IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 4, 2021, accepted November 1, 2021, date of publication November 9, 2021,
date of current version November 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126844

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

THONG-NHAT TRAN[1], (Graduate Student Member, IEEE), TOAN-VAN NGUYEN[2], (Member, IEEE), KYUSUNG SHIM[1], (Member, IEEE), DANIEL BENEVIDES DA COSTA[3,4], (Senior Member, IEEE), AND BEONGKU AN[5], (Member, IEEE)

[1]Department of Electronics and Computer Engineering in Graduate School, Hongik University, Seoul 04066, Republic of Korea
[2]Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322, USA
[3]Future Technology Research Center, National Yunlin University of Science and Technology, Douliu, Yunlin 64002, Taiwan
[4]Department of Computer Engineering, Federal University of Ceará, Sobral 62010-560, Brazil
[5]Department of Software and Communications Engineering, Hongik University, Seoul 04066, Republic of Korea

Corresponding author: Beongku An (beongku@hongik.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant NRF-2019R1A2C1083996. The work of Daniel Benevides Da Costa was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant 110-2222-E-224-003.

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 an 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 trees in real-time by offline training instead of online training as done in 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 non-deterministic 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

The associate editor coordinating the review of this manuscript and approving it for publication was Salekul Islam.

communications [9]. Moreover, RL technique is suitable for routing problems in distributed networks such as CR-MANETs since it has 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 [9]–[11].

A. RELATED WORK

A stable QoS multicast routing protocol was investigated by the authors in [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.* [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 [9], [14]. The authors in [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 [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 [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 [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 less 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.

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

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 an 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 online link values (Q^*-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 links and avoiding affected 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 packet delivery ratio (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 is 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:

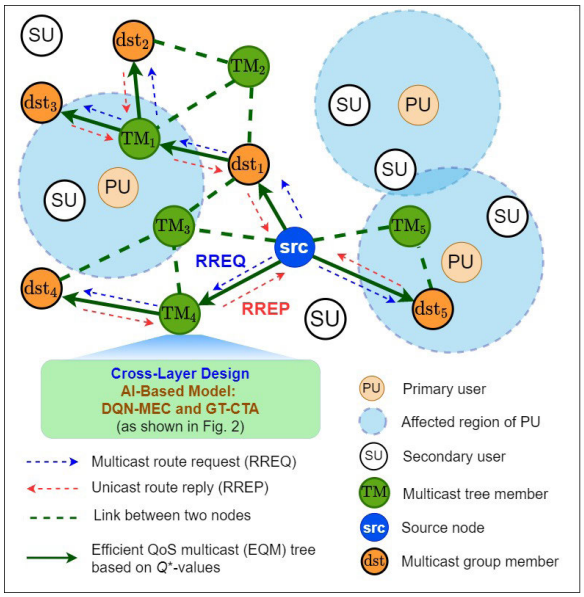

FIGURE 1. Basic concept of the DQMR protocol.

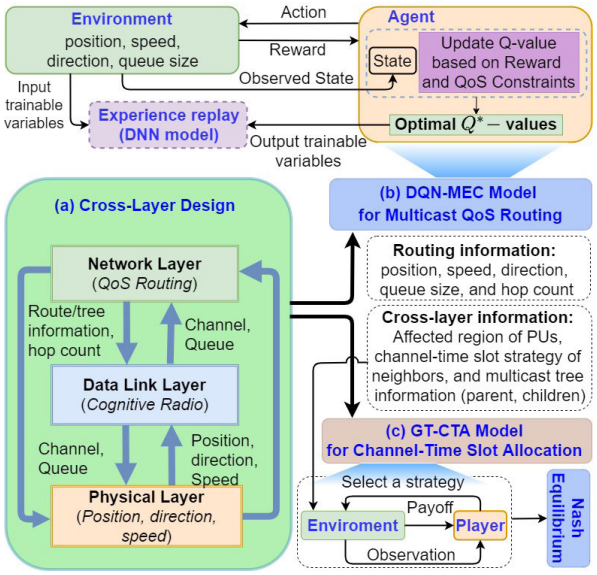

FIGURE 2. Illustration of the basic concept of the DQMR protocol: (a) cross-layer design, (b) DQN-MEC model, and (c) GT-CTA model for multicast QoS routing.

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 $\mathcal{D}$, it will require the information of neighbors. For every destination $\mathsf{dst}_i \in \mathcal{D}$, the src uses the DQN-MEC model to calculate link values $Q_i^*(\mathsf{src}, w)$ for all w in the set of the src's neighbors to select the best neighbor w_i^* associated with the highest value $Q_i^*(\mathsf{src}, w_i^*)$. 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 $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is a set of SUs, and $\mathcal{L}$ 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 $\mathcal{D}$.

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 λ and μ, 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 $\overline{\mathcal{T}} = 1/(\mu - \lambda)$, which including the queuing delay plus the service time; the average of time a packet spending in queue is $\overline{\mathcal{W}} = \overline{\mathcal{T}} - 1/\mu$; the average number of packets in the system is $\overline{\mathcal{N}} = \lambda\overline{\mathcal{T}}$; and the average number of packets in the queue is $\overline{\mathcal{N}}_Q = \lambda\overline{\mathcal{W}}$.

We define the queue size ratio as a part of the cost function in Eq. (3) which supports the DQN-MEC model to select optimal links with low queuing delay for routing process. The queuing size ratio of a link (v, w) can be expressed as follows:

$$\mathsf{Qr}(v, w) = \frac{\max\{\mathsf{Qz}(v), \mathsf{Qz}(w)\}}{\mathsf{Qz}_{\max}}, \tag{1}$$

where $\mathsf{Qz}(\cdot)$ is a queue size of a node and $\mathsf{Qz}_{\max}$ denotes the maximum Qz of a node.

C. LINK STABILITY

We use the link stability ratio in [25] as a part of the cost function in Eq. (3) 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 $\mathsf{D}_{t_i}(v, w)$ and $\mathsf{D}_{t_{i+1}}(v, w)$, 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:

$$\mathsf{LS}_{\Delta t}(l) = \begin{cases} 0, & \text{if } \mathsf{D}_{t_{i+1}}(l) \le \mathsf{D}_{t_i}(l), \\ \dfrac{\Delta \mathsf{D}(l)}{2\mathsf{v}_{\max}\Delta t}, & \text{otherwise}, \end{cases} \tag{2}$$

where $\Delta\mathsf{D}(l) = \mathsf{D}_{t_{i+1}}(l) - \mathsf{D}_{t_i}(l)$ and $\mathsf{v}_{\max}$ is the maximum speed of nodes. Note that the value of $\mathsf{LS}_{\Delta t}(l)$ indicates that the smaller $\mathsf{LS}_{\Delta t}(l)$ is, the higher the stability of the link l is.

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

$$\mathsf{cost}(l) = \alpha_1\mathsf{Qr}(l) + \alpha_2\mathsf{LS}_{\Delta t}(l), \tag{3}$$

where Δt is a period of time and $\alpha_1 + \alpha_2 = 1$.

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

$$\mathsf{cost}(\mathtt{P}) = \sum_{\substack{(n_i, n_{i+1}) \in \mathtt{P}, \\ \forall i = 0, \ldots, m-1}} \mathsf{cost}(n_i, n_{i+1}). \tag{4}$$

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 $\mathcal{C} = \{\mathsf{ch}_1, \ldots, \mathsf{ch}_L\}$.

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_c in time slot t is denoted as TN_c^t and the set of nodes transmitting in time slot t is $\mathsf{TN}^t = \mathsf{TN}_1^t \cup \cdots \mathsf{TN}_L^t$.

