:**

- Each node in the given CR-MANET initializes variables of routing table as follows:
  - The set of last visit nodes \( LV_{rt} = \emptyset \).
  - The route cost \( RC_{rt} = +\infty \).

**Step 1.** If a node needs to establish the tree to the multicast group \( D \), the node becomes a source node \( (src) \), go to Step 2. Otherwise, go to Step 3.

**Multicast Tree Discovery Process (Fig. 4):**

**Sending RREQ Process:**

- The src requires information of neighbors including position, speed, direction, queue size and channels of PUs information. For each destination \( dst_i \), the src predicts values \( Q^*_i(src, w) \) for all \( w \in NB_{src} \) by using the DQN-MEC model to select the best neighbor \( w^*_i \) with the highest value \( Q^*_i(src, w^*_i) \); for example, in Fig. 3, the best neighbors of src are nodes 3, 15, and 16, corresponding to destinations \( dst_1, dst_2, \) and \( dst_3 \).

  - The src updates the set of last visit nodes \( LV_{src} = LV_{rt} \cup \{ src \} \). The set of next visit nodes \( NV_{src} = \{ w^*_i, \forall dst_i \in D \} \) \( LV_{src} \), the list of costs from the src to all next visit nodes \( CL_{src} = \{ cost(src, w^*_i), \forall w^*_i \in NV \} \) and the route cost \( RC_{src} = 0 \). Next, the src generates a route request (RREQ) packet including \( LV_{rreq} = LV_{src}, NV_{rreq} = NV_{src}, CL_{rreq} = CL_{src} \) and \( RC_{rreq} = RC_{src} \), and broadcasts the RREQ to neighbors.

**Receiving RREQ Process:**

- If the node receives a RREQ, go to Step 3.1. Otherwise, the process is ended.

  - **Step 3.1.** The RREQ is dropped if at least one of the following cases is satisfied:
    - The node is not in the list \( NV_{rreq} \) of the RREQ.
    - The new cost \( RC_{rreq} + cost(w, node) \) is smaller than or equal to the route cost \( RC_{rt} \) in the route table, where \( cost(w, node) \) can be found in the \( CL_{rreq} \).

  - **Step 3.2.** The node records the sender’s ID as the previous node. Go back to Step 2.
Route Reply Process (Fig. 4 **∗∗):

- **Step 4.** If the node is the dst, go to Step 5. Otherwise, go to Step 6.
- **Step 5.** If the dst receives a RREQ packet, it will generate and reply a RREP packet to the previous node by unicast transmission, go to Step 9. Otherwise, the process is ended. The RREP packet contains the following fields:
  
  \[
  \text{packet\_type, hop\_count, multicast\_IP\_address, multicast\_seq\_number, source\_IP\_address}
  \]

- **Step 6.** If the node receives a RREP packet, it appends the sender to the set of next hops (NH) in the route table and goes to Step 7. Otherwise, the process is ended.

- **Step 7.** If the node is the src, go to Step 8. Otherwise, node \(v\) unicasts the RREP packet to the previous node, go to Step 8.

Channel-time slot allocation process (Fig. 4 ⋄ ⋄):

- **Step 8.** Each node of the obtained EQM tree, as shown in Fig. 3 applies the GT-CTA model to obtain an optimal channel-time slot schedule such that the EQM tree has minimum number of time slots for given number of channels, while preventing interference links and avoiding the affected regions of multiple PUs. For example in Fig. 3 the EQM tree uses 3 time slots \((t_1, t_2\), and \(t_3)\) and channels \(c_1, c_2\) and \(c_3\) to prevent interference links and avoid the affected regions of PUs. Go to Step 9.

Data Transmission Process (Fig. 4 ⋄ ⋄):

- **Step 9.** The source and mobile nodes of the obtained EQM tree unicasts data to the multicast group mem-
bers based on their next hops (NH) and the optimal channel-time slot strategy. Particularly, the source generates data packets based on the Poisson process. Next, the source multicasts the data packets to the next hops by using the channel-time slot strategy. If a node of the EQM tree receives a data packet, it will forward the data packet to the multicast group by the same way as the source.

