This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

Deep Q-Learning Based Node Positioning for Throughput-Optimal Communications in Dynamic UAV Swarm Network

Koushik A M [1], Fei Hu[1], *Member, IEEE,* Sunil Kumar[2], *Senior Member, IEEE*
[1]Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL
[2]Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA

***Abstract*—In this paper, we study the communication-oriented UAV placement issue in a typical Manned-and-UnManned (MUM) airborne network. The MUM network consists of a few powerful aircraft nodes in the higher layer and high-density unmanned air vehicles (UAVs) in the lower layer. While the aircraft network is relatively stable, the UAVs can form different swarm network topologies. Some UAVs are selected as gateway nodes to aggregate the received UAV data and send to a nearby aircraft which acts as a control node for the UAVs in a swarm. Assume a source UAV has data to be sent to its gateway node by using a route which may have broken links. Our goal is to guide the position of one or more relay UAVs to make up for the broken wireless links under the dynamic swarm topology. The placement of the relay node is determined by both traffic quality-of-service (QoS) requirements and the link conditions. We design a new queueing model, called multi-hop priority queue, to analyze the achievable QoS performance through multi-hop queue-to-queue accumulation modeling. To handle dynamic swarm topology and time-varying link conditions, we design a deep Q-learning (DQN) model to determine the optimal link between two UAV nodes, and then use an optimization algorithm to locally fine-tune the position of the UAV node to optimize the overall network performance. The DQN-based UAV link selection is computed in the powerful aircraft (control node) which maintains the graphs of the swarm topology, where the optimization is implemented at the UAV. Our simulation results validate the throughput efficiency of our DQN-based UAV positioning scheme.**

***Index Terms*—UAV Swarming, Deep Q-Learning Network (DQN), Deep Learning, Relay Placement, Multi-Hop Queueing Model, Manned-and-UnManned (MUM) Network**

I. Introduction

A. UAV Swarm Network

Unmanned aerial vehicles (UAVs) equipped with different types of sensors are useful in a wide variety of applications, including the surveillance, target tracking, search and rescue, and damage assessment. The swarm of micro UAVs [1] consisting of autonomous UAVs is inherently resilient and scalable as more nodes can be added and the redundant nodes removed. Effective communication among UAVs is required in a swarm to coordinate and achieve the desired tasks [2].

In this paper, we control the position of some UAVs (known as relay UAVs) to compensate for the broken RF links and achieve an end-to-end packet delivery in the UAV swarm network. To achieve efficient UAV swarm management, a hierarchical airborne network architecture is used, as shown in Fig. 1. It is also called the *manned-and-unmanned (MUM)* network due to the use of both *manned* aircrafts (in higher layer) and *unmanned* UAVs (in lower layer). The aircrafts have powerful computational capabilities and can communicate over long distances. These aircrafts are chosen as the *control node*, which can run computation-intensive algorithms (such as deep learning) in real-time. The UAVs can form different swarm topologies based on their mission and commands from the control node in the higher layer.

To simplify the management, the control node communicates only with a small number of selected UAVs, called gateway nodes (see the star nodes in Fig. 1). These gateway nodes can be the center of different swarms if a multi-swarm architecture is used. They typically have relatively less mobility and higher communication capability. The gateway UAVs can operate in multiple frequency bands. For example, one band can be used for UAV-to-UAV and the other for UAV-to-aircraft communication.

If a UAV has important event data to report, it will search for a shortest-hop route to reach the closest gateway. However, due to the dynamic swarm topology, the source UAV may not be able to find a 'good' 1-hop neighbor that has the shortest and high quality route to reach the gateway UAV. Therefore we use relay UAV node(s) in that sparse area to serve as the communication 'bridge'. Without such a relay node, the source node may need to forward its packets over a much longer route, which is not desirable for quality-of-service (QoS)-oriented applications.

B. Problem Statement

The goal of this paper is to find the optimal position for the relay nodes such that the QoS requirements of UAV swarm applications are met. We illustrate the challenges of our problem with the help of two examples below.

In the left part of Fig. 1, the Source 1 wants to send its text data to gateway node G1 but it does not have a good 1-hop neighbor. Assuming that the control node in the higher layer maintains the topology map of entire UAV swarm network, it guides a nearby relay UAV node (denoted as *A*) to move to a suitable location to serve as a relay. Here, node A has two choices: It can move to the P1 (or P2) location to relay the source data along Path 1 (or Path 2). Here, the control node needs to make a choice (P1 or P2) based on both paths'

Corresponding author: Dr. Fei Hu, email: fei@eng.ua.edu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

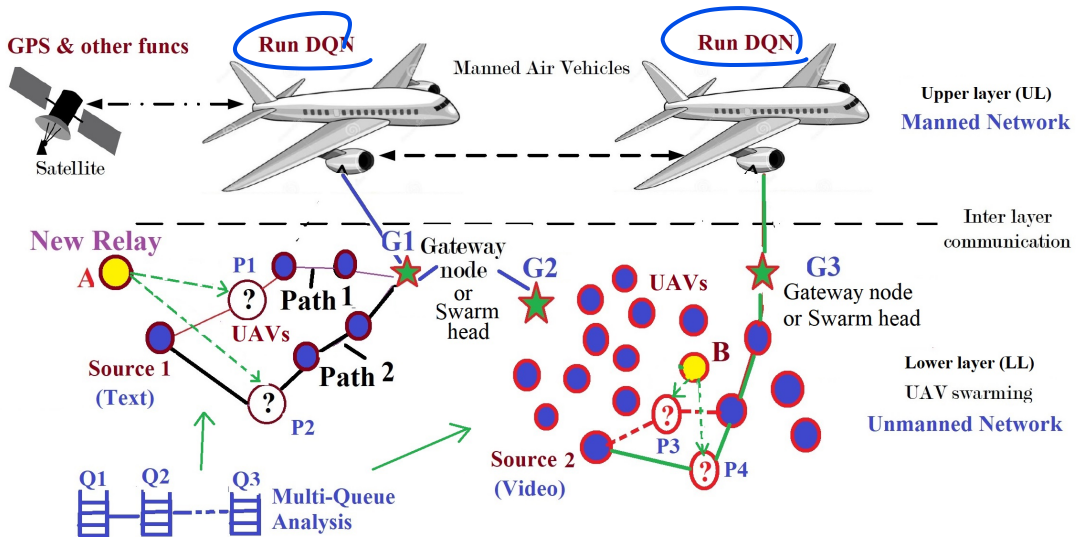

Fig. 1: Architecture of a UAV swarming network.

QoS support capabilities. Both the coarse and fine location information is needed where the coarse location tells which path/link the relay should be placed to and the fine location tells the exact location in that link.

In the right part of Fig. 1, the Source 2 needs a relay node in its 1st link, in order to send its video data to gateway G3. Here the location of relay node needs to be determined carefully. The P3 location for the relay node, which is close to a high-density swarm region, is likely to experience congestion. In contrast, the P4 location is near a sparse network region that may not generate much data traffic. From congestion avoidance viewpoint, P4 is a better location than P3.

Note that the need for a relay node can arise not only due to the long distance between two nodes but also due to strong interference experienced by the link. To get around a strong interference region, one or more relay nodes may be needed to form a detour path away from the interference region.

The relay UAV positioning can be modelled as an optimization problem. But the optimization algorithm might get stuck in local optima due to time-varying swarm network topology. Since the formation of swarming network may change with time, the relay node(s) should be repositioned to maintain the communication performance. Hence, the optimization process should be performed in stages to establish and maintain the throughput-optimal communication. It should also consider the impact of relay node's position in a specific stage on the overall throughput performance across all the stages. In addition, since UAV are power limited, running complex optimization algorithms on UAV nodes is not feasible.

C. Research Novelty

The first novelty of our work is the use of *multi-protocol-layer (MPL)* parameters for identification of UAV swarming *state*. The benefit of using different parameters from physical, data link and routing layers is that they comprehensively capture the wireless network conditions from both the radio signal and packet flow viewpoint. Moreover, these parameters can capture the single-link as well as multi-hop route variations. Our MPL parameters include: (1) In the physical layer, the signal-to-interference-noise-ratio (SINR) is used to represent the signal quality in each link. The *interference graph (IG)* is built at the control node to reflect the SINR distribution in all the links. (2) In the data link layer, the bit error rate (BER) is used. A well-designed medium access control (MAC) protocol can minimize the channel access collisions and thus reduce the BER. (3) In the routing layer, the packet drop rate (PDR) and *routing topology graph (RTG)* are used as the state parameters of the entire UAV network. The RTG is also maintained by the control node.

