August 2023
Doctoral Degree Thesis

# Routing Algorithms Based on Reinforcement Learning for Unmanned Aerial Vehicle Swarm Networks

Graduate School of Chosun University
Department of Computer Engineering
Muhammad Morshed Alam

# Routing Algorithms Based on Reinforcement Learning for Unmanned Aerial Vehicle Swarm Networks

무인 비행체 군집 네트워크를 위한 강화 학습 기반 라우팅 알고리즘

2023년 8월 25일

Graduate School of Chosun University

Department of Computer Engineering

Muhammad Morshed Alam

# Routing Algorithms Based on Reinforcement Learning for Unmanned Aerial Vehicle Swarm Networks

Advisor: Prof. Sangman Moh, Ph.D.

A thesis submitted in partial fulfillment of the requirements for a Doctoral degree

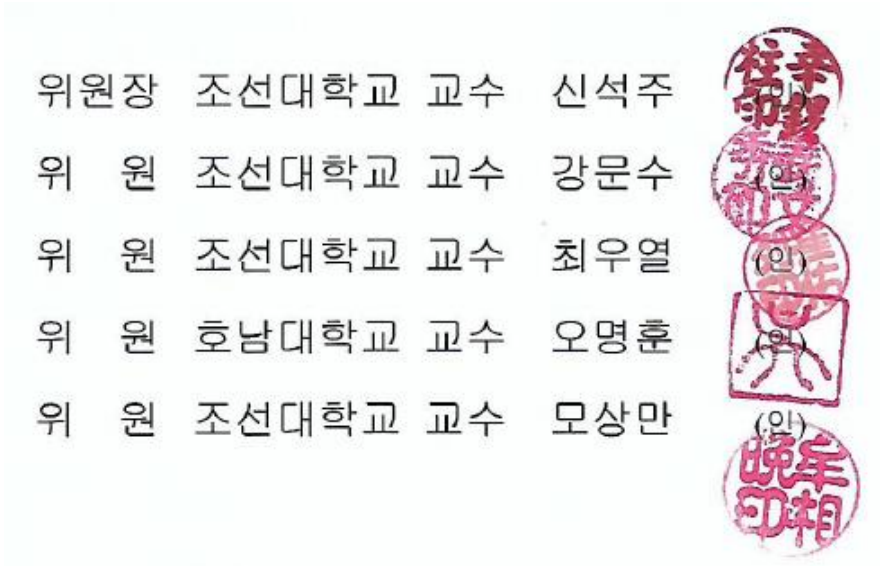
April 2023

Graduate School of Chosun University

Department of Computer Engineering

Muhammad Morshed Alam

# 알람 무하마드 물셰드의 박사학위논문을 인준함




2023년 6월

조선대학교 대학원

# Table of Contents

# List of Table

# List of Figure

# ABSTRACT

## Routing Algorithms Based on Reinforcement Learning for Unmanned Aerial Vehicle Swarm Networks

Muhammad Morshed Alam
Advisor: Prof. Sangman Moh, Ph.D.
Department of Computer Engineering
Graduate School of Chosun University

In recent years, unmanned aerial vehicles (UAVs) have attracted increased attention from academic and industrial research communities for their wide range of potential applications in military and civilian domains. Owing to the flexible three-dimensional (3D) mobility, on-demand deployment and low cost, a collaborative UAV swarm networks (UAVSNs) can effectively execute emerging missions such as surveillance and communication coverage in an emergency. Due to the high mobility, dynamic time-varying topology, limited onboard energy, and frequent link breakages, data packet routing from remote UAVs to base station (BS) produces excessive retransmissions, long delays, strong mutual interferences, energy holes, and loops. Therefore, in UAVSNs, collaborative mobility control, path stability defined by predictive 3D link duration (LD), link signal-to-interference-plus-noise ratio (SINR), delay, and residual energy of UAVs should be jointly taken into consideration to improve both mission and packet routing performance because they are tightly coupled. To effectively address the above challenges, we jointly consider the collaborative mobility control and multi-link quality metric packet routing in UAVSNs by utilizing nature-inspired swarming behavior-based adaptive mobility control and reinforcement learning, which are suitable to perform multi-objective optimization in a resource constraint dynamic UAVSNs.

In the first work, we propose a joint topology control and routing (JTCR) protocol comprising three modules to perform a crowd surveillance mission utilizing a UAVSN. The first JTCR module provides virtual force-based mobility control

(VFMC), which controls the mobility of UAVs to track the mobile ground target while ensuring stable connectivity in aerial links. The second module provides energy-efficient mobility-aware fuzzy clustering that clusters the UAVSN topology to aggregate the sensed data to each cluster head (CH) by utilizing the UAV mobility provided by the VFMC. The third module provides topology-aware Q-routing, which routes the aggregated data from CH UAVs to the BS by selecting an optimal path in terms of network delay, path stability, and energy consumption of UAVs.

In the second work, we propose a Q-learning (QL)-based routing protocol inspired by adaptive flocking control (QRIFC) to execute a surveillance mission in a post-disaster scenario. In QRIFC, the proposed adaptive flocking control algorithm generates optimal mobility with fairness in travel distance for each UAV to control the optimal node density. It also addresses the trade-off between aerial coverage and quality of service in connectivity by imposing constraints on the minimum separation distance and maximum allowable inter-UAV spacing using two-hop neighbor information. Additionally, it provides a stable LD between neighboring UAVs and optimizes the control overhead. Furthermore, QL performs multi-objective optimization by utilizing a new state exploration and exploitation strategy to select an optimal routing path in terms of delay, stable path selection defined by predictive 3D maximum-minimum LD, and energy consumption of UAVs.