Multicast tree maintenance process:
- **Step 10.** During the routing and data transmission processes, if one of established links from a node to the next hop is broken, the node will build alternative routes locally by the same approach as the src in the multicast routing process. Particularly, if the node cannot connect with at least one of next hops, it will require the information of neighbors and calculate LV_{req}, NV_{req}, and CL_{req}. Next, the node generates and broadcasts a RREQ packet to its neighbors. If a node w receives a RREQ from the node and knows routes to the multicast group, it will reply a RREP to the node to establish alternative routes. If node w receives a RREQ from the node and does not know routes to the multicast group, it will continue to find alternative routes to the multicast group by the same approach as the node. Thus, this maintenance process is a local process and it only establishes some alternative links to repair the broken EQM tree.

VII. E2E QUEUING DELAY ANALYSIS

In this section, we present E2E queuing delay analysis to show comparison with E2E queuing delay and routing delay in simulation for the established multicast routing trees.

A. E2E QUEUING DELAY ANALYSIS 1 IN RANDOM WAYPOINT MOBILITY MODEL

We present the analysis of E2D-MRT in the environment of RWP model. As shown in Fig. 3, nodes 1 to 11 move according to the RWP model which can be presented as follows: each node begins by pausing for a number of seconds. Next, the node selects a random direction (angle) in $(0, 2\pi)$ and a random speed in $(0, v_{\text{max}})$ to move in a number of seconds. Then, the node again pauses for a number of seconds before another random direction and speed. This process is repeated over the simulation times.

We assume that the network includes $N$ mobile nodes which are deployed in a square of $A = [0, 1]^2$ with area $S(A) = 1 \text{km}^2$ and nodes can move based on RWP model with the maximum speed $v_{\text{max}}$. We have

- The average distance between two nodes \( \alpha ^2 \) is calculated by the expected distance between two independent points chosen uniformly at random in $A$, which is \( L_A = 0.521405 \).
- The average number of nodes in a region $B \subset A$ is \( N(B) = N \times S(B)/S(A) \).
- The average speed of a node is \( \upsilon = 0.5 \times v_{\text{max}} \).
- The average direction deviation between two any nodes can be calculated by the expected distance between two independent points chosen uniformly at random in $[0, 2\pi]$ which is $\pi = 2\pi/3$.
- Let $v_v$ and $v_w$ be the speeds of node $v$ and $w$, respectively. The distance deviation (DD) between $v$ and $w$ in an interval time $\Delta t = t_{i+1} - t_i$ can be calculated as
  \[
  \text{DD}(v, w, \Delta t) = |D_{ti+1}(v, w) - D_i(v, w)|,
  \]
  where $D_i(v, w)$ and $D_{ti+1}(v, w)$ are the distances between node $v$ and node $w$ at time $t_i$ and $t_{i+1}$, respectively.

**Lemma 1.** The average distance deviation (DD) between two nodes in an interval time $\Delta t$ can be expressed as
  \[
  \text{DD}(v_{\text{max}}, \Delta t) = |\sqrt{AX^2 + BX + C - D}|,
  \]
  where $A = 0.75$, $B = 0.7821075$, $C = 0.271863$, $D = 0.521405$, $X = v_{\text{max}}\Delta t$ and $v_{\text{max}}$ is the maximum speed of each node.

**Proof.** Considering two nodes $v$ and $w$ with $v_{vw} = \alpha_v + \pi = \alpha_w + 2\pi/3$, $v_v = v_w = \upsilon = 0.5 \times v_{\text{max}}$, we have
  \[
  \text{DD}(v_{\text{max}}, \Delta t) = \text{DD}(v, w, \Delta t),
  \]
  where $\Delta t = t_{i+1} - t_i$.

We denote $(x_i(v), y_i(v))$ is the position of node $v$ at time $t_i$. Without loss of generality, we can assume that $x_i(v) > x_i(w)$, $y_i(v) = y_i(w) = 0$ and $\alpha_v = 0$. We have
  \[
  x_{i+1}(v) - x_{i+1}(w) = x_i(v) - x_i(w) + \upsilon\Delta t (\cos \alpha_v - \cos \alpha_w),
  \]
  \[
  y_{i+1}(v) - y_{i+1}(w) = \upsilon\Delta t (\sin \alpha_v - \sin \alpha_w),
  \]
  \[
  D_i(v, w) = \frac{L_A}{2} = 0.521405, \quad D_{ti+1}(v, w) = \sqrt{(L_A + 1.5\upsilon\Delta t)^2 + (0.866025\upsilon\Delta t)^2},
  \]
  \[
  \text{DD}(v_{\text{max}}, \Delta t) = |\sqrt{AX^2 + BX + C - D}|,
  \]
  where $A = 0.75$, $B = 0.7821075$, $C = 0.271863$. The proof of Lemma 1 is concluded.