The second novelty of our work is the design of a *multi-hop queuing model with M/G/1 Preemptive Repeat Priority (MHQ-PRP)*. The queueing model is an important tool for calculating the packet transmission latency and PDR as it can quantify the queue congestion status in each node and the data delivery efficiency. Most conventional queueing models [3] [4] can only reflect the queue dynamics in a single node. The advantages of using a multi-hop queueing (MHQ) model are: *First*, it can be used to analyze the correlations among different neighboring queues. *Second*, it can help a source UAV to choose the best position for relay node(s) by comparing the total transmission delay in different multi-node combinations. For example, the combination of nodes 'source-P1-G1' (i.e., Path 1 in Fig. 1) may be better than the combination of 'source-P2-G1' (i.e., Path 2). *Third*, it can effectively determine the interference experienced in each link. The MHQ helps to compare the regional interference levels by co-analyzing the multiple nodes' queues. We build MHQ-PRP to reflect different priority and QoS requirements of data flows.

Since the optimization algorithm alone may not achieve the optimal relay positioning, **the third novelty** of our research is to integrate an optimization model with deep Q-learning network (DQN)-based algorithm for determining the relay UAV position. The relay node positioning is performed in two phases: One of the links between UAV nodes is first selected as the candidate link for placing the relay UAV node. The relay node then performs local optimization to find an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

optimal position within the link to improve the communication performance.

The use of DQN has two benefits: First, DQN has a deep learning module [5], which can be used to accurately identify the swarm network state based on the comprehensive analysis of network parameters. Here, the hidden layers of the convolutional neural network (CNN) provide a high-resolution pattern recognition based on the large number of feature parameters in the input layer. Second, DQN has a reinforcement learning (RL)-compatible 'action' generation engine, which can search a globally optimized policy to find the most suitable *action* based on the learned *state*. Here the 'action' refers to the selection of a suitable link for placing the relay node between two UAV nodes. The use of DQN for link selection and the optimization for accurate relay location determination is **a unique feature** of our proposed scheme, compared with conventional relay placement schemes such as [6], [7], [8].

The DQN model selects a suitable link by considering the MAC and physical layer parameters at each link. Since our model also considers the network layer information, it can capture the information and irregularities at each layer for every link. Hence, the PDR seen at every link also acts as an input parameter at every stage of decision making in DQN algorithm.

Traditional Q-learning [9] algorithms are not able to keep track of different UAV swarming patterns and channel conditions. Similarly, the algorithms we proposed in [10] [11] cannot learn the changes in the graph patterns. Hence, we adopt the *memory replay* concept [12], i.e., different patterns of the UAV swarming and channel conditions are recorded and used to retrain the convolutional layer when the UAV is unable to choose an optimal action. The *replay buffer* is maintained at the control node, and the optimal Q-function values are sent to UAV nodes to perform an optimal action. By doing so, the overhead of finding the optimal Q-function in a large swarm network is reduced.

Note that there is a close relationship between our proposed queueing model and DQN-based node positioning scheme. This is because DQN is just a machine learning algorithm to recognize the network patterns and make decisions. It cannot run without knowing the input parameters. Hence we use the queueing model's results to serve as the DQN input. Especially, since our multi-hop queueing model includes the information from both the routing layer (such as multi-hop architecture) and MAC layer parameters (such as link SINR), we use the queueing results (such as queueing delay and traffic service rate) as inputs to DQN. Therefore, it is necessary to integrate the DQN and queueing models together, to find the globally optimal UAV relay position in a congested or broken route.

Paper Organization: The rest of the paper is organized as follows: The related work is summarized in Section II. The swarm network model is described in Section III, folowed by the MHQ-PRP queueing model in Section IV. In Section V, the deduced queueing results and other swarm network parameters are used as the input to the DQN model for accurate UAV positioning. The performance analysis is discussed in Section VI, followed by the conclusions in Section VII.

II. Related Work

In this section, we briefly review the literature on UAV swarm network, followed by the multi-hop queueing models and relay positioning solutions.

A. UAV Swarm Network

Some studies on UAV swarming strategies have been performed in [13] [14] to address the node movement/trajectory control issues. As mentioned in Section I, the integration of swarming and networking schemes are critical. Since the goal of swarming is to mainly generate a particular formation, it can result in the "communication hole(s)" when the neighboring nodes have a long distance between them. A relay node must move there to build a wireless communication 'bridge'. The research on UAV communication aims to design the optimal routing and MAC protocols for a group of UAVs [15] [16].

B. M/G/1 Preemptive Repeat Priority (PRP) Queueing Model

An M/G/1 repeat priority queueing model was proposed in [17] for the multi-hop video transmission applications. The video packets were prioritized based on their QoS requirements in a multi-source, multi-receiver wireless network. In this paper, we extend it to multimedia data including voice, real-time video, pre-encoded video, and delay-tolerant data transmissions. A delay- and rate-based priority model for multi-hop network was proposed in [18]. But it did not consider the packet-level retransmissions to meet the QoS demands. This does not fit many practical networks which use TCP for reliable transmissions. A location-based multi-hop queueing model was proposed in [19] where packets are forwarded to the next hop based on the distance of the source node to the gateway. But the packet priorities were not considered. Our MHQ-PRP model considers the QoS differences among multimedia flows and deduces the queueing delay in each UAV.

C. UAV Positioning Problem

Optimization theory for UAV positioning was proposed in [6], [7], which considered both energy reduction and throughput maximization. But it only targets a single UAV case which acts a relay node between the source and destination. Thus it is not suitable for a swarm network with dynamic topology. The UAV was considered as a relay node and its throughput optimization formulation was proposed for both uplink and downlink rate adjustments in [20]. But it assumed a simple network scenario and focused on the placement of a single moving node. It did not consider the entire multi-hop route and the impact of dynamic network architecture.

A few other studies did consider the impact of routing scheme on relay node placement. For example, the UAV swarming with routing optimization was studied in [21], where a hueristic approach was used to improve the information control plane (ICP) performance for building the high-quality routes among the static nodes. It also used physical control plane to guide the nodes to reconfigure their positions, according to the instructions from ICP. But this scheme can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

easily get stuck in local maxima since it does not have a reliable method to estimate what will occur in a route in the future communication stages. In our work, we use a DQN-based positioning algorithm to find the globally optimized relay positions.

A time-varying formation tracking protocol with Riccati equation was proposed in [22] which uses the neighboring UAV's profile information (such as traffic types) to guide a follower UAV to follow a leader UAV. Although this work considered dynamic topology, it did not not differentiate the traffic priorities and cannot guarantee the global optimization of relay node placement.

The path planning of mobile robots was proposed in [23] by predicting the wireless links through a supervised learning of the sensed data collected from the neighboring robots. However, our work aims to determine the suitable location(s) of intermediate node(s) given the source and destination. It is not a path planning problem although we also need to determine location of the next forwarding node.

In a nut shell, although some studies on relay placement in wireless networks have been conducted, our unique contributions here include the use of a deep-learning-based global optimization algorithm to determine the best locations for one or more relay UAV nodes, as well as the adoption of prioritized multi-hop queueing model to determine the best route for each type of traffic, in a highly dynamic MUM-based swarm network.

III. System Model

As mentioned in Section I, we assume a MUM network with two layers of nodes. Three types of nodes are present in the UAV swarm network:

(1) *Swarm nodes*: Most UAVs (> 90%) belong to this type of nodes, which form different swarm topologies based on the application. Fig. 2 shows a typical swarm formation where N nodes are arranged in a spiral pattern (or circular pattern) with radius r of S spirals. The center of spiral is the leader UAV. Such a spiral formation has good surveillance coverage (its shape naturally covers a specific region). It also has good communication performance because nodes in the outer circles can use gradient routing to reach an inner circle node. The distance between a node n_i and any of its immediate neighbors, such as n_j, have the minimum separation $D_{ij} \geq D_{min}$, where D_{min} is the minimum separation that the two nodes should maintain. When two nodes are out of each other's transmission range, a relay node is needed between them.