1) THE CHANNEL-TIME SLOT CONDITION FOR PREVENTING INTERFERENCE

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

$$\begin{cases} \mathsf{crx}_w^t = \mathsf{ctx}_v^t, \forall w \in \mathsf{RCV}_v^t, & (5) \\ \mathsf{NB}_{\mathsf{RCV}_v^t} \cap \mathsf{TN}_{\mathsf{ctx}_v^t}^t = \{v\}, & (6) \\ w \notin \mathsf{TN}^t, \forall w \in \mathsf{RCV}_v^t. & (7) \end{cases}$$

In a time slot t, the condition (5) implies that all receiving channels crx_w^t of all nodes $w \in \mathsf{RCV}_v^t$ 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 $w \in \mathsf{RCV}_v^t$ 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 $w \in \mathsf{RCV}_v^t$ 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 $\mathtt{T}$ as a set of routes from a source to multiple destinations

$$\mathtt{T} = \{\mathtt{P}_1 = \mathtt{P}(\mathsf{src}, \mathsf{dst}_1), \ldots, \mathtt{P}_M = \mathtt{P}(\mathsf{src}, \mathsf{dst}_M)\}, \quad (8)$$

where src is the source, dst_i is a destination belonging to multicast group $\mathcal{D}$, and M is the number of destinations. The E2E cost of the tree $\mathtt{T}$ can be represented as $\mathsf{cost}(\mathtt{T}) = (\mathsf{cost}(\mathtt{P}_1), \ldots, \mathsf{cost}(\mathtt{P}_M))$. A tree $\mathtt{T}^*$ is a minimum E2E cost tree if every route $\mathtt{P}_i^* \in \mathtt{T}^*$ has a minimum $\mathsf{cost}(\mathtt{P}_i^*)$. We have that $\mathtt{T}^* = \arg\min_{\mathtt{T}\in\mathcal{T}} \mathsf{cost}(\mathtt{T})$, where $\mathcal{T}$ is the set of trees from a source to a multicast group and $\min_{\mathtt{T}\in\mathcal{T}} \mathsf{cost}(\mathtt{T}) = \{\min_{\mathtt{T}\in\mathcal{T}} \mathsf{cost}(\mathtt{P}_1), \ldots, \min_{\mathtt{T}\in\mathcal{T}} \mathsf{cost}(\mathtt{P}_M)\}$.

We define a set of time slots as $\mathcal{TS} = \{\mathsf{ts}_1, \ldots, \mathsf{ts}_M\}$. A node v has a channel-time slot strategy which is defined as $\mathsf{CT}_v = (\mathsf{ts}_v^{\mathsf{tx}} = t^{\mathsf{tx}}, \mathsf{ctx}_v^{t^{\mathsf{tx}}}, \mathsf{ts}_v^{\mathsf{rx}} = t^{\mathsf{rx}}, \mathsf{crx}_v^{t^{\mathsf{rx}}})$, where $\mathsf{ts}_v^{\mathsf{tx}}, \mathsf{ts}_v^{\mathsf{rx}} \in \mathcal{TS}$, $\mathsf{ctx}_v^{t^{\mathsf{tx}}}, \mathsf{crx}_v^{t^{\mathsf{rx}}} \in \mathcal{C} = \{\mathsf{ch}_1, \ldots, \mathsf{ch}_L\}$ and $\mathsf{ts}_v^{\mathsf{tx}} \neq \mathsf{ts}_v^{\mathsf{rx}}$. A channel-time slot strategy of a tree $\mathtt{T}$ is defined as $\mathsf{CT}_\mathtt{T} = \{\mathsf{CT}_v : \forall v \in \mathtt{T}\}$. The number of time slots of a route $\mathtt{P}$ is defined as $\mathsf{TS}(\mathsf{CT}_\mathtt{P}) = \max_{v\in\mathtt{P}}\{\mathsf{ts}_v^{\mathsf{tx}}\}$, and the number of time slots of the multicast tree $\mathtt{T}$ is defined as

$$\mathsf{TS}(\mathsf{CT}_\mathtt{T}) = \max_{v\in\mathtt{T}}\{\mathsf{ts}_v^{\mathsf{tx}}\} = \max_{\mathtt{P}\in\mathtt{T}}\{\mathsf{TS}(\mathsf{CT}_\mathtt{P})\}. \quad (9)$$

The problem can be formulated as follows:

$$\begin{aligned} (\mathtt{P}): \min_{\mathtt{T}\in\mathcal{T}} \mathsf{cost}(\mathtt{T}) \text{ and } \min_{\mathsf{CT}_\mathtt{T}\in\mathcal{CT}_\mathtt{T}} \mathsf{TS}(\mathsf{CT}_\mathtt{T}) & \quad (10a) \\ \text{s. t. } \mathsf{Qr}(\mathtt{P}_i) \le \mathsf{Qr}_{\mathsf{th}} \quad \forall \mathtt{P}_i \in \mathtt{T}, & \quad (10b) \\ \mathsf{LS}(\mathtt{P}_i) \le \mathsf{LS}_{\mathsf{th}}, \quad \forall \mathtt{P}_i \in \mathtt{T}, & \quad (10c) \\ \#\mathsf{hops}(\mathtt{P}_i) \le \#\mathsf{hops}_{\mathsf{th}}, \quad \forall \mathtt{P}_i \in \mathtt{T}, & \quad (10d) \\ \mathsf{CT}_\mathtt{T} \text{ satisfies the PI condition}, & \quad (10e) \\ \mathsf{CT}_\mathtt{T} \text{ satisfies the TT condition}, & \quad (10f) \\ \mathsf{CT}_\mathtt{T} \text{ does not affect PUs}, & \quad (10g) \end{aligned}$$

where the queue size ratio (Qr) and link-stability ratio (LS) of a route $\mathtt{P}$ are defined as follows:

$$\mathsf{Qr}(\mathtt{P}) = \max_{(v,w)\in\mathtt{P}}\{\mathsf{Qr}(v, w)\}, \quad \mathsf{LS}(\mathtt{P}) = \max_{(v,w)\in\mathtt{P}}\{\mathsf{LS}(v, w)\}, \quad (11)$$

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

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

The problem (P) is an 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

$$\begin{aligned} \mathtt{MEC}: \min_{\mathtt{T}\in\mathcal{T}} \mathsf{cost}(\mathtt{T}) \\ \text{s. t. (10b), (10c), (10d).} \end{aligned} \quad (12)$$

The CTA problem is formulated to find an optimal channel-time slot strategy of a tree $\mathtt{T}$ 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

$$\begin{aligned} \mathtt{CTA}: \min_{\mathsf{CT}_\mathtt{T}\in\mathcal{CT}_\mathtt{T}} \mathsf{TS}(\mathsf{CT}_\mathtt{T}) \\ \text{s. t. (10e), (10f), (10g).} \end{aligned} \quad (13)$$

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 the DQMR protocol in establishing EQM trees. For every destination $\mathsf{dst}_i \in \mathcal{D}$, we need to find a route $\mathtt{P}_i^*(\mathsf{src}, \mathsf{dst}_i)$ which is a solution of the MEC problem. Hence, we first propose a DQN-MEC model for the MEC problem in the 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 the resulting DNN model to predict the Q^*-values for routing process in real-time. When the network environment is changed with network size and number of nodes, the training process will be retrained, and each node will update the new DNN model. The proposed DQN-MEC model is modeled as a model-free RL which includes Q-learning model and experience replay as follows:

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 finds the route to a destination which is an optimal solution of MEC problem.
- **State:** The agent has a set of states $\mathcal{S}$ which is considered as the set of nodes $\mathcal{V}$. At a certain time, if the agent is at node $v \in \mathcal{V}$, 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 $\mathcal{A}_v$ of the agent, i.e., the agent can move to any neighbor in NB_v. We denote a node $w \in \mathcal{A}_v$ 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 $\mathsf{a}_w \in \mathcal{A}_v$, the reward function of a link l is defined as

$$\mathsf{RW}(l) = \begin{cases} -\alpha_c \mathsf{cost}(l) - \alpha_h \mathsf{Wgt}_h, \\ \qquad \text{if } l \text{ satisfies the QoS conditions}, \\ \mathsf{RW}_{\min}, \qquad \text{otherwise}, \end{cases} \tag{14}$$

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

$$\text{(a) } w \in \mathcal{A}_v, \quad \text{(b) } \mathsf{Qr}(l) \le \mathsf{Qr}_{\mathsf{th}}, \ \text{(c) } \mathsf{LS}(l) \le \mathsf{LS}_{\mathsf{th}}, \tag{15}$$