**Lemma 2.** The average number of nodes moving out of a node $v$’s transmission range (number of node $v$’s broken links) in an interval time $\Delta t$ is
  \[
  \bar{R}_{\text{out}}(X) = (N_{B_v} + 1) \left( \frac{R^2 - (R - \bar{D}(X))^2}{R^2} \right),
  \]
  where $X = (v_{\text{max}}, \Delta t)$, $v_{\text{max}}$ is the maximum speed of each node, $R$ is the transmission range of each node, and $N_{B_v}$ is the average number of node $v$’s neighbors which can be expressed as
  \[
  N_{B_v} = N \pi R^2 / S(A) - 1.
  \]

**Proof.** Lemma 2 can be easily proved based on Lemma 1.
Lemma 3. The average number of packets in a node is
\[ \mathcal{N}_{\text{rwp}}(v_{\text{max}}, \Delta t) = \mathcal{N} + \frac{\lambda}{N_{B_v}} \Delta t \mathcal{N}_{\text{rwp}}^\text{out}(v_{\text{max}}, \Delta t), \] (27)
where \( \lambda \) is the arrival rate of queuing delay model, \( \mathcal{N} \) is the average number of packets in the system, \( v_{\text{max}} \) is the maximum speed of each node, \( \Delta t \) is the maximum lifetime of a node, and \( N_{B_v} \) is the average number of neighbors of a node which is presented as (26).

Proof. The Eq. (27) can be explained as follows:
- The first term in the right-hand side of (27) presents the average number of packets in the system of queuing delay model.
- The second term is the average number of packets that can not be sent to receiver nodes which move out of the transmission range of node \( v \), i.e., these packets still in the queue until lifetime expires.

Thus, the lemma is proven. \( \square \)

For a tree \( T = \{ P_1 = P(\text{src}, \text{dst}_1), \ldots, P_M = P(\text{src}, \text{dst}_M) \} \) in (3), where src is the source, dst is a destination belonging to multicast group \( D \), and \( M \) is the number of destinations. The E2E queuing delay of the tree \( T \) can be calculated as
\[ \text{EQD}(T) = \frac{1}{M} \sum_{i=1}^{M} \sum_{n_i \in P_i} q_{\text{delay}}(n_i), \] (28)
where \( q_{\text{delay}}(n_i) \) is the queuing delay of node \( n_i \).

Theorem 2. We assume that the maximum lifetime of a packet is \( \Delta t \). When a new routing packet arrives at a node at a certain time, the average time of this packet spending in this node is
\[ \mathcal{T}_{\text{rwp}}(v_{\text{max}}, \Delta t) = (\mathcal{N}_{\text{rwp}}(v_{\text{max}}, \Delta t) + 1)/\mu, \] (29)
where \( \mu \) is service time rate of the queuing delay model, \( v_{\text{max}} \) is the maximum speed of each node, \( \mathcal{N}_{\text{rwp}} \) is the average number of packets in a node which is presented as (27).

As a consequence, the EQD queuing delay of a tree \( T \) can be calculated as
\[ \text{EQD}_{\text{rwp}}(T) = (n_{\text{hop}} + 1)\mathcal{T}_{\text{rwp}}(v_{\text{max}}, \Delta t), \] (30)
where \( n_{\text{hop}} \) is the average # hops of routes of the tree \( T \).

Proof. Theorem 2 can be proved by using the results of Lemmas 1, 2, 3 and (25). \( \square \)

B. E2E QUEUING DELAY ANALYSIS 2 IN REFERENCE POINT GROUP MOBILITY MODEL

We present the analysis of EQD-MRT in the environment of RPM model. As shown in Fig. [3] nodes 12 to 22 are divided into three groups and moving according to the RPM model [19], which satisfy the following characteristics:
- The network is divided into multiple adjacent regions. Each region is only occupied by a single group (in-place mobility model).
- Each group has a group leader node and multiple members.
- Each group leader can move according to the RWP model in a fixed region. Each member deviates from the group leader by some degree.