Note that the spiral pattern in Fig. 2 is used only as an example of UAV swarm pattern. In fact, our scheme can be applied to any type of UAV swarming pattern since our proposed DQN-based node placement algorithm is independent of UAV swarm formation.

(2) *Gateway nodes*: a small number of nodes (< 5%) are chosen as gateways which can directly communicate with the higher-layer control (i.e., aircraft) nodes. They typically have a longer communication range than other UAVs. They run data aggregation algorithms to fuse the received data from different UAVs and forward the packets to the closest aircraft. They

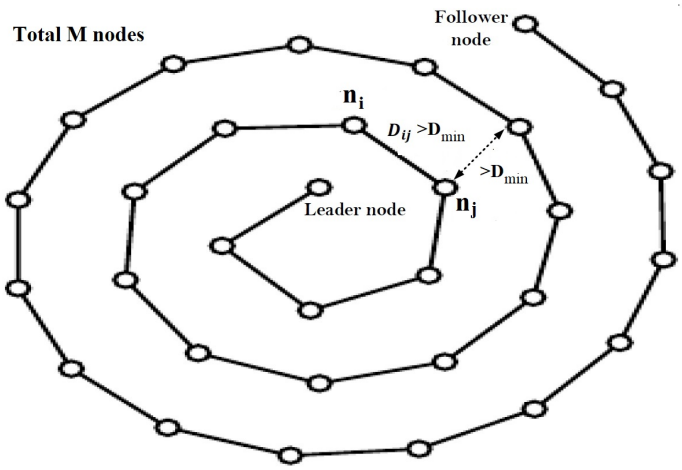

Fig. 2: UAV swarming pattern.

can also perform dual-band communications. For example, one band can be used for UAV network, and the other band can be used to forward the data to the aircraft. Swarm UAV nodes need to find a multi-hop route to reach the gateway node.

(3) *Relay nodes*: A small number of nodes (< 5%) among the swarm nodes do not participate in the swarm formation. Instead, they are used for compensating the 'communication holes'. In other words, they move to the broken wireless links to serve as the communication relays. This paper studies the movement control of these relay nodes.

In our two-layer network architecture shown in Fig. 1, the upper layer has a few aircraft nodes which are more powerful than the lower layer UAV nodes. These control nodes serve multiple purposes: (1) Execute the DQN algorithm because such an algorithm needs higher CPU capacity; (2) Control the global routing topology and send commands to ask the relay nodes to move to the suitable locations based on the DQN algorithm; (3) Collect all nodes' communication status information from the lower layer UAV nodes.

In addition, we assume that the location of each swarming node is known via the GPS services. Since the network has some more powerful gateway UAVs equipped with better antennas and RF transceivers, they can provide accurate position information to the control node. If some UAVs can not provide their accurate location information due to poor weather conditions or high mobility, an existing wireless positioning mechanism can be used, such as the GPS-free positioning [24], time-of-arrival (ToA) or direction-of-arrival (DoA) based positioning [25]. Since gateways nodes have reliable location information, they can also be used to deduce the location of other UAVs by using an existing scheme, such as the landmark-based positioning scheme [26].

We assume the network is assigned a transmission bandwidth of B Hz. The Doppler effect due to the mobility of UAVs is assumed to be properly compensated. Every swarming node moves with an approximately constant speed in 2D plane. We also assume that time interval T is divided into N_T discrete time slots with $T = N_T * \delta t$, where δt is the length of a time slot. The value of δt is sufficiently small and the UAV location is assumed to remain unchanged within each time slot.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

Parameters List	
Parameters	**Description**
H	Maximum number of hops
d_i	Delay deadline of priority i packet, $i \in \{1,2,3,4\}$
$\lambda_{i,s,h}$	Arrival rate of packets generated at the source node at h^{th} hop, $h \in \{1,2,3,..,H\}$. The packets have priority i
$\lambda_{i,r,h}$	Arrival rate of relayed priority i packets received by a node at h^{th} hop.
L_i	Length of priority i packet
R_i	Bit rate of priority i packet
$E[X_{i,h}]$	Packet service time of priority i packet at h^{th} hop due to channel quality
W_h	Channel access delay at a node of h^{th} hop
$W_{i,h}$	Queueing delay of priority i packet at a node of h^{th} hop
$\rho_{i,h,h+1}$	Packet error rate (PER) for priority i packets due to the link quality of h^{th} and $(h+1)^{th}$ hops
$\Psi_{i,h}$	Packet dropping rate (PDR) for priority i packets seen at $(h)^{th}$ hop due to channel access delay (W_h) and channel quality
$C_{h,h+1}$	Channel capacity of the nodes between h^{th} and $(h+1)^{th}$ hops
$\delta_{i,h,h+1}$	Maximum number of retransmissions for priority i packets between h^{th} and $(h+1)^{th}$ hops

TABLE I: Parameters List

We assume the link between the two nodes located at (x_i, y_i) and (x_j, y_j) has the Rician fading. The associated SINR level observed between these two nodes at time slot n can be defined as

$$\gamma_{ij}[n] = \frac{P_i[n]G_{ij}[n]}{\sum_{k=1,k\neq i}^{N} P_k[n]G_{ik}[n] + \sigma_j^2}, \quad n = 1,2,....,N_T \qquad (1)$$

Here P_i is the transmit power of node i, P_k is the transmit power of node k causing interference to node j, σ_j^2 is the Gaussian noise variance, and $G_{ij}[n]$ is the associated channel response. Thus, the achievable average data rate per time slot observed at node i is given by,

$$R_i = \frac{1}{N_T}\sum_{n=1}^{N_T} R_i[n] = \frac{1}{N_T}\sum_{n=1}^{N_T} B.log_2\left(1+\gamma_{ij}[n]\right) \qquad (2)$$

IV. Proposed MHQ-PRP Queueing model for UAV path

We consider four priority classes of data packets: 1. Real-time voice, 2. Real-time video, 3. Non-real-time video, and 4. General delay-tolerant data. We denote d_i, $i \in \{1,2,..,4\}$ as the delay deadline of each packet priority, where $d_1 < d_2 < d_3 < d_4$. The used parameters are listed in Table 1.

To make our queueing model more general, we do not tie it to a specific MAC layer protocol. Each node in the network maintains two queues, one for its own data with arrival rate $\lambda_{s,h}$, and another for relay packets to the gateway node with the arrival rate $\lambda_{r,h}$. The gateway node then delivers all the data packets to the control node in higher layer.

To meet the delay constraints of each packet over the multi-hop path, we design a queueing model for a multi-hop path based on the packet priority level. The structure of queueing model is shown in Fig. 3. For any incoming or generated packet at the source node, the information such as packet priority, multi-hop information, time-to-live (TTL), and SINR, are captured at each hop. If the node is a follower node (tail end), it directly forwards the packet to the next hop. If the node is a relay node, it determines the service time, TTL and arrival rate of each packet.

To analyze the queueing delay in multi-hop environment, the parameters from different layers are taken into consideration as discussed below:

1) Application Layer: In this layer, the packet priority level is assigned based on application's QoS requirements. For example, the real-time voice data and commands sent from the aircraft are assigned the highest priority.

2) Network Layer: Although a shortest path routing protocol like AODV [27] is used, we also attempt to build a much shorter path by adding one or more relay nodes on poor quality links between the source and gateway nodes. When selecting a relay node, the following parameters are considered: (i) long channel idle time (i.e., the channel available in the new link is not used in neighboring links and thus has the long idle time), (ii) minimum contention from the neighboring nodes during channel access, and (iii) minimum transmission power required to reach the relay node.

3) MAC Layer: Let $p(h)$ be the probability of successful channel access for the h hop node. The determination of $p(h)$ should consider the RF interference from other links, channel contention when multiple neighbors compete for the channel, the number of available RF channels (if multiple RF channels are available) and the MAC protocol (such as CSMA-CA or a schedule-based time division multiple access (TDMA)).