The conditions (15a) – (15c) imply the QoS constraints (10b) – (10c) of the MEC problem. For the cost of route (12) 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 α_c and α_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 $w^* = \pi^*(\mathsf{s}_{\mathsf{src}})$, if the number of hops is greater than the constraint $\mathsf{hop}_{\mathsf{th}}$, the weights α_q and α_h of the reward function are adjusted by $\alpha_q = \alpha_q - \varepsilon$ 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 $\mathsf{a}_w \in \mathcal{A}_v$ to obtain the $Q-$function which is presented as follows:

$$\begin{aligned} Q(\mathsf{s}_v, \mathsf{a}_w) := (1-\alpha) Q(\mathsf{s}_v, \mathsf{a}_w) \\ + \alpha \Big(\mathsf{RW}(\mathsf{s}_v, \mathsf{a}_w) + \gamma \max_{a \in \mathcal{A}_w} \{ Q(\mathsf{s}_w, a) > 0 \} \Big), \end{aligned} \tag{16}$$

where α and γ is the learning rate and discount factor, respectively. We set $\max_{a \in \mathcal{A}_{\mathsf{dst}}} Q(\mathsf{s}_{\mathsf{dst}}, a) = 0$ 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 π^* that takes state s_v as input and returns the action to be taken by the agent. The policy π^* can be expressed as $\pi^*(\mathsf{s}_v) = w^* = \arg\max_{\mathsf{a}_w \in \mathcal{A}_v} Q^*(\mathsf{s}_v, \mathsf{a}_w)$. 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 $e_{t,v} = (\mathsf{s}_{t,v}, \mathsf{a}_{t,w}, r_t(v, w), \mathsf{s}_{t+1,w}, q_t(v, w))$, in a dataset $\mathcal{D} = \{e_1, \ldots, e_N\}$, where $r_t(v, w) = \mathsf{RW}_t(v, w)$ 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.
- For every generated environments, we use the Q-learning model to obtain optimal Q^*-values. A sample of datasets for DNN is generated as follows:

  - The input variables include the source's position, the destination's position, the current node v's position, and the information of neighbors nodes ($w \in \mathcal{V}$): position of w and reward $\mathsf{RW}(v, w)$. To fix the number of neighbor nodes for training process, we assign a maximum value to information value of node w, for all $w \notin \mathsf{NB}_v$. Thus, the total number of variables for the input of DNN is $50 \times 2 + 3 = 123$.
  - The output is a vector including $Q^*(\mathsf{s}_v, \mathsf{a}_w)$, $\forall w \in \mathcal{V}$, if node $w \notin \mathsf{NB}_v$, and we assign a maximum value to $Q^*(v, w)$. Thus, the number of variables for the output is 50.
- An environment provides 50 samples corresponding to the number of current states; thus, the obtained dataset has $200 \times 50 = 10,000$ samples.

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 links and avoiding regions of multiple primary users. For a multicast-tree $\mathtt{T}$, the GT-CTA model is proposed as a static best-response potential game [26] as follows:

- **Player:** Each node of the tree $\mathtt{T}$ is considered as a player.
- **Environment:** An agent at a certain time has an environment which includes the channel-time slot schedule of node v's neighbors and the affected regions of PUs.
- **Strategy:** A strategy of node v is defined as $\mathsf{s}_v = \mathsf{CT}_v = (\mathsf{ts}_v^{\mathsf{tx}} = t^{\mathsf{tx}}, \mathsf{ctx}_v^{t^{\mathsf{tx}}}, \mathsf{ts}_v^{\mathsf{rx}} = t^{\mathsf{rx}}, \mathsf{crx}_v^{t^{\mathsf{rx}}})$. The set of strategies of node v is denoted as $\mathcal{S}_v$. At the initial time, each node is assigned a strategy $\mathsf{s}^\infty = (-\infty, 0, -\infty, 0)$. For a strategy $\mathsf{s}_v \in \mathcal{S}_v$, we denote s_{-v} as the strategies of all agents except for agent v and $\mathcal{S}_{-v}$ as the set of all s_{-v}.
- **Strategy selection (SS) rules:** The game is operated into epochs. In an epoch, each node v observes the environment to calculate a set of strategies $\mathcal{S}_v$. If the parent w of node v has not already selected a strategy, i.e., $\mathsf{s}_w = \mathsf{s}^\infty$, the set of strategies $\mathcal{S}_v$ is assigned to $\emptyset$. Otherwise, a strategy s_v must satisfy the following rules:
  (i) The s_v does not affect to the licensed channel of PUs, i.e., $\mathsf{ctx}_v^{t^{\mathsf{tx}}}$ and $\mathsf{crx}_v^{t^{\mathsf{rx}}}$ are not in the affected region of licensed channels.
  (ii) The time slot $\mathsf{ts}_v^{\mathsf{tx}}$ must be greater than $\mathsf{ts}_w^{\mathsf{tx}}$, where w is the parent of v.
  (iii) The time slot $\mathsf{ts}_v^{\mathsf{rx}}$ must be the same as $\mathsf{ts}_w^{\mathsf{tx}}$, where w is the parent of v.
  (iv) Node v is the only transmitting neighbor of node v's children set except for children of node v in time slot $t = \mathsf{ts}_v^{\mathsf{tx}}$. It means that $\mathsf{AN}_{\mathsf{Child}_v} \cap \mathsf{TN}^t_{\mathsf{ctx}_v^t} = \{v\}$, where $\mathsf{AN}_{\mathsf{Child}_v} = \mathsf{NB}_{\mathsf{Child}_v} \setminus \mathsf{Child}_v$ which is the neighbors set of the node v's children set except for children of node v. This rule imply that each node v has priority to choose its strategy 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 $\mathsf{s}_v \in \mathcal{S}_v$ is defined as

$$\mathsf{RW}_v(\mathsf{s}_v, \mathsf{s}_{-v}) = \begin{cases} -\mathsf{ts}_v^{\mathsf{tx}}, & \text{if } \mathsf{s}_v \text{ satisfies (SS) rules,} \\ -\infty, & \text{otherwise.} \end{cases} \tag{17}$$

- **Best Response:** The best-response of node v can be expressed as

$$\pi_v(\mathsf{s}_{-v}) = \mathsf{s}_v^* = \arg\max_{\mathsf{s}_v \in \mathcal{S}_v} \mathsf{RW}_v(\mathsf{s}_v, \mathsf{s}_{-v}). \tag{18}$$

- **Potential function:** The potential function of the game can be defined as

$$\begin{aligned} \Phi :\ & \mathcal{CT} \to \mathbb{R} \\ & \mathsf{s}_\mathtt{T} \mapsto \Phi(\mathsf{s}_\mathtt{T}) = \min_{v \in \mathtt{T}} \mathsf{RW}(\mathsf{s}_v, \mathsf{s}_{-v}). \end{aligned} \tag{19}$$

where $\mathcal{CT}_\mathtt{T}$ is a set of channel-time slot strategies and $\mathsf{s}_T = \mathsf{CT}_T$ is a strategy of tree $\mathtt{T}$.

Theorem 1: The proposed game is the best-response potential game, i.e., we have that

$$\pi_v(\mathsf{s}_{-v}) = \arg\max_{\mathsf{s}_v \in \mathcal{S}_v} \Phi(\mathsf{s}_v, \mathsf{s}_{-v}). \tag{20}$$

Besides, the best-response of the game will converge to a Nash equilibrium point within $1 + M \times (N_{\mathsf{hop}} - 1)$ *iterations at most, where* N_{hop} *is the maximum number of hops of the multicast tree and* M *is the number of destinations of the multicast group.*

This theorem indicates that the game has a Nash equilibrium point and it will converge to a Nash equilibrium point within finite iterations. Moreover, the potential function (19) *is equivalent to the objective function* (13) *of the CTA problem at optimum; thus, the Nash equilibrium point of the best-response of the game is also a subset of the feasibility set of the CTA problem.*

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 $\mathsf{s}_v = \mathsf{CT}_v = (\mathsf{ts}_v^{\mathsf{tx}} = t^{\mathsf{tx}}, \mathsf{ctx}_v^{t^{\mathsf{tx}}}, \mathsf{ts}_v^{\mathsf{rx}} = t^{\mathsf{rx}}, \mathsf{crx}_v^{t^{\mathsf{rx}}})$, the strategy s_v satisfies the rule SS-(iv), i.e., s_v does not conflict with strategies of all neighbors except for 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 $\mathsf{s}_v \in \mathcal{S}_v$, $\Phi(\mathsf{s}_v, \mathsf{s}_{-v}) = \mathsf{RW}(\mathsf{s}_w, \mathsf{s}_{-w})$ with $w \neq v$, we have $\arg\max_{\mathsf{s}_v \in \mathcal{S}_v} \Phi(\mathsf{s}_v, \mathsf{s}_{-v}) = \mathcal{S}_v$. Thus, the condition (20) is satisfied.
  - If there exists a strategy $\mathsf{s}_v \in \mathcal{S}_v$ such that $\Phi(\mathsf{s}_v, \mathsf{s}_{-v}) = \mathsf{RW}(\mathsf{s}_v, \mathsf{s}_{-v})$ and $\max_{\mathsf{s}_v \in \mathcal{S}_v} \Phi(\mathsf{s}_v, \mathsf{s}_{-v}) = \mathsf{RW}(\mathsf{s}_v^*, \mathsf{s}_{-v}^*)$ with $\mathsf{s}_v^* \in \mathcal{S}_v$, we have that $\mathsf{RW}(\mathsf{s}_v^*, \mathsf{s}_{-v}^*)$ must be greater than or equal to $\mathsf{RW}(\mathsf{s}_v, \mathsf{s}_{-v})$. Thus, the condition (20) is satisfied.