Corollary 1. Assume that the network includes \( N \) nodes, \( K \) groups which are deployed in a square of \( A \) and each node has a fixed radio range \( R \). The average number of nodes moving out of a node \( v \)’s transmission range (number of node \( v \)’s broken links) in an interval time \( \Delta t \) is
\[ \mathcal{N}_{\text{rwp}}^\text{out}(X) = \frac{\mathcal{N}_{\text{rwp}}^\text{out}(X)}{N_{B_v}}, \] (31)
where \( X = (v_{\text{max}}, \Delta t) \), \( v_{\text{max}} \) is the maximum speed of each node and \( N_{B_v} \) is the average number of outside neighbors of node \( v \) which is calculated by (32).

Proof. Given a node \( v \) in a group \( G \), we can consider the region of group \( G \) as a disc \( D_G \) with center \( v_0 \) and radius \( R_G = \sqrt{2(\text{Area} / \pi)} \) while the transmission region of node \( v \) is a disc \( D_v \) with center \( v \) and radius \( R_v = R \). The region \( D_v \setminus (D_G \cap D_v) \) includes nodes which are called outside neighbors of node \( v \). The average number of outside neighbors of node \( v \) can be expressed as follows:
\[ N_{B_v} = N \frac{S(D_v \setminus (D_G \cap D_v))}{S(A)}. \] (32)
The average distance between node \( v \) and the center \( v_0 \) of \( D_G \) (the distance between two centers of \( D_G \) and \( D_v \)) is \( \bar{d} = 2R_G/3 \). Since node \( v \) is in \( D_G \), the value \( R_G + R_v \) is always greater than or equal \( \bar{d} \), i.e., \( R_G + R_v \geq \bar{d} \). We have the following cases:
- If the region \( D_v \) is a subset of the region \( D_G \), i.e., \( R_G - R_v > \bar{d} \),
  \[ S(D_v \setminus (D_G \cap D_v)) = S(\emptyset) = 0. \] (33)
- If the region \( D_G \) is a subset of the region \( D_v \), i.e., \( R_v - R_G > \bar{d} \),
  \[ S(D_v \setminus (D_G \cap D_v)) = S(D_v \setminus D_G) = \pi R_v^2 - \pi R_G^2. \] (34)
- If two regions \( D_v \) and \( D_G \) are overlapped, i.e., \( |R_G - R_v| < \bar{d} \), we have
  \[ S(D_v \setminus (D_G \cap D_v)) = S(D_v) - (A + B - C), \] (35)
where
\[ A = R_{\text{min}}^2 \cos^{-1} \left( \frac{d^2 + R_{\text{min}}^2 - R_{\text{max}}^2}{2dR_{\text{min}}} \right), \]
\[ B = R_{\text{max}}^2 \cos^{-1} \left( \frac{d^2 + R_{\text{max}}^2 - R_{\text{min}}^2}{2dR_{\text{max}}} \right), \]
\[ C = 0.5 \sqrt{4R_{\text{min}}^2R_{\text{max}}^2 - (a - b)^2}, \] (36)
\[ a = (d + R_{\text{min}} + R_{\text{max}}), \]
\[ b = (d + R_{\text{min}} - R_{\text{max}}), \]
\[ c = (d - R_{\text{min}} + R_{\text{max}}). \]
The number $\overline{N}_{\text{out}}$ can be considered as the average number of node $v$’s neighbors which move based on RWP model related to node $v$. Hence, based on (25), the corollary is concluded.

**Corollary 2.** Using the assumptions as in Lemma 3 and Theorem 2 we have

- The average number of packets in a node is calculated the same as in (27), i.e.,
  \[ \overline{N}_{\text{pgm}}(v_{\text{max}}, \Delta t) = \overline{N} + \frac{\lambda}{N_{\text{Be}}} \Delta t \overline{N}_{\text{pgm}}(v_{\text{max}}, \Delta t). \]  

- The average time of this packet spending in a node is calculated the same as in (29), i.e.,
  \[ T_{\text{pgm}}(v_{\text{max}}, \Delta t) = (\overline{N}_{\text{pgm}}(v_{\text{max}}, \Delta t) + 1)/\mu. \]  

- The E2E queuing delay of a tree $T$ is calculated by same as in (30), i.e.,
  \[ \overline{E}_{\text{QD}}_{\text{pgm}}(T) = (n_{\text{hop}} + 1)T_{\text{pgm}}(v_{\text{max}}, \Delta t). \]  

**VIII. PERFORMANCE EVALUATION**

**A. ENVIRONMENTS FOR PERFORMANCE EVALUATION**

In this section, we presents the environments and parameters for the performance evaluation as shown in Table 2.