Generally, the node close to a gateway needs to be given more channel access opportunities, due to its large amount of relayed traffic. Besides, we use $\delta_{i,h,h+1}$ to denote the maximum number of retransmissions allowed for the priority i packets between node h and $h+1$, which depends on the delay deadline of packets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

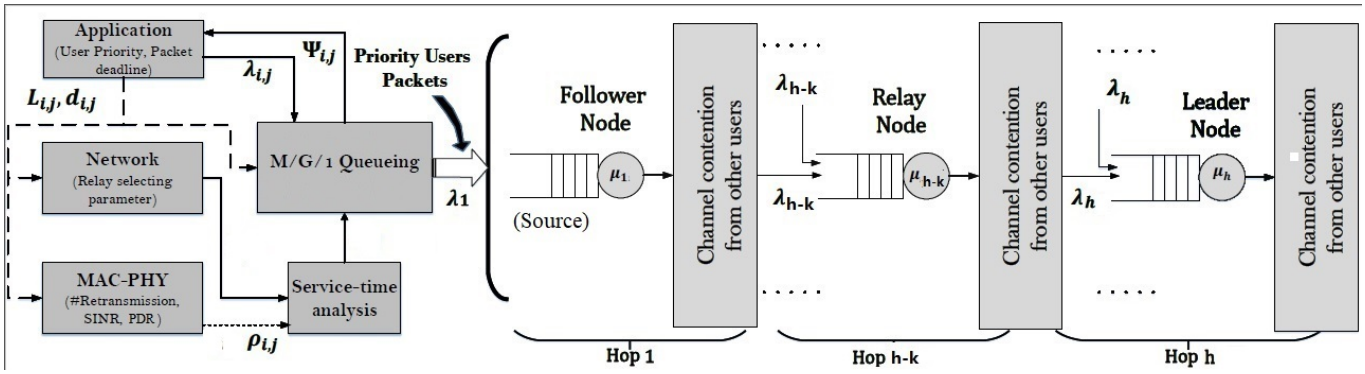

Fig. 3: Proposed queueing model for multi-hop case.

4) Physical Layer: Over H-hops, the end-to-end PDR for priority i packets is given by

$$\Psi_i = 1-(1-\Psi_{i,0})\left(\prod_{h=1}^{H} 1-\Psi_{i,h}\right) \tag{3}$$

Here $\Psi_{i,h}$ is the PDR incurred due to delay deadline expiration in hop h. Parameter $\Psi_{i,0}$ is the initial PDR observed at the source node. We provide main queueing delay results below after adapting some parameters from [11] to the proposed MHQ-PRP model.

A. Packet Arrival Rate

Assume the packets with priority i arrive at hop h with PDR $\Psi_{i,h}$. The expected packet arrival rate is given by

$$(1-\Psi_{i,h})\lambda_{i,r,h} = \prod_{\hat{h}=1}^{h}(1-\Psi_{i,\hat{h}})\lambda_{i,s,h} \tag{4}$$

B. Packet Service Time

Assuming the geometric distribution of service time, the first moment of service time of priority i packets at hop h with transmission rate $R_{i,h,h+1}$ *packets/sec* due to channel quality can be expressed as

$$E[X_{i,h}] = \frac{L_i(1-\rho_{i,h,h+1}^{\gamma_{i,h,h+1}})}{R_{i,h,h+1}(1-\rho_{i,h,h+1})} \tag{5}$$

After $\gamma_{i,h,h+1}$ retransmission attempts at the MAC layer, we can approximate $(1-\rho_{i,h,h+1}^{\gamma_{i,h,h+1}}) \approx 1$ [17]. Then we have

$$E[X_{i,h}] = \frac{L_i}{R_{i,h,h+1}(1-\rho_{i,h,h+1})} \tag{6}$$

Hence, the average service time of priority i packet at hop h with average channel access delay $E[W_h]$ is given by

$$E[S_{i,h}] = E[W_h] + E[X_{i,h}] \tag{7}$$

C. Average Queueing Delay and Packet Dropping Rate

Let $E[W_{i,h}]$ be the average queueing delay of priority i packet at hop h. Based on the priority queueing analysis for the preemptive priority M/G/1 queueing model [28], we get

$$E[W_{i,h}] = \frac{\sum_{i=1}^{4}\lambda_{i,h}E[S_{i,h}^2]}{2\left(1-\sum_{i=1}^{4-1}\lambda_{i,h}E[S_{i,h}^2]\right)\left(1-\sum_{i=1}^{4}\lambda_{i,h}E[S_{i,h}^2]\right)} \tag{8}$$

where $E[S_{i,h}]$ is the expected service time and $E[S_{i,h}^2]$ is its second moment. Hence, the average end-to-end PDR at hop h for the packets sent from source node can be determined as shown in (9).

V. DQN-BASED OPTIMAL POSITIONING FOR RELAY NODES

In this section, we discuss our relay node positioning model based on the network condition, such as link quality. We consider UAV positioning problem as deep reinforcement learning (DQN)-based optimization.

We assume N UAV nodes in the network which form M links including the links between gateway nodes and control nodes. The first stage of relay node positioning relates to selecting a suitable link for node placement. The second stage of positioning is local optimization for determining accurate location of relay node in the selected link.

The network parameters we consider for each link h include: (1) SINR (can be deduced from path loss model), (2) PDR (determined from the previous section), (3) Interference from external source, which is a binary value (0 or 1) to indicate the presence of external signals (i.e., not generated from the network itself).

A. Deep Q-Network

Using the information collected at each link, we train our DQN model to select an optimal link for the relay UAV placement. After link selection, we run an optimization algorithm to locally optimize the relay node location. Our DQN model is represented as a tuple $\{s,a,\mathbb{R}\}$ as discussed below.

1) State, s: Q-learning is used to derive the best long-term policy to determine the optimal UAV positioning pattern under different conditions, such as low channel quality and external interference signals. The action (in terms of determining the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

$$\Psi_{i,h} = Prob\left(W_{i,h} > d_i - \sum_{j=0}^{h-1} E[W_{i,j}]\right) = \left(\sum_{i=1}^{4} \lambda_{i,h} E[S_{i,h}]\right) exp\left(-\frac{\left(d_i - \sum_{j=1}^{h} E[W_{i,j}]\right)\left(\sum_{i=1}^{4} \lambda_{i,h} E[S_{i,h}]\right)}{E[W_{i,h}]}\right) \quad (9)$$

UAV location) is selected based on the current system *state* s^n at time slot n. Specifically, the system state s^n consists of the information regarding link SINR (γ_{ij}), PDR ($\Psi_{i,h}$), and external interference condition (J_h), for the time slot $[n-1]$ at link h, i.e., $s^n = \{\gamma_{hj}, \Psi_{i,h}, J_h\}^{(n-1)}$.

2) Action, a: The actions are used to change the UAV behavior in response to the states seen at time slot n. They are executed sequentially. To optimally deploy the UAVs, we define our action set as the probability of selecting a link l_h, where $h \in H$. Optimal action selection maximizes the average reward of the network.

3) Optimization: After a particular link is chosen, we use the optimization process to place the UAV at the optimal location within the grid A_g, $g \in N_g$, where N_g is the total number of grid locations. The optimization problem for optimal positioning is depicted as:

$$\begin{aligned} & max \ \frac{1}{T} \int_{n=0}^{T} \mathbb{R}^n dn \\ s.t: \ & i. \quad d_{t,k} \geq d_{min}; \ k \in \{1,2,...,N_{neigh}\} \\ & ii. \quad E[W_j] \leq D_j; \ j \in \{1,2,...,4\} \\ & iii. \quad J_g = 0; \ g \in N_g \end{aligned} \quad (10)$$

Here, T is the entire optimization time duration, d_{min} is the required minimum inter-UAV distance. The objective function defines the achievable reward $\mathbb{R}$ at a specific position at time slot n. The first constraint defines the minimum distance parameter the UAV should maintain with each of its neighboring UAVs to avoid physical body collisions. The second constraint specifies that the average queuing delay at grid A_g should be less than the delay threshold D_j. The third constraint defines the presence of external interference signal to the network, where a binary decision variable 1 (or 0) represents that ESS is present (or absent).

4) Reward, $\mathbb{R}$: The reward defines the effect of relay UAV positioning scheme in the current state s for the adopted action a in time slot n. The DQN scheme optimizes the position of UAV when it maximizes the SINR level. In our model we use the total SINR of the entire swarm network as a parameter to quantify the optimal positioning of UAV since the existence of a route can have impact on other paths. Using (1), the reward term is defined as follows,

$$\mathbb{R} = \gamma_{ij}[n] = \frac{P_i[n] G_{ij}[n]}{P_{ESS} J_g + \sum_{k=1,k\neq i}^{M} P_k[n] G_{ik}[n] + \sigma_j^2}, \quad n = 1,2,....,N_T \quad (11)$$

The updated and locally optimized location with Reward $\mathbb{R}$ is chosen, and the location information with reward value is stored in the *memory replay* in deep learning model.