– If there exists a strategy $\mathsf{s}_v \in \mathcal{S}_v$ such that $\Phi(\mathsf{s}_v, \mathsf{s}_{-v}) = \mathsf{RW}(\mathsf{s}_v, \mathsf{s}_{-v})$ and $\max_{\mathsf{s}_v \in \mathcal{S}_v} \Phi(\mathsf{s}_v, \mathsf{s}_{-v}) = \Phi(\mathsf{s}_v^*, \mathsf{s}_{-v}^*) = \mathsf{RW}(\mathsf{s}_w, \mathsf{s}_{-w})$ with $w \neq v$, we have that $\mathsf{RW}(\mathsf{s}_v^*, \mathsf{s}_{-v}^*)$ must be greater than or equal to $\mathsf{RW}(\mathsf{s}_v, \mathsf{s}_{-v})$ because if $\mathsf{RW}(\mathsf{s}_v^*, \mathsf{s}_{-v}^*) < \mathsf{RW}(\mathsf{s}_v, \mathsf{s}_{-v}) = \Phi(\mathsf{s}_v, \mathsf{s}_{-v}) < \Phi(\mathsf{s}_v^*, \mathsf{s}_{-v}^*)$, we have $\mathsf{RW}(\mathsf{s}_v^*, \mathsf{s}_{-v}^*) < \Phi(\mathsf{s}_v^*, \mathsf{s}_{-v}^*) = \mathsf{RW}(\mathsf{s}_w, \mathsf{s}_{-w})$ that contraries with the assumption. Thus, the condition (20) is satisfied.

- **Case 2:** The strategy s_v of node v conflicts with a strategy of a child w of node v. It means that the function Φ always takes $-\infty$ and $\arg\max_{\mathsf{s}_v \in \mathcal{S}_v} \Phi(\mathsf{s}_v, \mathsf{s}_{-v}) = \mathcal{S}_v$. Thus, the condition (20) is satisfied.

The second part: In the multicast tree, there is only the source, which transmits data to the next nodes at the first hop, that needs one time slot to transmit data by multicast technique. From the second hop of the multicast tree, the maximum number of links that can interfere with each other is N_{hop}; thus, 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 $1 + M \times (N_{\mathsf{hop}} - 1)$.

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 $1 + M \times (N_{\mathsf{hop}} - 1)$ 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 $w \in \mathsf{NB}_v$.

Step 2. Node v calculates the set of available strategies $\mathcal{S}_v$ 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 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 $\mathcal{D}$, 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 $\mathsf{LV}_{\mathrm{rt}} = \emptyset$.
  The route cost $\mathsf{RC}_{\mathrm{rt}} = +\infty$.
- **Step 1.** If a node needs to establish the tree to the multicast group $\mathcal{D}$, 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:

- **Step 2.** The src requires information of neighbors including position, speed, direction, queue size and channels of PUs information. For each destination $\mathsf{dst}_i \in \mathcal{D}$, the src predicts values $Q_i^*(\mathsf{src}, w)$ for all $w \in \mathsf{NB}_{\mathsf{src}}$ by using the DQN-MEC model to select the best neighbor w_i^* with the highest value $Q_i^*(\mathsf{src}, w_i^*)$; for example in Fig. 3, the best neighbors of src are nodes 3, dst_2, 15 and 16 corresponding to destinations dst_1, (dst_2, dst_3), dst_4 and dst_5. The src updates the set of last visit nodes $\mathsf{LV}_{\mathsf{src}} = \mathsf{LV}_{\mathrm{rt}} \cup \{\mathsf{src}\}$, the set of next visit nodes $\mathsf{NV}_{\mathsf{src}} = \{w_i^*, \ \forall \mathsf{dst}_i \in \mathcal{D}\} \setminus \mathsf{LV}_{\mathsf{src}}$, the list of costs from the src to all next visit nodes $\mathsf{CL}_{\mathsf{src}} = \{\mathsf{cost}(\mathsf{src}, w_i^*), \ \forall w_i^* \in \mathsf{NV}\}$ and the route cost $\mathsf{RC}_{\mathsf{src}} = 0$. Next, the src generates a route request (RREQ) packet including $\mathsf{LV}_{\mathrm{rreq}} = \mathsf{LV}_{\mathsf{src}}$, $\mathsf{NV}_{\mathrm{rreq}} = \mathsf{NV}_{\mathsf{src}}$, $\mathsf{CL}_{\mathrm{rreq}} = \mathsf{CL}_{\mathsf{src}}$ and $\mathsf{RC}_{\mathrm{rreq}} = \mathsf{RC}_{\mathsf{src}}$, and broadcasts the RREQ to neighbors. Go to Step 4. The RREQ packet contains the following fields:

$$\left\langle \begin{array}{c} \mathit{packet\_type}, \mathit{hop\_count}, \mathit{rreq\_id}, \\ \mathit{multicast\_IP\_address}, \\ \mathit{multicast\_seq\_number}, \\ \mathit{source\_IP\_address}, \mathit{source\_seq\_number}, \\ \mathit{last\_visit}, \mathit{next\_visit}, \mathit{link\_cost}, \mathit{route\_cost} \end{array} \right\rangle$$

Receiving RREQ Process:

- **Step 3.** 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 $\mathsf{NV}_{\mathrm{rreq}}$ of the RREQ.
    * The new cost $\mathsf{RC}_{\mathrm{rreq}} + \mathsf{cost}(w, \mathrm{node})$ is smaller than or equal to the route cost $\mathsf{RC}_{\mathrm{rt}}$ in the route table, where $\mathsf{cost}(w, \mathrm{node})$ can be found in the $\mathsf{CL}_{\mathrm{rreq}}$.

    For example in Fig. 3, nodes 9 and 10 drop a RREQ from the src. If the RREQ is dropped, the process is ended. Otherwise, go to Step 3.2.
  - **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.

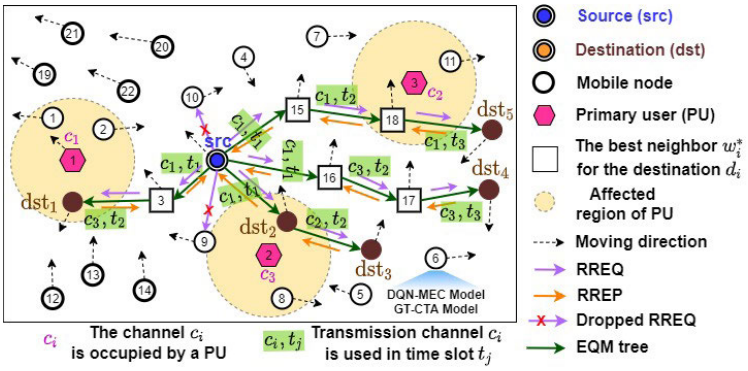

FIGURE 3. Illustration of the proposed DQMR protocol by using Figure in CR-MANETs.