**TABLE 2:** Simulation Environments and Parameters.

| Parameter                      | Value                   |
|--------------------------------|-------------------------|
| Network size                   | $1000 \times 1000 \text{m}^2$ |
| Number of nodes                | 50                      |
| Mobility model                 | RWP and RPGM            |
| Max speed of nodes             | $20 \text{km/h} \sim 80 \text{km/h}$ |
| Number of PU                   | 3 \sim 5                |
| Transmission range of nodes    | 250m                    |
| Coverage range of PU           | 250m                    |
| Number of licensed channels    | 5                       |
| Queue arrival rate             | $\lambda = 10 \text{packets/s}$ |
| Queue service time rate        | $\mu = 20 \text{packets/s}$ |
| Data transmission process      | Poisson                 |
| Data transmission rate         | $5 \text{packets/s}$    |
| The lifetime of a packet       | $\Delta t = 0.3s$       |
| Session length                 | 5s                      |
| Simulation time                | 1,000s                  |

The DQMR protocol is implemented under RWP model in VII-A and RPGM model VII-B. In the RWP model in VII-A, we set the pausing time as 3 seconds, the moving time as 5 seconds. In the RPGM model, we set the number of groups is 4 or 9.

**B. PERFORMANCE METRICS**

To evaluate the performance of the DQMR, the following metrics are considered:

- Routing delay is defined by the average time to establish a multicast tree per one session.
- The control overhead is defined by the average number of control packets to establish a multicast tree per session per node.
- The PDR is defined by the average number of data packets delivered to multicast group over the number of data packets supposed to be delivered to destination per session.
- E2E queuing delay is defined by the average E2E queuing analysis delay of multicast routing trees in (28) per one session.

**C. THE CONVERGENCE PERFORMANCE OF THE DQN-MEC MODEL AND THE GT-CTA MODEL**

The convergence performance of the DQN-MEC model is shown in Fig. 5(a). This confirms the DQN-MEC model converges quickly after 1,000 epoches which shows that the DQN-MEC model can achieve the $Q^*$ – values for routing process in training process. Moreover, Fig. 5(b) shows the rapid convergence of the total payoffs of the GT-CTA model within only $1 + M \times (N_{\text{hop}} - 1) = 1 + (4 - 1) \times 5 = 16$ iterations, where $N_{\text{hop}} = 4$ is the maximum number of hops, $M = 5$ is the number of destinations. This result show that the proposed game with GT-CTA model achieves the optimal solution of problem (??) with small iterations, which also confirms the results in Theorem 1. This short-time convergence may expedite the feasibility of the practical implementation of the channel-time slot allocation based on game theory in CR-MANETs.

**D. NUMERICAL RESULTS FOR THE RWP MODEL**

We present the numerical results of the DQMR protocol in the environment of RWP model by using simulation.

In Fig. 6 we show the routing delay as a function of node speed for RWP model. As can be observed, the routing delay of the DQMR protocol is lower than that of the MAODV-based one in most of node speed. The reason is that instead of flooding the RREQ packets in MAODV-based protocol, the DQMR protocol only multicasts RREQs to the predicted best neighbors based on the DQN-MEC model, thus, reducing routing delay. In addition, the DQN-MEC and GT-CTA models support the DQMR protocol to obtain EQM trees.
with high stability and high reliability, which also alleviates the re-routing processes and routing delay.

Fig. 7 presents the control overhead as a function of node speed for RWP model. As can be observed, the control overhead increases gradually with the growth of maximum speed of node, and the control overhead of the DQMR protocol is lower than that of the MAODV-based one. The reason is that the DQMR protocol just multicasts RREQs to the predicted best neighbors instead of conventional flooding. Moreover, the DQMR protocol can form EQM trees with high stability and high reliability based on the DQN-MEC and GT-CTA models. Hence, the control overhead of the DQMR protocol can be effectively reduced.