There are mainly two components in our DQN model: (1) convolutional neural network (CNN) [5], and (2) Q-learning based decision model. As depicted in Fig. 4, the system uses CNN to enhance the learning rate of Q-learning module as the UAV swarm formation and network communication capacity change over time. Similar to Q-learning, DQN updates Q-function for each state-action pair, which is the expected discounted long-term reward for state s and action a at time slot n. The Q-function is given by [5]:

$$Q(s,a) = E_{s'}\left[\mathbb{R}_s + \gamma \max_{a'} Q(s',a) | s,a\right] \quad (12)$$

where $\mathbb{R}_s$ is the reward received at the state s for action a which results in the next state s' with a discount factor γ, defining the uncertainty of the UAV nodes about the future reward. The discount factor reflects the less impact on the throughput performance from the older actions.

In fact, the Q-function can be approximated by using CNN with tunable weight parameters. It is a non-linear approximator for each action. However, due to network dynamics, the CNN model needs to be retrained to adapt to the UAV swarming process. Hence, a *replay memory* is used with the collection of past experienced state-action pairs and their respective rewards.

The CNN consists of two convolutional layers and two fully connected (FC) layers. The first convolutional layer consists of *20* filters, each with the size of 3×3 and stride *1*, and the second convolutional layer consists of *40* filters with the size of 2×2 and no change to stride value. Rectified linear units (ReLU) is used as an activation function in each layer including FC layers. The first FC layer consists of 180 ReLU units and the second FC has $M+1$ ReLU units, where M is the total number of links. At time slot n, the weight of the filter in each layer is denoted by θ^n.

Furthermore, at time slot n, the observed state sequence for $\mathcal{B}$ system state-action pairs is denoted as $\varphi^n = \{s^{n-\mathcal{B}}, a^{n-\mathcal{B}}, ..., a^{n-1}, s^n\}$. Input to the CCN is from the replay buffer by reshaping the state sequence into 6×6 matrix to estimate the $Q(\varphi^n, a|\theta^n)$. The state sequence in replay buffer is chosen randomly from the *experience memory pool*, $\mathbb{D} = \{e^1, ..., e^n\}$, where $e^n = (\varphi^n, a^n, \mathbb{R}_s^n, \varphi^{n+1})$. Basically, experience replay chooses an experience e^d randomly, with $1 \leq d < n$ to update the weight parameter θ^n according to the stochastic gradient descent (SGD) method [5]. Updating θ^n results in the minimum mean-squared error of the targeted optimal Q-function with the minibatch updates, and the following loss

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

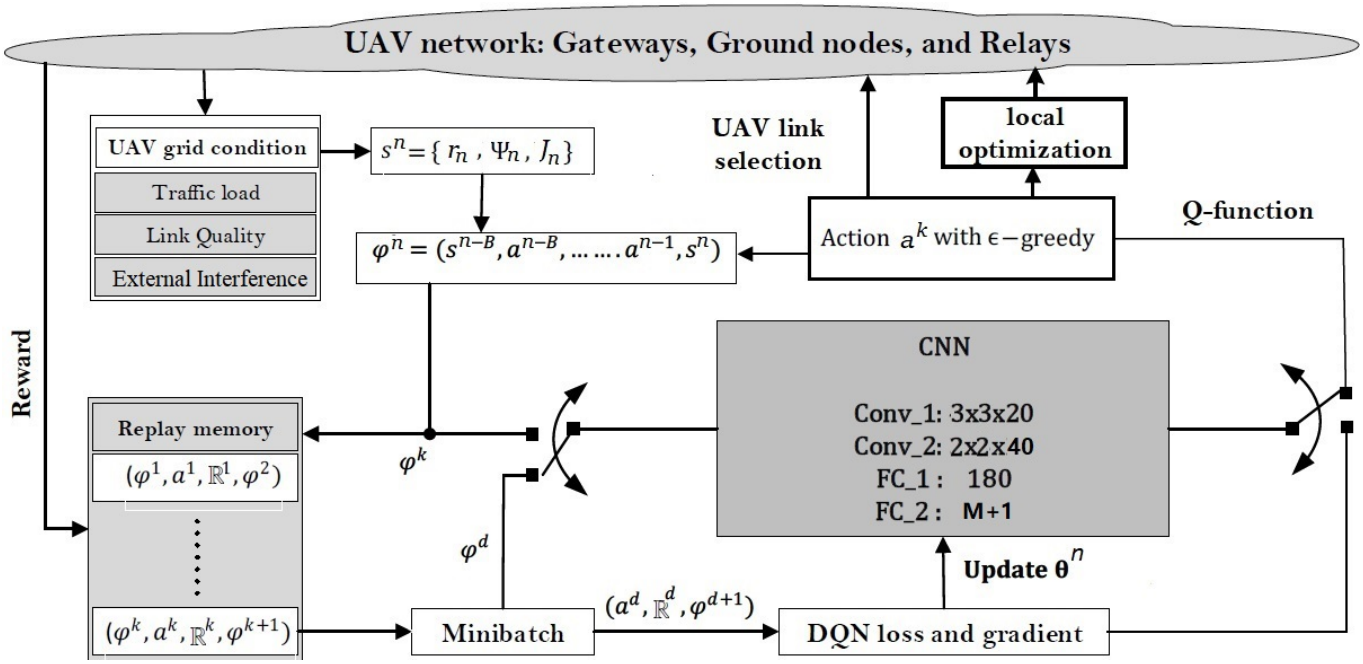

Fig. 4: DQN-based UAV deployment.

function can be denoted as [5]:

$$\mathbb{L}(\theta^n) = \mathbb{E}_{\varphi^n, a, \mathbb{R}_s, \varphi^{n+1}} \left[\left(Q_{Target} - Q(\varphi^n, a : \theta^{n+1}) \right)^2 \right] \quad (13)$$

where Q_{Target} is the target optimal Q-fucntion, which is given by

$$Q_{Target} = \mathbb{R}_s + \gamma \max_{a'} Q(\varphi^{n+1}, a' : \theta^{n-1}) \quad (14)$$

The weights θ^n are updated by using the gradient of loss function $\mathbb{L}$ w.r.t to the weights θ^n. The loss gradient $\nabla_{\theta^n} \mathbb{L}(\theta^n)$ can be expressed as

$$\nabla_{\theta^n} \mathbb{L}(\theta^n) = \mathbb{E}_{\varphi^n, a, \mathbb{R}_s, \varphi^{n+1}} \left[Q_{Target} \nabla_{\theta^n} Q(\varphi^n, a : \theta^n) \right] - \mathbb{E}_{\varphi^n, a, \mathbb{R}_s, \varphi^{n+1}} \left[Q(\varphi^n, a : \theta^n) \nabla_{\theta^n} Q(\varphi^n, a : \theta^n) \right] \quad (15)$$

The weight parameter θ^n is updated for every time slot. The update repeats by randomly selecting the experiences from the experience pool. Finally, with the updated Q-function, the action a^n is chosen for the state s^n according to the ϵ-greedy algorithm. The optimal action is chosen from the set of Q-functions with the probability of $(1-\epsilon)$, and we have:

$$a^* = arg \max_{a'} Q(\varphi^n, a') \quad (16)$$

When a link is selected for UAV placement, the UAV observes SINR as a reward information from the swarming network. Based on the next state, the UAV stores the new experience $\{\varphi^n, a^n, \mathbb{R}_s^n, \varphi^{n+1}\}$ in the replay memory as shown in Fig. 4.

As shown in Fig. 4, in DQN-based UAV positioning scheme, the UAV network's features in terms of link quality (SINR), traffic-load (PDR), ESS condition of each link, are created at each timestamp. The current state, the next state, the action, and the reward incurred due to such an action, are all stored as the feature vector for CNN in replay memory, which stores D past experiences. In each iteration, B out of D batches are selected to train the CNN module to fine-tune the decision-making result to fit the environment conditions. Once the DQN makes the position selection, the UAV performs the fine-tuning of the position where it is placed, and the whole process repeats again.