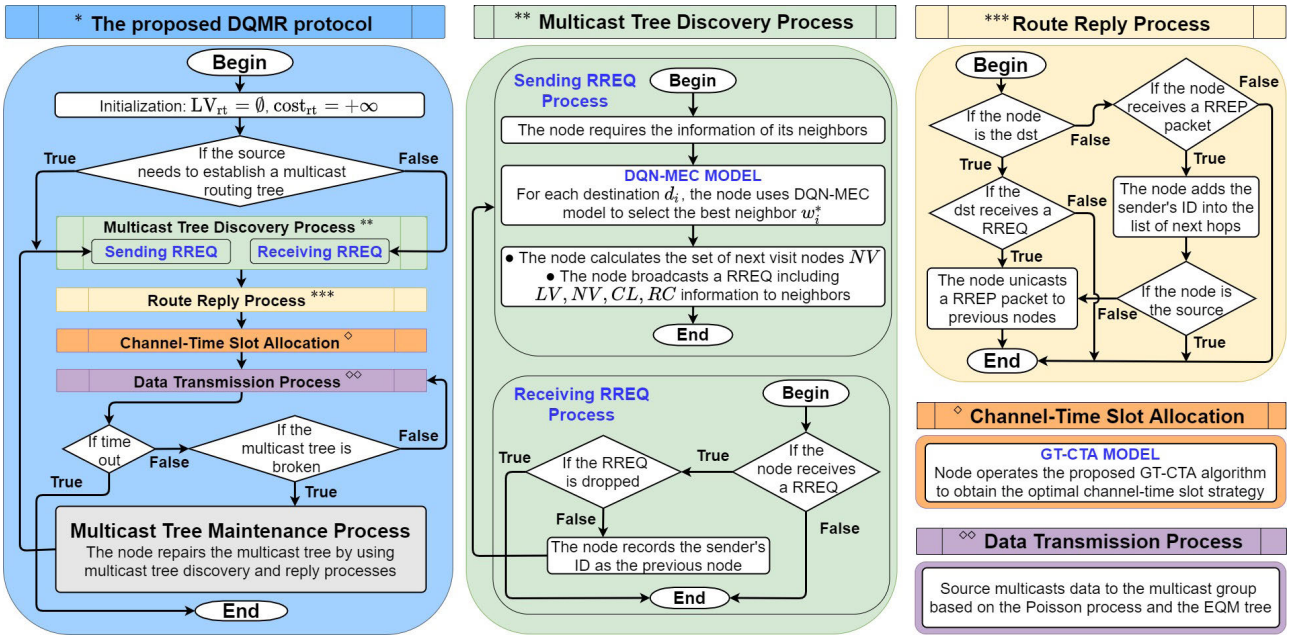

FIGURE 4. Illustration of the proposed DQMR protocol by using Flowchart: the proposed DQMR protocol *; the multicast tree discovery process **; the route reply process ***; the channel-time slot allocation ◇; data transmission process ◇◇.

- **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:

$$\left\langle \begin{array}{c} packet\_type, hop\_count, \\ multicast\_IP\_address, \\ multicast\_seq\_number, \\ source\_IP\_address \end{array} \right\rangle$$

- **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 multicasts data to the multicast group members 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 hops 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 $\mathsf{LV}_{\text{rreq}}$, $\mathsf{NV}_{\text{rreq}}$, and $\mathsf{CL}_{\text{rreq}}$. 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 replies 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 EQD-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 $(0, \mathsf{v}_{\max})$ 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 $\mathcal{A} = [0, 1]^2$ with area $S(\mathcal{A}) = 1km^2$ and nodes can move based on RWP model with the maximum speed $\mathsf{v}_{\max}$. We have

- The average distance between two nodes [27] is calculated by the expected distance between two independent points chosen uniformly at random in $\mathcal{A}$, which is $\overline{L}_{\mathcal{A}} = 0.521405$.
- The average number of nodes in a region $\mathcal{B} \subset \mathcal{A}$ is $\overline{\mathsf{N}}(\mathcal{B}) = N \times S(\mathcal{B})/S(\mathcal{A})$.
- The average speed of a node is $\overline{\mathsf{v}} = 0.5 \times \mathsf{v}_{\max}$.
- 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 $\overline{\alpha} = 2\pi/3$.
- 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

$$\mathsf{DD}(v, w, \Delta t) = |\mathsf{D}_{t_{i+1}}(v, w) - \mathsf{D}_{t_i}(v, w)|, \tag{21}$$

where $\mathsf{D}_{t_i}(v, w)$ and $\mathsf{D}_{t_{i+1}}(v, w)$ are the distances between node v and node w at time t_i and t_{i+1}, respectively.

Lemma 1: The average distance deviation ($\overline{\mathsf{DD}}$) between two nodes in an interval time Δt can be expressed as

$$\overline{\mathsf{DD}}(\mathsf{v}_{\max}, \Delta t) = |\sqrt{AX^2 + BX + C} - D|, \tag{22}$$

where $A = 0.75$, $B = 0.7821075$, $C = 0.271863$, $D = 0.521405$, $X = \mathsf{v}_{\max}\Delta t$ and $\mathsf{v}_{\max}$ is the maximum speed of each node.

Proof: Considering two nodes v and w with $\alpha_w = \alpha_v + \overline{\alpha} = \alpha_v + 2\pi/3$, $\mathsf{v}_v = \mathsf{v}_w = \overline{\mathsf{v}} = 0.5 \times \mathsf{v}_{\max}$, we have $\overline{\mathsf{DD}}(\mathsf{v}_{\max}, \Delta t) = \mathsf{DD}(v, w, \Delta t)$, 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

$$\begin{aligned}
x_{i+1}^{(v)} - x_{i+1}^{(w)} &= x_i^{(v)} - x_i^{(w)} + \overline{\mathsf{v}}\Delta t(\cos\alpha_v - \cos\alpha_w) \\
&= x_i^{(v)} - x_i^{(w)} + 1.5\overline{\mathsf{v}}\Delta t \\
y_{i+1}^{(v)} - y_{i+1}^{(w)} &= \overline{\mathsf{v}}\Delta t(\sin\alpha_v - \sin\alpha_w) \\
&= -0.866025\overline{\mathsf{v}}\Delta t, \qquad (23) \\
\mathsf{D}_{t_i}(v, w) &= \overline{L}_{\mathcal{A}} = 0.521405, \\
\mathsf{D}_{t_{i+1}}(v, w) &= \sqrt{(\overline{L}_{\mathcal{A}} + 1.5\overline{\mathsf{v}}\Delta t)^2 + (0.866025\overline{\mathsf{v}}\Delta t)^2} \\
&= \sqrt{A\mathsf{v}_{\max}^2\Delta t^2 + B\mathsf{v}_{\max}\Delta t + C}, \qquad (24)
\end{aligned}$$

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 Δt is

$$\overline{\mathsf{N}}_{\text{rwp}}^{\text{out}}(X) = (\overline{\mathsf{Nb}}_v + 1)\frac{R^2 - (R - \overline{\mathsf{DD}}(X))^2}{R^2}, \tag{25}$$

where $X = (\mathsf{v}_{\max}, \Delta t)$, $\mathsf{v}_{\max}$ is the maximum speed of each node, R is the transmission range of each node, and $\overline{\mathsf{Nb}}_v$ is the average number of node v's neighbors which can be expressed as

$$\overline{\mathsf{Nb}}_v = N\frac{\pi R^2}{S(\mathcal{A})} - 1. \tag{26}$$

Proof: Lemma 2 can be easily proved based on Lemma 1. □

Lemma 3: The average number of packets in a node is

$$\overline{\mathcal{N}}_{rwp}(\mathsf{v}_{\max}, \Delta t) = \overline{\mathcal{N}} + \frac{\lambda}{\overline{\mathsf{Nb}}_v}\Delta t\overline{\mathsf{N}}_{\text{rwp}}^{\text{out}}(\mathsf{v}_{\max}, \Delta t), \tag{27}$$

where λ is the arrival rate of queuing delay model, $\overline{\mathcal{N}}$ is the average number of packets in the system, $\mathsf{v}_{\max}$ is the maximum speed of each node, Δt is the maximum lifetime each packet and $\overline{\mathsf{Nb}}_v$ 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 cannot 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. □

For a tree $\mathtt{T} = \{\mathtt{P}_1 = \mathtt{P}(\mathsf{src}, \mathsf{dst}_1), \ldots, \mathtt{P}_M = \mathtt{P}(\mathsf{src}, \mathsf{dst}_M)\}$ in (8), where src is the source, dst_i is a destination belonging to multicast group $\mathcal{D}$, and M is the number of destinations. The E2E queuing delay of the tree $\mathtt{T}$ can be represented as

$$\mathsf{EQD}(\mathtt{T}) = \frac{1}{M}\sum_{i=1}^{M}\sum_{n_i \in \mathtt{P}_i} \mathsf{q\_delay}(n_i), \tag{28}$$

where $\mathsf{q\_delay}(n_i)$ is the queuing delay of node n_i.