In Fig. 9, we show the scalability of the DQMR protocol by demonstrating the PDR as a function of multicast group size for RWP model. As can be observed, the PDR has almost constant value and is not affected by the number of destinations. The reason is that our DQMR protocol employs the DQN-MEC model and GT-CTA model to create the underlying tree-based structure that can improve the stability and scalability of the DQMR protocol under different sizes of multicast group.

In Fig. 10, we plot the number of time slots allocating for packet transmission as a function of number of PUs for RWP model. When the number of PUs is increased, the system requires more time slots for data packet transmission to avoid interfering with the licensed channel of PUs. It is observed that the protocols without using GT-CTA model consumes more time slots for packet transmission than the ones with GT-CTA model. The reason is that the GT-CTA model can help the DQMR to form EQM trees with minimum number of time slots.

E. NUMERICAL RESULTS FOR THE RPGM MODEL

We present the numerical result of the DQMR protocol in the environment of RPGM model by using simulation.
In Fig. 11, we show the routing delay as a function of node speed for RPGM model. As can be observed, the routing delay of the DQMR protocol is lower than that of the MAODV-based one in most of node speed. The reason is that based on the DQN-MEC and GT-CTA models, the DQMR protocol which only multicasts RREQs to the predicted best neighbors can obtain a high stability and reliability EQM trees. Thus, it can reduces the re-routinging processes and routing delay. Besides, based on the simulation parameters in Table 2, the EQD-MRT can be calculated by Corollary 1 to show that the EQD-MRT and routing delay of RPGM model with 9 groups is smaller than its counterpart with 4 groups.

Fig. 12 presents the control overhead as a function of node speed for RPGM model. It can be observed that the control overhead of DQMR protocol is lower than that of the MAODV-based one. With the deployment of the DQN-MEC and GT-CTA models, the DQMR protocol just multicasts RREQs to the predicted best neighbors and establishes EQM trees with high stability and high reliability. Moreover, based on Eq. (31), the average number of a node’s broken links of RPGM model with 9 groups is smaller than its counterpart with 4 groups. This leads to a smaller control overhead of the RPGM model with 9 groups compared to its counterpart with 4 groups.

In Fig. 13, we consider the number of time slots allocating for packet transmission as a function of number of PUs for RPGM model. The system requires more time slots for packet transmission to avoid interfering with the licensed channel of PUs as the number of PUs increases. It is shown that the protocols without using GT-CTA model consumes more time slots for data transmission than the ones with GT-CTA model. This shows the benefit of the designed game.

Fig. 14 shows the PDR of protocols with 3 PUs as a function of node speed for RPGM model. At the maximum speed of 80 km/h with RPGM (9 group) mobility model, the DQMR protocol achieves about 95% while the MAODV-based one is only about 87%. The DQMR protocol can establish high stability EQM trees having optimal channel-time slot strategies that helps the data packet to reach the destination faster and more reliability than MAODV-based protocol. Furthermore, the PDR of all protocols assuming the RPGM model with 9 groups is also higher than that of using 4 groups due to the smaller node’s broken links when deploying a larger number of groups as in (31).

In Fig. 15, we consider the number of time slots allocating for packet transmission as a function of number of PUs for RPGM model. The system requires more time slots for packet transmission to avoid interfering with the licensed channel of PUs as the number of PUs increases. It is shown that the protocols without using GT-CTA model consumes more time slots for data transmission than the ones with GT-CTA model. This shows the benefit of the designed game.
4 of EQD-MRT is calculated by nodes on a route of the multicast tree is RPGM model. As can be observed, the average number of MRT) with the comparison of the simulation result for RWP speed.

Thus, the analysis of EQD-MRT instead of routing delay, that show an approximation and the same pattern as the simulation result of routing delay, which confirms the correctness of the developed analysis.

The small gap between the analytical results and simulation ones in Figs. [16] and [17] is due to the fact that the analysis is performed based on the average time of a packet spending in a node, as shown in (29) and (38). On the other hand, the cost in (3) includes queue size ratio parameter and the simulation results rely on the DQMR protocol to find EQM trees with high stability and high reliability. Thus, the simulation result of routing delay is smaller than the analysis of EQD-MRT.