The entire relay node positioning algorithm is shown in *Algorithm 1* and its convergence to the optimal point is proved in Appendix. It takes the SINR, PDR and ESS conditions as the input parameters, and finds the best position for a relay UAV. It uses an iterative process to perform training and inference for DQN with replay memory. It converges when the maximum reward is achieved.

VI. Performance Analysis

A. Average Multihop Queueing Delay

In this section, we analyze the performance of our proposed M/G/1 PRP multi-hop queueing model for a network structure shown in Fig. 5. Each hop has its own source data and also helps to relay the data forwarded from the previous hop with a link capacity of *5 Mbps*. We consider the following four types of data transmissions and the packet length L_k is *1000* bytes:

1) Voice data with a bitrate of *50 Kbps* and latency constraint of *50 ms*.

2) Skype-like real-time video with a bitrate of *500 Kbps* and latency constraint of *100 ms*.

3) Pre-encoded HD video with a bitrate of *3 Mbps* and playback delay deadline of *1 sec*.

4) Data with *5 sec* delay constraint (such as file download at *2 Mbps*).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

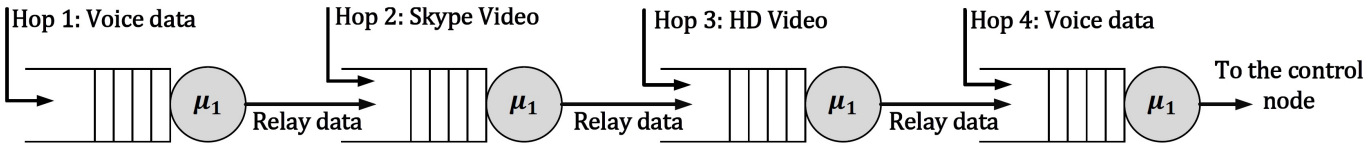

Fig. 5: Elementary network structure used for simulations.

Algorithm 1 : DQN based relay UAV positioning

Initialize, $A_g, \theta, \gamma, SINR^0, J^0$
Initialize state-action pairs, $\mathcal{B}$
Initialize batch size, B
Initialize replay memory size, D $\leftarrow \emptyset$
Estimate $E[W]$ from (8)
Input: $s^0 = [SINR^0, PDR^0, J^0]$
Output: DQN:- Optimal UAV location with maximum $\mathbb{R}$.

for n **do**=1,2,...
  if n$\leq \mathcal{B}$ **then**:
    Choose link randomly $a^n \in \{1,2,...N\}$
  else
    Obtain CNN output $Q(\varphi^n, a|\theta^n)$ with input φ^n and weights θ^n
    Choose a^n via ϵ-greedy algorithm
  end if
  Perform optimization locally for grid A_{a^n}
  Observe $SINR^n, PDR^n, J^n$
  Estimate the Reward $\mathbb{R}$ and obtain $s^{n+1} = [SINR^{n+1}, PDR^{n+1}, J^{n+1}]$
  Create state, action, reward vector: $\varphi^{n+1} = \{s^{n-\mathcal{B}+1}, a^{n-\mathcal{B}+1}, ..., a^n, s^{n+1}\}$
  Add the new experience to $\{\varphi^n, a^n, \mathbb{R}^n, \varphi^{n+1}\}$ to memory D
  for d **do**=1,2,...,B
    Select randomly $(\varphi^d, a^d, \mathbb{R}^d, \varphi^{d+1})$ from D
    Train CNN n' iterations
    Calculate Q_{Target} using (14) in Section V-A4
  end for
  Update weight parameter θ^n using (15)
end for

We analyze the performance of our proposed multihop queueing model in terms of the average queueing delay and compare with a FIFO queueing model [19], where the packet priority based on the latency requirement is not considered. For the sake of simplicity, we assume that the average waiting delay due to channel access operations for each node is $E[W] = 0$. This is a reasonable assumption when enough bandwidth is available in the network, especially when the TDMA-based MAC scheme is used (it does not cause any channel access conflict).

The expected queueing delay for priority i packets, $E[W_i]$ is shown in Fig. 6. Since the hop-1 source data has the voice data with a low latency requirement, it is given the highest priority across all hops. Hence it experienced almost zero queueing delay at all hops. The Skype-like real-time video generated at 2nd hop is given the second highest priority since

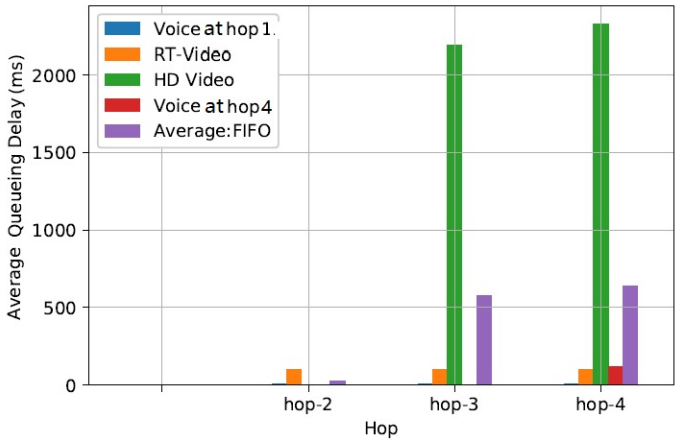

Fig. 6: Analytical value of average queueing delay at each hop for the source as well as relay data, compared with FIFO queuing method.

its latency requirement is less strict than the voice. However, the voice data generated at 4th hop is given a lower priority than the real-time video. Therefore the voice and real-time video experience comparable queueing delays at hop 4. In addition, hop-3 source data is a HD video with a higher play-back time, and is given a lower priority. Therefore, it experiences a much higher queueing delay than other data. The proposed queueing model performs better than the FIFO model [19] for the high-priority data, which demonstrates its capability of differentiating the traffic priorities.

Fig. 7 shows the average queueing delay experienced by the source data of each hop. As stated earlier, HD video at hop-3 experiences the highest delay, whereas the voice data at hop-1 experiences the least delay.

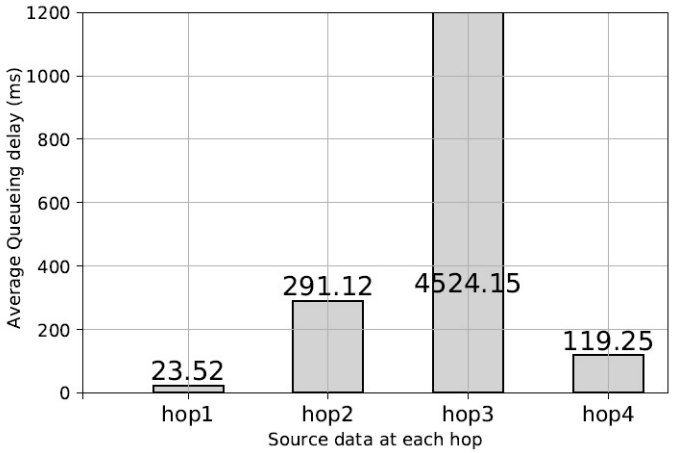

Fig. 7: Average queueing delay for the source data for the network structure shown in Fig. 7.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

B. DQN-based UAV Positioning Scheme

In this section, we analyze the performance of our DQN-based UAV positioning scheme in the ESS and ESS-free environments. We considered the real-time video data with a bitrate of 500 Kbps and 100ms latency. Input to the CNN is the 6x6x3 tensor with 36 past experiences of each of the three parameters (SINR, PDR, and ESS (1/0)). Here the past experiences are converted to a 6x6 matrix for each parameter. The true label for each input is the Reward seen in the past experience. As mentioned before, considering UAV positioning as a pure optimization task through gradient methods will not suffice since we may have multiple local optima. We use a swarming network of 150 UAV nodes. At each time stamp, the network parameters for each link can be obtained at the higher-layer node through the gateway nodes and the DQN model is trained using those network parameters. Over several iterations, the higher-layer node learns the optimal decision model and updates the *replay memory* with the current network observations. Once the relay UAV is placed at a particular link, local optimization is performed by the UAV to determine its location on the link. To demonstrate the optimization, we use an elementary UAV positioning structure in Fig. 8, in which 4 UAVs are present at a certain distance on the four ends of the 100x100 square meters area. The relay UAV should be placed in the cross-section area such that the throughput is maximized under dynamic network conditions.