Theorem 2: We assume that the maximum lifetime of a packet is Δt. When a new routing packet arrives at a node at a certain time, the average time of this packet spending in this node is

$$\mathcal{T}_{rwp}(\mathsf{v}_{\max}, \Delta t) = (\overline{\mathcal{N}}_{rwp}(\mathsf{v}_{\max}, \Delta t) + 1)/\mu, \tag{29}$$

where μ is service time rate of the queuing delay model, the $\mathsf{v}_{\max}$ is the maximum speed of each node, $\overline{\mathcal{N}}_{rwp}$ is the average number of packets in a node which is presented as (27).

As a consequence, the E2E queuing delay of a tree $\mathtt{T}$ can be calculated as

$$\overline{\mathsf{EQD}}_{rwp}(\mathtt{T}) = (n_{\mathsf{hop}} + 1)\mathcal{T}_{rwp}(\mathsf{v}_{\max}, \Delta t), \tag{30}$$

where n_{hop} is the average $\#\mathsf{hops}$ of routes of the tree $\mathtt{T}$.

Proof: Theorem 2 can be proved by using the results of Lemmas 1, 2, 3 and (28). □

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

We present the analysis of EQD-MRT in the environment of RPGM model. As shown in Fig. 3, nodes 12 to 22 are divided into three groups and move according to the RPGM 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 $\mathcal{A}$ 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 Δt is

$$\overline{\mathsf{N}}^{\mathrm{out}}_{\mathrm{rpgm}}(X) = \overline{\mathsf{N}}^{\mathrm{out}}_{\mathrm{rwp}}(X)\frac{\overline{\mathrm{Nb}}^{\mathrm{os}}_v}{\overline{\mathrm{Nb}}_v}, \tag{31}$$

where $X = (\mathsf{v}_{\max}, \Delta t)$, $\mathsf{v}_{\max}$ is the maximum speed of each node and $\overline{\mathrm{Nb}}^{\mathrm{os}}_v$ 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 $\mathcal{D}_G$ with center v_0 and radius $R_G = \sqrt{S(\mathcal{A})/(K\pi)}$ while the transmission region of node v is a disc $\mathcal{D}_v$ with center v and radius $R_v = R$. The region $\mathcal{D}_v \backslash (\mathcal{D}_G \cap \mathcal{D}_v)$ includes nodes which are called outside neighbors of node v. The average number of outside neighbors of node v can be expressed as follows:

$$\overline{\mathrm{Nb}}^{\mathrm{os}}_v = N\frac{S(\mathcal{D}_v \setminus (\mathcal{D}_G \cap \mathcal{D}_v))}{S(\mathcal{A})}. \tag{32}$$

The average distance between node v and the center v_0 of $\mathcal{D}_G$ (the distance between two centers of $\mathcal{D}_G$ and $\mathcal{D}_v$) is $\overline{d} = 2R_G/3$. Since node v is in $\mathcal{D}_G$, the value $R_G + R_v$ is always greater than or equal $\overline{d}$, i.e., $R_G + R_v \geq \overline{d}$. We have the following cases:

- If the region $\mathcal{D}_v$ is a subset of the region $\mathcal{D}_G$, i.e., $R_G - R_v > \overline{d}$,

$$S(\mathcal{D}_v \setminus (\mathcal{D}_G \cap \mathcal{D}_v)) = S(\emptyset) = 0. \tag{33}$$

- If the region $\mathcal{D}_G$ is a subset of the region $\mathcal{D}_v$, i.e., $R_v - R_G > \overline{d}$,

$$S(\mathcal{D}_v \setminus (\mathcal{D}_G \cap \mathcal{D}_v)) = S(\mathcal{D}_v \setminus \mathcal{D}_G) = \pi(R_v^2 - R_G^2). \tag{34}$$

- If two regions $\mathcal{D}_v$ and $\mathcal{D}_G$ are overlapped, i.e., $|R_G - R_v| < \overline{d}$, we have

$$S(\mathcal{D}_v \setminus (\mathcal{D}_G \cap \mathcal{D}_v)) = S(\mathcal{D}_v) - (A + B - C), \tag{35}$$

where

$$\begin{aligned} A &= R_{\min}^2 \cos^-1((\overline{d}^2 + R_{\min}^2 - R_{\max}^2)/(2\overline{d}R_{\min})),\\ B &= R_{\max}^2 \cos^-1((\overline{d}^2 + R_{\max}^2 - R_{\min}^2)/(2\overline{d}R_{\max})),\\ C &= 0.5\sqrt{abcd}, \end{aligned} \tag{36}$$

$a = (-d + R_{\min} + R_{\max})$, $b = (d + R_{\min} - R_{\max})$, $c = (d - R_{\min} + R_{\max})$, $d = (d + R_{\min} + R_{\max})$, $R_{\min} = \min\{R_G, R_v\}$ and $R_{\max} = \max\{R_G, R_v\}$.

Moreover, node v's outside neighbors can be considered as neighbors that move based on RWP model related to node v. The number $\overline{\mathrm{Nb}}^{\mathrm{out}}_v$ 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.,*

$$\overline{\mathcal{N}}_{rpgm}(\mathsf{v}_{\max}, \Delta t) = \overline{\mathcal{N}} + \frac{\lambda}{\overline{\mathrm{Nb}}_v}\Delta t\overline{\mathsf{N}}^{\mathrm{out}}_{\mathrm{rpgm}}(\mathsf{v}_{\max}, \Delta t). \tag{37}$$

- *The average time of this packet spending in a node is calculated the same as in* (29)*, i.e.,*

$$\mathcal{T}_{rpgm}(\mathsf{v}_{\max}, \Delta t) = (\overline{\mathcal{N}}_{rpgm}(\mathsf{v}_{\max}, \Delta t) + 1)/\mu. \quad (38)$$

- *The E2E queuing delay of a tree* $\mathtt{T}$ *is calculated by same as in* (30)*, i.e.,*

$$\overline{\mathsf{EQD}}_{rpgm}(\mathtt{T}) = (n_{\mathsf{hop}} + 1)\mathcal{T}_{rpgm}(\mathsf{v}_{\max}, \Delta t). \quad (39)$$

VIII. PERFORMANCE EVALUATION

A. ENVIRONMENTS FOR PERFORMANCE EVALUATION

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

TABLE 2. Simulation environments and parameters.

Network size	$1000 \times 1000m^2$
Number of nodes	50
Mobility model	RWP and RPGM
Max speed of nodes	$20km/h \sim 80km/h$
Number of PU	$3 \sim 5$
Transmission range of nodes	$250m$
Coverage range of PU	$250m$
Number of licensed channels	5
Queue arrival rate	$\lambda = 10\ packets/s$
Queue service time rate	$\mu = 20\ packets/s$
Data transmission process	Poisson
Data transmission rate	$5\ packets/s$
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 $1,000$ epochs 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 $1 + M \times (N_{\mathsf{hop}} - 1) = 1 + (4 - 1) * 5 = 16$ iterations, where $N_{\mathsf{hop}} = 4$ is the maximum number of hops, $M = 5$ is the number of destinations. This result shows that the proposed game with GT-CTA model achieves the optimal solution of CTA 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.

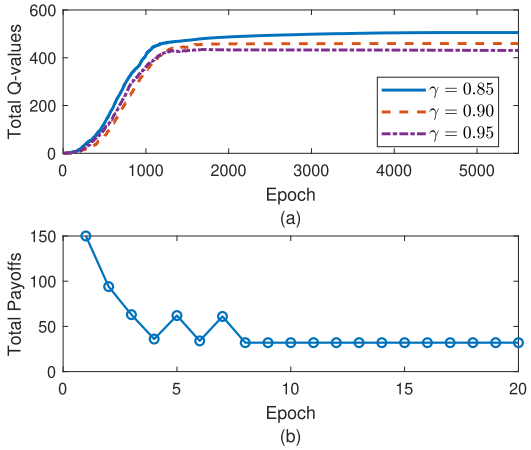

FIGURE 5. Total Q-values and payoffs convergences of the proposed DQN-MEC (a) and GT-CTA (b) models, respectively.

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.