IX. CONCLUSIONS
In this paper, we proposed a DQMR protocol assisted by game-based channel-time slot allocation to establish EQM trees in CR-MANETs. Particularly, the DQMR protocol used the DQN-MEC model to establish shortest-path multicast trees with minimum E2E cost subject to QoS constraints. Besides, the DQMR protocol also used the GT-CTA model for the obtained tree to minimize the number of time slots, prevent interference links and avoid regions of primary users. Moreover, the DQMR protocol was also guaranteed to have high stability, low routing delay, low control overhead and high PDR. Furthermore, exact closed-form expressions for the EQD-MRT are also derived assuming RWP model and RPGM model to compare with routing delay in simulation. The evaluation results showed that the DQMR protocol outperformed the MAODV-based one in terms of control overhead, PDR, and routing delay, showing to be an efficient protocol in CR-MANETs. In future works, we will propose multicast routing protocol with deep reinforcement learning and different mobility models to address the multiple sources problem, which promises in providing an ultra-reliable and low-latency routing protocol in high dynamic environments for 5G and future CR-MANETs.

REFERENCES
[1] T.-N. Tran, T.-V. Nguyen, K. Shim, and B. An, “DQR: A deep reinforcement learning-based QoS routing protocol in cognitive radio mobile ad hoc

FIGURE 15: Number of time slots with 50 km/h and 5 licensed channels as a function of number of PUs for RPGM model.

FIGURE 16: The EQD-MRT of the proposed DQMR protocol as a function of node speed for RWP model.

FIGURE 17: The EQD-MRT of the proposed DQMR protocol as a function of node speed for RPGM model.

theory approach in Section [7] which helps to improve the resource utilization of DQMR protocol.

F. ANALYSIS RESULTS OF DELAY: EQD-MRT
We presents the delay analysis results for E2E queuing delay of a multicast routing tree (EQD-MRT) with the comparison of the simulation results. Since the routing delay depends on many factors such as different kinds of delay, mobility model, network topology and so on; it cannot be analyzed correctly. Thus, we analyze the EQD-MRT instead of routing delay, that show an approximation and the same pattern as the simulation result of routing delay, which confirms the correctness of the developed analysis.

Fig. [16] presents the analysis of E2E queuing delay (EQD-MRT) with the comparison of the simulation result for RWP model. As can be observed, the average number of nodes on a route of the multicast tree is 4. Thus, the analysis of EQD-MRT is calculated by 4\(T_{\text{wp}}\) in (30) of Theorem [2]. The analytical result of the EQD-MRT has the same pattern as the simulation result of routing delay which can well estimate the tendency and behaviors of EQD-MRT in terms of node speed.

Fig. [17] presents the analysis of E2E queuing delay (EQD-MRT) with the comparison of the simulation result for RPMG model. As can be observed, the average number of nodes on a route of the multicast tree is 4. Thus, the analysis of EQD-MRT is calculated by 4\(T_{\text{rpmg}}\) in (39) of Corollary

for RWP Model

for RPGM Model

Routing Delay (Simulation)

E2E Queuing Delay (Analysis)
networks,” in 2021 Int. Conf. on Electron., Inf., and Commun. (ICEIC), 2021, pp. 1–4.

[2] T. N. Tran, T.-V. Nguyen, K. Shim, and B. An, “An optimal QoS multicast routing protocol in IoT enabling cognitive radio MANETs: A deep Q-learning approach,” in 2021 Int. Conf. on Artif. Intell. in Inf. and Commun. (ICAIC), 2021, pp. 279–283.

[3] F. R. Yu and H. Tang, Cognitive radio mobile ad hoc networks. Springer, 2011, vol. 507.

[4] I. P. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, “CRAHNs: Cognitive radio ad hoc networks,” Ad Hoc Networks, vol. 7, no. 5, pp. 810–836, 2009.

[5] J. Yu, N. Wang, G. Wang, and D. Yu, “Connected dominating sets in wireless ad hoc and sensor networks—a comprehensive survey,” Comput. Commun., vol. 36, no. 2, pp. 121–134, 2013.

[6] T. Lu and J. Zhu, “Genetic algorithm for energy-efficient QoS multicast routing,” IEEE Commun. Lett., vol. 17, no. 1, pp. 31–34, Jan. 2012.

[7] Chenxi Zhu, “Medium access control and quality-of-service routing for mobile ad hoc networks,” Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20906, 2001.

[8] Chenxi Zhu and M. S. Corson, “QoS routing for mobile ad hoc networks,” in Proc. 21st Ann. Joint Conf. IEEE Comput. Commun. Soc. IEEE INFOCOM Conf. Comput. Commun., vol. 2, 2002, pp. 958–967.