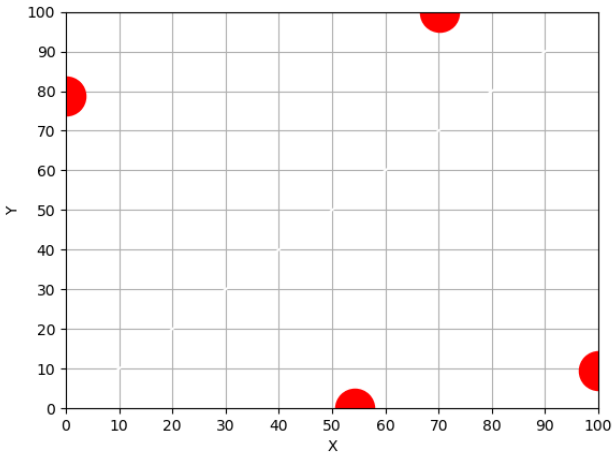

Fig. 8: The UAV network model with four nodes.

1) UAV network without ESS: We consider the UAV swarming network area without the presence of ESS. Due to multi-path and path-loss effect, broken RF links are present in the area. The performance of the proposed DQN-based UAV positioning scheme is compared with Q-learning based scheme. As shown in Fig. 9, the DQN-based optimization algorithm significantly outperforms the Q-learning algorithm since it can achieve a normalized throughput that approaches 1.

2) UAV network with ESS: Here we consider the presence of ESS so that the UAV positioning area in Fig. 8 is affected by external interference. Thus some of the UAVs in this

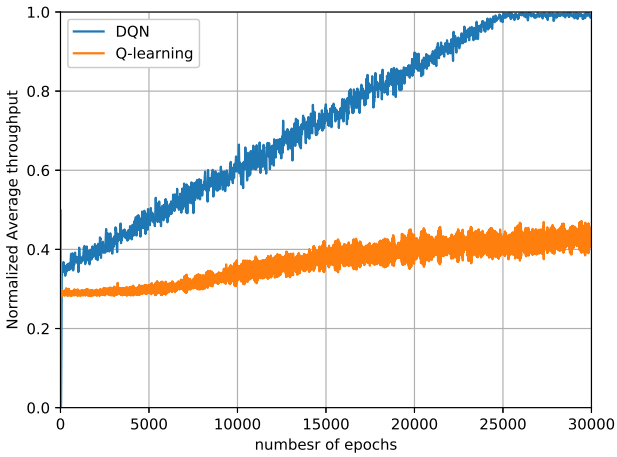

Fig. 9: Performance comparison of UAV positioning for DQN and Q-learning algorithm without ESS.

area cannot communicate with each other. For this ESS case, the performance of our DQN-based optimization algorithm is compared with the Q-learning scheme. As shown in Fig. 10, our DQN algorithm significantly outperforms the Q-learning algorithm and is able to achieve the maximum normalized throughput close to 0.65. As expected, this performance is lower than the ESS-free case. This performance would further decreases if the area affected by external interference is increased.

Note that our DQN algorithm achieves the optimal solution in less iterations when ESS is present as compared to the ESS-free network. Since the reward (i.e., normalized throughput) maps to zero for the regions affected by ESS, the action space becomes small and helps the algorithm to achieve optimal condition faster.

An important advantage of DQN over Q-learning (or other traditional reinforcement-learning-based algorithms) is that it can be trained for many different network conditions, including ESS. Training it with CNN models creates a knowledge base for the learning agent to choose the optimal action that fits the current network conditions. We also observed that the optimal throughput achieved by the DQN model linearly increases with the number of epochs, which is not the case with the Q-learning model. A reason behind this could be that we each parameter of UAV network model is used as the input to the CNN model for finding the optimized Q values, which generates a better pattern separation among the input parameters.

Next, we compare the performance of our proposed DQN model by considering (a) only PDR, (b) only SINR, and (c) PDR+SINR+ESS case with ESS=0 in Fig. 11. When PDR over the link is considered as the only Reward value in DQN learning model, the node finds the best link with low PDR. To implement this in DQN model, we considered 36 previously seen PDR values in that link as one state. Based on these previously seen values, the DQN makes decision on selecting the link with the lowest PDR to place the node.

When SINR over the link is considered as the only Reward

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2019.2907520, IEEE Transactions on Cognitive Communications and Networking

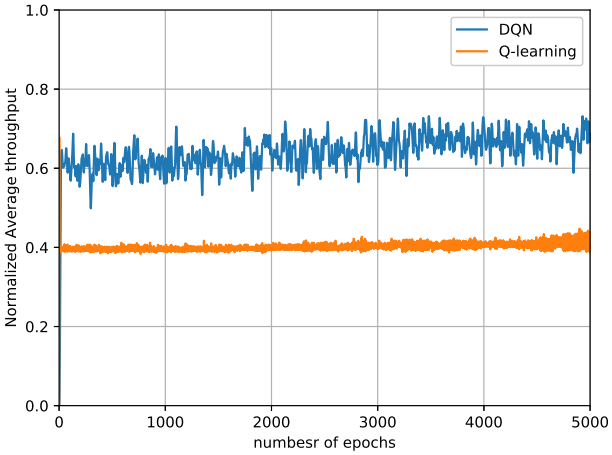

Fig. 10: Performance comparison of UAV positioning for DQN and Q-learning algorithm in the presence of ESS.

in DQN learning model, the node finds the best link which has high SINR. The results in Fig. 11 show that the maximum normalized throughput is achieved only when all the parameters, i.e. PDR, SINR and ESS (we have considered ESS=0 case) are considered. However, the DQN model based on individual parameter (PDR or SINR) outperformed the simple Q-learning model. For an ideal condition (with PDR = 0, very high SINR and ESS = 0), the normalized network throughput of 1 was achieved in the very first iteration.

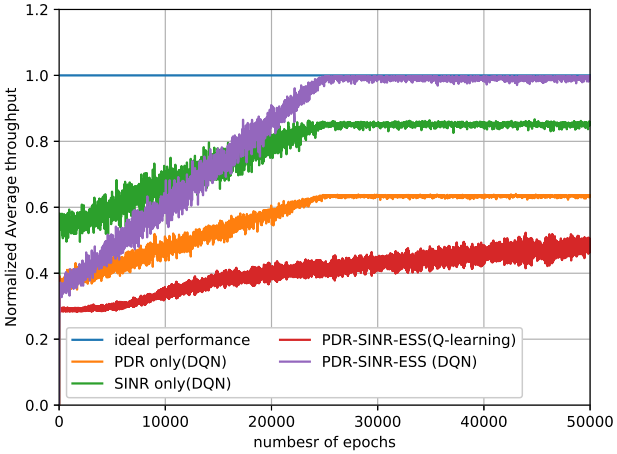

Fig. 11: UAV positioning performance comparison in different scenarios.

Finally, we compare the performance of our DQN model for real-time video at 500kbps with the delay deadline of 100ms at different hops in Fig. 12. We analyze the performance of DQN over 1 to 3 hops for network configuration shown in Fig. 5. For the first hop, the achieved throughput is close to the maximum since it experienced no waiting delay except some varying channel conditions. For the same hop (hop-1) the Q-learning algorithm does not achieve such a high throughput. The throughput value decreases in the subsequent hops due to two reasons: (i) Increase in congestion level at each hop, and (ii) Increase in the number of higher priority packets. In each scenario, our DQN algorithm has considered the dynamic channel conditions with temporal variations (such as the multi-hop waiting delay) and spatial variations (such as SINR variations).

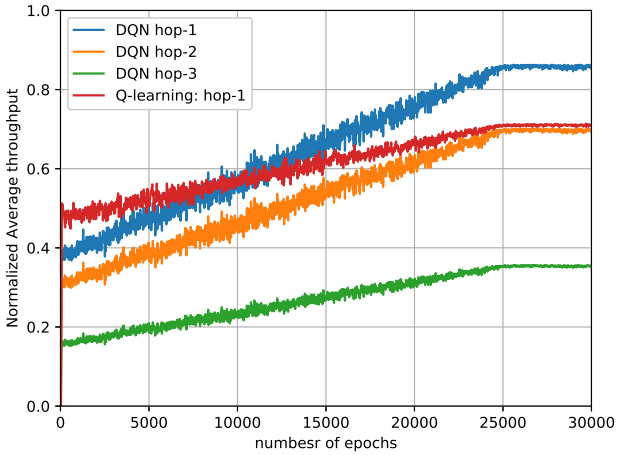

Fig. 12: Average normalized throughput comparison between DQN model and Q-learning in ESS-free UAV deployment.