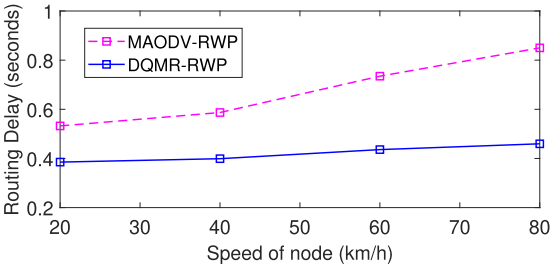

FIGURE 6. Routing delay with 3 PUs as a function of node speed for RWP model.

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 scenarios 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.

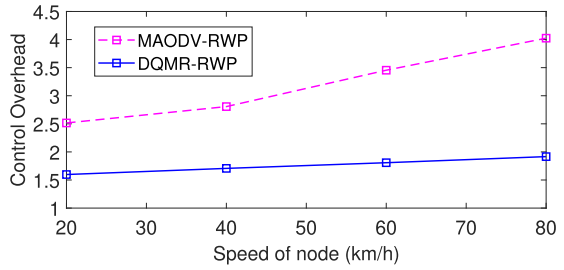

FIGURE 7. Control overhead with 3 PUs as a function of node speed for RWP model.

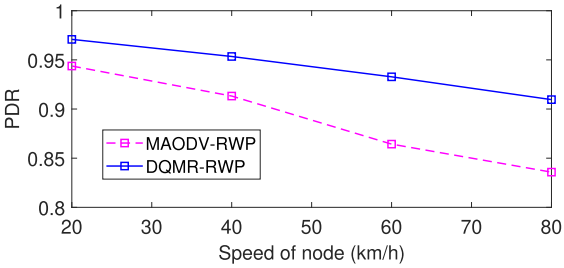

FIGURE 8. PDR with 3 PUs as a function of node speed for RWP model.

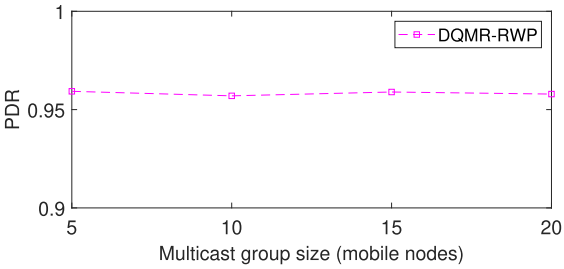

FIGURE 9. Scalability with 3 PUs and 50 km/h as a function of multicast group size for RWP model.

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.

Fig. 8 shows the PDR of protocols with 3 PUs as a function of node speed for RWP model. As can be observed, at the maximum speed of 80 km/h, the DQMR protocol achieves about 91% while the MAODV-based protocol is only around 84%. The reason is that the DQMR protocol provides EQM trees with high stability and optimal channel-time slot strategies that helps data to reach the destination faster and more reliable than MAODV-based protocol.

In Fig. 9, we show the scalability of the DQMR protocol by demonstrating the PDR as a function of multicast group size (number of destinations) 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.

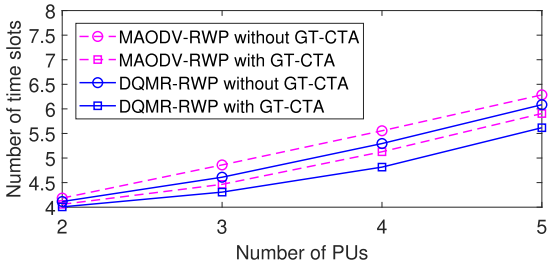

FIGURE 10. Number of time slots with 50 km/h and 5 licensed channels as a function of number of PUs for RWP model.

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 consume 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-routing 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.

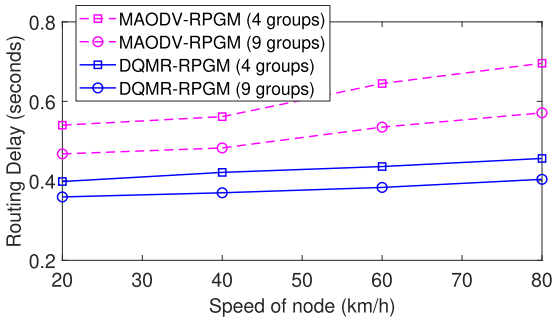

FIGURE 11. Routing delay with 3 PUs as a function of node speed for RPGM model.

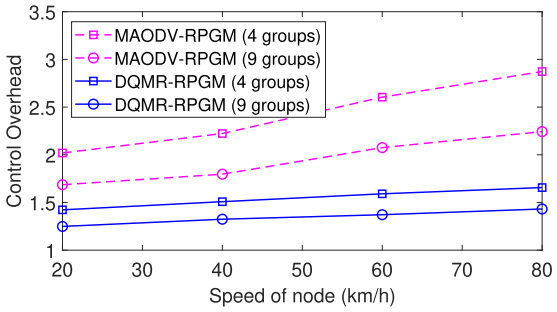

FIGURE 12. Control overhead with 3 PUs as a function of node speed for RPGM model.

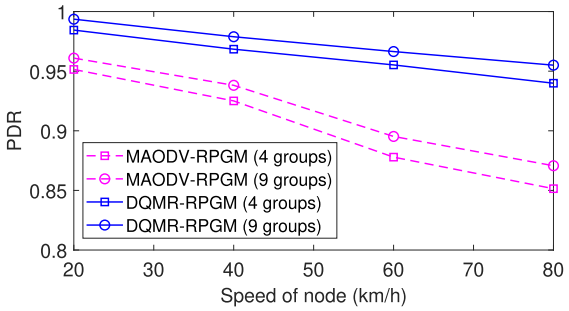

FIGURE 13. PDR with 3 PUs as a function of node speed for RPGM model.

Fig. 13 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. 14, we show the scalability of the DQMR protocol by demonstrating the PDR as a function of multicast group size for RPGM model. As can be observed, the PDR has almost constant value and is not affected by the number of destinations. The reason is that the DQMR protocol applies the DQN-MEC and GT-CTA models to obtain EQM trees

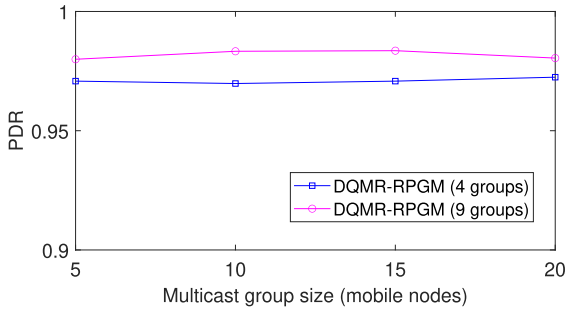

FIGURE 14. Scalability with 3 PUs and 50 km/h as a function of multicast group size for RPGM model.

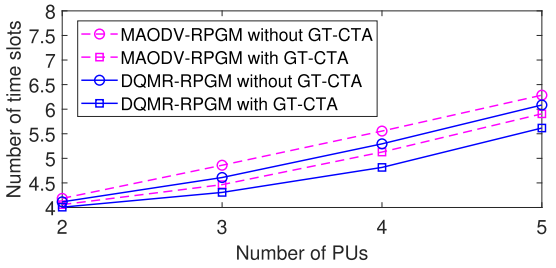

FIGURE 15. Number of time slots with 50 km/h and 5 licensed channels as a function of number of PUs for RPGM model.

that can help the DQMR protocol to achieve the stability and scalability under different sizes of multicast group.

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 theory approach in Section V, which helps to improve the resource utilization of DQMR protocol.

F. ANALYSIS RESULTS OF DELAY: EQD-MRT

We present 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 several 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 shows 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 $4\mathcal{T}_{rwp}$ 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 routing delay in terms of node speed.

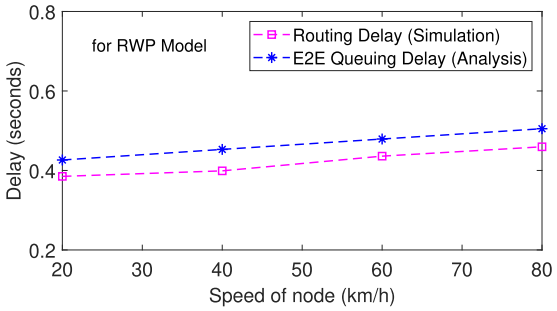

FIGURE 16. The EQD-MRT of the proposed DQMR protocol as a function of node speed for RWP model.