[9] Z. Mammeri, “Reinforcement learning based routing in networks: Review and classification of approaches,” IEEE Access, vol. 7, pp. 55 916–55 950, Apr. 2019.

[10] L. Zhao, J. Wang, J. Liu, and N. Kato, “Routing for crowd management in smart cities: A deep reinforcement learning perspective,” IEEE Commun. Mag., vol. 57, no. 4, pp. 88–93, 2019.

[11] G. Künzel, L. S. Indrusiak, and C. E. Pereira, “Latency and lifetime enhancements in industrial wireless sensor networks: A Q-learning approach for graph routing,” IEEE Trans. Ind. Inform., vol. 16, no. 8, pp. 5617–5625, 2019.

[12] W. Sheikh, “A resource-tuned QoS multicast routing protocol,” Int. J. of Commun. Syst., vol. 31, no. 11, p. e3570, 2018.

[13] B. Yang, Z. Wu, Y. Fan, X. Jiang, and S. Shen, “Non-asymptotic capacity study in multicast mobile ad hoc networks,” IEEE Access, vol. 7, pp. 115 109–115 121, 2019.

[14] H. Yao, H. Liu, P. Zhang, S. Wu, C. Jiang, and S. Guo, “A learning-based approach to intra-domain QoS routing,” IEEE Trans. Veh. Technol., 2020.

[15] A. Serhani, N. Naja, and A. Jamali, “QLAR: A Q-learning based adaptive routing for MANETs,” in Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl. (AICCSA), 2016, pp. 1–7.

[16] T. Hendriks, M. Camejo, and S. Latré, “Q2-routing: A QoS-aware Q-routing algorithm for wireless ad hoc networks,” in Proc. 14th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob), IEEE, 2018, pp. 108–115.

[17] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, and Y. Xu, “QMRE: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks,” Comput. Commun., vol. 150, pp. 304–316, 2020.

[18] E. M. Royer and C. E. Perkins, “Multicast operation of the ad-hoc on-demand distance vector routing protocol,” in Proc. of the 5th Annu. ACM/IEEE Int. Conf. on Mobile Comput. and Netw., ser. MobiCom ’99, New York, NY, USA: Association for Computing Machinery, 1999, p. 207–218. [Online]. Available: https://doi.org/10.1145/313451.313538

[19] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, “A group mobility model for ad hoc wireless networks,” in Proc. of the 2nd ACM Int. Workshop on Model., Anal. and Simul. of Wireless and Mobile Syst., ser. MSWIM ’09, New York, NY, USA: Assoc. for Comput. Machinery, 1999, p. 53–60. [Online]. Available: https://doi.org/10.1145/1313451.131358

[20] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” Wireless commun. and mobile comput., vol. 2, no. 5, pp. 483–502, 2002.

[21] Beongku An and S. Papavassiliou, “A mobility-based hybrid multi-cast routing in mobile ad-hoc wireless networks,” in 2001 MILCOM Proc. Commun. for Network-Centric Oper.: Creating the Inf. Force (Cat. No.01CH37277), vol. 1, 2001, pp. 316–320 vol.1.

[22] A. A. Papadopoulos and J. A. McCann, “Towards the design of an energy-efficient, location-aware routing protocol for mobile, ad-hoc sensor networks,” in Proc. 15th Int. Workshop on Database and Expert Syst. Appl., 2004, pp. 705–709.

[23] K. Chowdhury and M. Felice, “SEARCH: A routing protocol for mobile cognitive radio ad-hoc networks,” Comput. Commun., vol. 32, no. 18, pp. 1993–1997, 2009.

[24] D. Bertsekas and R. Gallager, Data Networks (2nd Ed.). Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1992.

[25] T. N. Tran, T. V. Nguyen, K. Shim, and B. An, “A game theory based clustering protocol to support multicast routing in cognitive radio mobile ad hoc networks,” IEEE Access, vol. 8, pp. 141 310–141 330, 2020.

[26] M. Voornveeld, “Best-response potential games,” Economics Letters, vol. 66, no. 3, pp. 289–295, 2000.

[27] C. Bettstetter, G. Resta, and P. Santi, “The node distribution of the random waypoint mobility model for wireless ad hoc networks,” IEEE Trans. Mobile Comput., vol. 2, no. 3, pp. 257–269, 2003.