VII. Conclusions

A novel UAV positioning scheme was presented to achieve the optimal communication among swarming nodes. Unlike conventional single-UAV relay placement schemes, our solution is suitable to the networked, dynamic UAV swarming applications by using the replay-buffer-based DQN learning algorithm which can keep track of the network topology changes. Our DQN scheme used the major parameters in different protocol layers that reflect the swarm network conditions, including the physical layer SINR, routing layer PDR, and application layer QoS levels. The MHQ-PNP queueing model was used to calculate the multi-hop latency and PDR level. We implemented the DQN-based swarm simulation platform and generated results, which validated the global throughput optimization by using our proposed UAV relay positioning scheme.

In the future, we will extend the above MUM swarm network to a distributed UAV swarm network without centralized control. We will also use a few concrete swarm formations to study the inter-swarm communication optimization issues when multiple swarm groups are present.

References

[1] B. T. Clough, "UAV swarming? so what are those swarms, what are the implications, and how do we handle them?" Air Force Research Lab, Wright-Patterson AFB, OH Air Vehicles Directory, Tech. Rep., 2002.

[2] Z. Han, A. L. Swindlehurst, and K. R. Liu, "Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *IEEE Trans. Vehicular Technology*, vol. 58, no. 7, pp. 3533–3546, 2009.

[3] R. Iyengar and B. Sikdar, "A queueing model for polled service in WiMAX/IEEE 802.16 Networks." *IEEE Trans. Commun.*, vol. 60, no. 7, pp. 1777–1781, 2012.

[4] B. Sikdar, "An analytic model for the Delay in IEEE 802.11 PCF MAC-based wireless networks." *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1542–1550, 2007.

[5] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[6] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, 2017.

[7] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for mobile relaying systems," in *Globecom Workshops (GC Wkshps), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[8] Y. Zeng and R. Zhang, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, 2016.

[9] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[10] A. Koushik, F. Hu, and S. Kumar, "Intelligent spectrum management based on transfer actor-critic learning for rateless transmissions in cognitive radio networks," *IEEE Trans. Mobile Computing*, vol. 17, no. 5, pp. 1204–1215, 2018.

[11] Y. Wu, F. Hu, Y. Zhu, and S. Kumar, "Optimal spectrum handoff control for CRN based on hybrid priority queuing and multi-teacher apprentice learning," *IEEE Trans. Vehicular Technology*, vol. 66, no. 3, pp. 2630–2642, 2017.

[12] R. Liu and J. Zou, "The effects of memory replay in reinforcement learning," *arXiv preprint arXiv:1710.06574*, 2017.

[13] J. Hahn, C. Peterson, S. Noghanian, and P. Ranganathan, "Optimization of Swarms of UAVs," in *IEEE Int. Conf. Electro Information Technology*, May 2016, pp. 0793–0797.

[14] M. R. Brust and B. M. Strimbu, "A networked swarm model for UAV deployment in the assessment of forest environments," in *IEEE 10th Int. Conf. Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, April 2015, pp. 1–6.

[15] V. R. Khare, F. Z. Wang, S. Wu, Y. Deng, and C. Thompson, "Ad-hoc network of unmanned aerial vehicle swarms for search and destroy tasks," in *4th IEEE Int. Conf. Intelligent Systems*, Sept 2008, pp. 665–672.

[16] D. T. Ho, E. I. GrÃŷtli, S. Shimamoto, and T. A. Johansen, "Optimal Relay Path Selection and Cooperative Communication Protocol for a Swarm of UAVs," in *2012 IEEE Globecom Workshops*, Dec 2012, pp. 1585–1590.

[17] H.-P. Shiang and M. Van Der Schaar, "Multi-user video streaming over multi-hop wireless networks: a distributed, cross-layer approach based on priority queuing," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, 2007.

[18] S. Tamilarasan and P. Kumar, "Dynamic resource allocation using priority queue scheduling in multi-hop cognitive radio networks," in *Computational Intelligence and Computing Research (ICCIC), 2016 IEEE Int. Conf.* IEEE, 2016, pp. 1–5.

[19] T. Liu and W. Liao, "Location-dependent throughput and delay in wireless mesh networks," *IEEE Trans. Vehicular Technology*, vol. 57, no. 2, pp. 1188–1198, 2008.

[20] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications with unmanned aerial vehicles: Performance and optimization," *IEEE Trans. Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2068–2085, 2011.

[21] R. K. Williams, A. Gasparri, and B. Krishnamachari, "Route swarm: Wireless network optimization through mobility," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2014, pp. 3775–3781.

[22] X. Dong and G. Hu, "Time-varying formation tracking for linear multiagent systems with multiple leaders," *IEEE Trans Automatic Control*, vol. 62, no. 7, pp. 3658–3664, July 2017.

[23] E. F. Flushing, M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "Spatial prediction of wireless links and its application to the path control of mobile robots," in *9th IEEE Int. Sympo. Industrial Embedded Systems*. IEEE, 2014, pp. 218–227.

[24] S. Capkun, M. Hamdi, and J. P. Hubaux, "GPS-Free positioning in mobile ad-hoc networks," in *Proc. 34th Annual Hawaii Int. Conf. System Sciences*, Jan 2001.

[25] M. Navarro and M. Najar, "TOA and DOA estimation for positioning and tracking in IR-UWB," in *IEEE Int. Conf. Ultra-Wideband*, Sept 2007, pp. 574–579.

[26] J. Schmackers and A. Glasmachers, "Landmark-based fast positioning for sensor data fusion, receiver design and measurement results," in *14th IEEE Int. Conf. Intelligent Transportation Systems*, Oct 2011, pp. 25–30.

[27] I. D. Chakeres and E. M. Belding-Royer, "AODV Routing Protocol Implementation Design," in *24th Int. Conf. Distributed Computing Systems Workshops*, March 2004, pp. 698–703.

[28] R. W. Conway, W. L. Maxwell, and L. W. Miller, "Theory of scheduling," *Massachussets: Addison-Wesley*, 1967.

VIII. Appendix

A. Optimality of Reinforcement Learning

In a given RL environment with state set S, action set A, and reward r, the optimal value function in RL is the maximum achievable value represented as $Q^*(s,a)$, where $s \in S$ and $a \in A$. Using Bellman equation, the optimal values decompose into Bellman equation as [5]:

$$Q^*(s,a) = E_{s'}[r + \gamma \max_{a'}(Q^*(s',a')|s,a)] \tag{17}$$

We can treat $r + \gamma \max_{a'}(Q^*(s',a')$ as the target function for the DQN. Then using Q-values, $Q(s,a)$ stored in the Experience Replay we want to estimate MSE loss by stochastic gradient descent as

$$I = (r + \gamma \max_{a'}(Q^*(s',a') - Q(s,a,w))^2 \tag{18}$$

where w is the hyper-parameter (weight) associated with each target value in the DQN. Hence, optimization task is

$$\begin{aligned} min(r + \gamma \max_{a'}(Q^*(s',a') - Q(s,a,w))^2 \\ = min\frac{1}{2}(f_i(w)) \end{aligned} \tag{19}$$

where $f_i(w) = (r + \gamma \max_{a'}(Q^*(s',a') - Q(s,a,w))^2$.

With the above optimization formulation, the stochastic gradient method is implemented as follows,

$$w^{t+1} = w^t - \alpha \nabla(f_i(w^t)) \tag{20}$$

where α is the learning rate, $0 \le \alpha < 1$. In RL, the algorithm converges as $t \to \infty$ with

$$\sum_t \alpha = \infty \qquad \sum_t \alpha^2 < \infty \tag{21}$$

In addition, from the results of [5] and eqn (19),

$$E[||w^{t+1} - w^*||_2^2] \le (1 - \frac{\sigma_{min}(Q)^2}{||Q||_F^2}) E[||w^t - w^*||_2^2] \tag{22}$$

The above equation shows that, SGD converges exponentially fast and Q matrix does not need to be a full rank matrix.