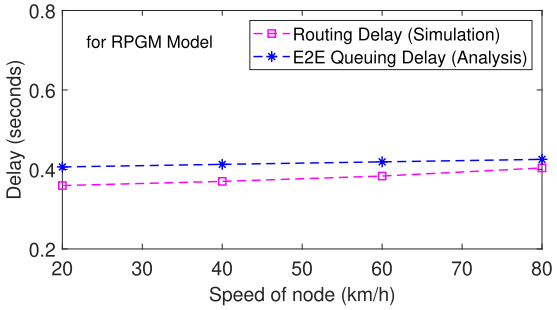

FIGURE 17. The EQD-MRT of the proposed DQMR protocol as a function of node speed for RPGM model.

Fig. 17 presents the analysis of E2E queuing delay (EQD-MRT) with the comparison of the simulation result for RPGM 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 $4\mathcal{T}_{rpgm}$ in (39) of Corollary 2. The analytical result of the EQD-MRT also 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.

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. CONCLUSION

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 were 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.

ACKNOWLEDGMENT

An earlier version of this paper was presented in part at the 2021 International Conference on Electronics, Information, and Communication (ICEIC), Jeju Shinhwa World, Republic of Korea, in January 31, 2021–February 3, 2021 [DOI: 10.1109/ICEIC51217.2021.9369756], and in part at the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Republic of Korea, in April 13, 2021–April 16, 2021 [DOI:10.1109/ICAIIC51459.2021.9415188].

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 networks," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 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 *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Apr. 2021, pp. 279–283.

[3] F. R. Yu and H. Tang, *Cognitive Radio Mobile Ad Hoc Networks*, vol. 507. New York, NY, USA: Springer, 2011.

[4] I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Netw.*, vol. 7, pp. 810–836, Jul. 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. 2013.

[7] C. Zhu, "Medium access control and quality-of-service routing for mobile ad hoc networks," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Maryland, College Park, MD, USA, 2001.

[8] C. Zhu and M. S. Corson, "QoS routing for mobile ad hoc networks," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Soc. IEEE INFOCOM Conf. Comput. Commun.*, vol. 2, Jun. 2002, pp. 958–967.

[9] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," *IEEE Access*, vol. 7, pp. 55916–55950, 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, Apr. 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. Informat.*, vol. 16, no. 8, pp. 5617–5625, Aug. 2020.

[12] W. Sheikh, "A resource-tuned QoS multicast routing protocol," *Int. J. Commun. Syst.*, vol. 31, no. 11, p. e3570, Jul. 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. 115109–115121, 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.*, vol. 69, no. 6, pp. 6718–6730, Jun. 2020.

[15] A. Serhani, N. Naja, and A. Jamali, "QLAR: A Q-learning based adaptive routing for MANETs," in *Proc. IEEE/ACS 13rd Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2016, pp. 1–7.
[16] T. Hendriks, M. Camelo, 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)*, Oct. 2018, pp. 108–115.
[17] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, and Y. Xu, "QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks," *Comput. Commun.*, vol. 150, pp. 304–316, Jan. 2020.
[18] E. M. Royer and C. E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *Proc. 5th Annu. ACM/IEEE Int. Conf. Mobile Comput. Netw. (MobiCom)*. New York, NY, USA: Association for Computing Machinery, 1999, pp. 207–218, doi: 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. 2nd ACM Int. Workshop Modeling, Anal. Simulation Wireless Mobile Syst. (MSWiM)*. New York, NY, USA: Association for Computing Machinery, 1999, pp. 53–60, doi: 10.1145/313237.313248.
[20] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, Sep. 2002.
[21] B. An and S. Papavassiliou, "A mobility-based hybrid multicast routing in mobile ad-hoc wireless networks," in *Proc. Commun. Netw.-Centric Oper., Creating Inf. Force (MILCOM)*, vol. 1, 2001, pp. 316–320.
[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 Database Expert Syst. Appl.*, 2004, pp. 705–709.
[23] K. R. Chowdhury and M. D. Felice, "Search: A routing protocol for mobile cognitive radio ad-hoc networks," *Comput. Commun.*, vol. 32, no. 18, pp. 1983–1997, Dec. 2009.
[24] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 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. 141310–141330, 2020.
[26] M. Voorneveld, "Best-response potential games," *Econ. Lett.*, vol. 66, no. 3, pp. 289–295, Mar. 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, Jul. 2003.

THONG-NHAT TRAN (Graduate Student Member, IEEE) received the B.S. degree in mathematics-informatics and the M.S. degree in mathematics from Dalat University, Vietnam, in 2002 and 2006, respectively. He is currently pursuing the Ph.D. degree in electronics and computer engineering with Hongik University, Republic of Korea. His current research interests include clustering, routing, optimization, and machine learning for wireless networks.

TOAN-VAN NGUYEN (Member, IEEE) received the B.S. degree in electronics and telecommunications engineering and the M.S. degree in electronics engineering from the HCMC University of Technology and Education, Vietnam, in 2011 and 2014, respectively, and the Ph.D. degree in electronics and computer engineering from Hongik University, Republic of Korea, in 2021. He is currently a Postdoctoral Researcher with the Electrical and Computer Engineering Department, Utah State University, Logan, UT, USA. His current research interests include mathematical modeling of 5G networks and machine learning for wireless communications.

KYUSUNG SHIM (Member, IEEE) received the bachelor's degree in computer and information communications engineering and the M.S. and Ph.D. degrees in information system and electronics and computer engineering from Hongik University, Sejong, Republic of Korea, in 2012, 2017, and 2021, respectively. He is currently a Postdoctoral Researcher at Hongik University. His research interests include wireless communication, physical layer security, *ad-hoc* networks, and routing protocol.

DANIEL BENEVIDES DA COSTA (Senior Member, IEEE) was born in Fortaleza, Ceará, Brazil, in 1981. He received the B.Sc. degree in telecommunications from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil, in 2003, and the M.Sc. and Ph.D. degrees in electrical engineering, area: telecommunications from the University of Campinas, São Paulo, Brazil, in 2006 and 2008, respectively. From 2008 to 2009, he was a Postdoctoral Research Fellow with INRS-EMT, University of Quebec, Montreal, QC, Canada. Since 2010, he has been with the Federal University of Ceará, where he is currently an Associate Professor. Recently, he joined as a Full Professor at the National Yunlin University of Science and Technology (YunTech), Taiwan. He is the Founder and the Head of the Intelligent Wireless Communications (IWiCom) Research Group, the first research group created under the umbrella of the Future Technology Research Center (a new landmark at the YunTech Campus, since 2020). His Ph.D. thesis was awarded as the Best Ph.D. Thesis in Electrical Engineering by the Brazilian Ministry of Education (CAPES) at the 2009 CAPES Thesis Contest. In 2019, he was awarded with the prestigious Nokia Visiting Professor Grant.

BEONGKU AN (Member, IEEE) received the B.S. degree in electronic engineering from Kyungpook National University, Republic of Korea, in 1988, the M.S. degree in electrical engineering from Polytechnic University (NYU), NY, USA, in 1996, and the Ph.D. degree from the New Jersey Institute of Technology (NJIT), NJ, USA, in 2002. After graduation, he joined the Faculty of the Department of Software and Communications Engineering, Hongik University, Republic of Korea, where he is currently a Professor. From 1989 to 1993, he was a Senior Researcher at RIST, Pohang, Republic of Korea. He was also a Lecturer and a RA at NJIT, from 1997 to 2002. His current research interests include mobile wireless networks and communications, such as *ad-hoc* networks, sensor networks, wireless cognitive radio networks, and cellular networks. In particular, he is interested in cooperative communication, multicast routing, QoS routing, energy harvesting, physical layer security, M2M/D2D, the IoT, visible light communication (VLC), crosslayer technology, 5G/beyond 5G, NOMA, SWIPT, machine learning/deep learning, block chain, and mobile cloud computing. He was listed in Marquis Who's Who in Science and Engineering and Marquis Who's Who in the World. He was the President of IEIE Computer Society (The Institute of Electronics and Information Engineers, Computer Society), in 2012. Since 2013, he has been working as the General Chair of the International Conference on Green and Human Information Technology (ICGHIT).