In the last work, we propose joint trajectory control, frequency resource allocation, and packet routing (JTFR), in which link utility is maximized by jointly considering the link stability, SINR, queuing delay, and residual energy of UAVs. Finding the optimal link utility is extremely challenging because of the complex sequential decision-making process based on multiple constraint parameters in cross layer design. JTFR employs adaptive distributed multi-agent deep deterministic policy gradient coupled with swarming behavior to obtain the optimal solution. For each UAV, an actor network is established by utilizing a long short-term memory-based state representation layer containing two-hop neighbor information to adopt the dynamic time-varying topology. Subsequently, a scalable multi-head attentional critic network is set up to adaptively adjust the actor-network policy of each UAV by collaborating with neighbors.

Extensive computer simulation is performed to evaluate the performance of each proposed protocol by rigorously comparing it with existing baseline protocols.

According to our performance study, the proposed JTCR shows 34% better tracking-coverage rate, 9.5% better connectivity rate, 7−21% average better packet delivery ratio (PDR), 9−37% less average end-to-end delay (AE2ED), and 15−23% less energy consumption in comparison to existing routing protocols. This is mainly enabled by the realistic mobility control of the UAV swarm at the reasonable cost of control overhead and a smaller number of retransmissions. The proposed QRIFC outperforms existing routing protocols by 21−40 % less AE2ED and 9−23% higher average PDR with fewer retransmissions. Similarly, the proposed JTFR outperforms existing routing protocols by 30−60% less AE2ED, 15−32% better average PDR, and 20−46% less energy consumption.

# 한글 요약

## 무인 비행체 군집 네트워크를 위한 강화 학습 기반 라우팅 알고리즘

알람 무하마드 물셰드
지도교수: 모상만
컴퓨터공학과
조선대학교 대학원

최근 무인 비행체(UAV)는 군사 및 민간 영역에서 광범위한 잠재적 응용 분야로 학계 및 산업 연구 커뮤니티의 관심을 끌고 있다. 유연한 3 차원(3D) 이동성, 주문형 배치 및 저렴한 비용으로 인해 협업 무인 비행체 군집 네트워크(UAVSN)는 비상 시 감시 및 통신 범위 확대와 같은 새로운 임무를 효과적으로 수행할 수 있다. 높은 이동성, 동적 토폴로지, 제한된 에너지 및 빈번한 링크 손상으로 인해 원격 UAV 에서 기지국(BS)으로의 데이터 패킷 라우팅은 과도한 재전송, 긴 지연시간, 강력한 상호 간섭, 에너지 소모 불균형 및 전송 루프를 발생시킨다. 따라서 UAVSN 에서는 협업 이동성 제어, 예측 3D 링크 지속시간(LD)으로 정의된 경로 안정성, 링크 신호 대 간섭 잡음비(SINR), 지연시간 및 잔여 에너지가 긴밀하게 결합되어 있기 때문에, 임무 수행 및 라우팅 성능을 모두 향상시키기 위해 동시에 고려되어야 한다. 이를 효과적으로 해결하기 위해, 본 연구에서는 군집 행동 기반 적응형 이동성 제어 및 강화 학습을 활용하여 UAVSN 에서 협업 이동성 제어 및 다중 링크 품질 기반 라우팅을 고찰한다. 이같은 접근 방법은 자원 제약 동적 UAVSN 에서 다중 목적 최적화를 수행하는데 적합하다.

첫 번째 연구에서는 UAVSN 을 활용한 군중 감시 임무를 수행하기 위해 세 개의 모듈로 구성된 토폴로지 제어 및 라우팅 결합(JTCR) 프로토콜을 제안한다. JTCR 의 첫째 모듈은 무선 링크에서 안정적인 연결을 보장하면서 이동식 지상 목표를 추적하기 위해 UAV 의 이동성을 제어하는 가상 힘 기반 이동성 제어(VFMC)를 수행한다. 둘째 모듈은 VFMC 에서 제공하는 UAV 이동성을 활용하여 UAVSN 토폴로지를 클러스터링하여 감지된 데이터를 각 클러스터 헤드(CH)로 집계하는 에너지 효율적인 이동성 기반 퍼지 클러스터링을 수행한다. 셋째 모듈은 네트워크 지연시간, 경로 안정성 및 UAV 의 에너지 소비 측면에서 최적의 경로를 선택하여 수집된 데이터를 CH UAV 에서 BS 로 전송하는 토폴로지 기반 라우팅을 수행한다.

두 번째 연구에서는 재난 상황에서 감시 임무를 실행하기 위해 적응형 군집 제어로부터 착안한 Q-러닝(QL) 기반 라우팅(QRIFC) 프로토콜을 제안한다. QRIFC 에서 제안된 적응형 군집 제어 알고리즘은 최적의 노드 밀도를 제어하기 위해 각 UAV 에 대해 이동 거리의 형평성과 함께 최적의 이동성을 생성한다. 또한 2 홉 이웃 정보를 사용하여 최소 이격 거리와 허용 가능한 UAV 상호 간격에 제약을 둠으로써 통신 가능 범위와 연결 서비스 품질 간의 절충점을 찾는다. 또한, 인접 UAV 간 안정적인 LD 를 제공하고 제어 오버헤드를 최소화한다. QL 은 새로운 상태 탐색 및 이용 전략을 활용하여 예측 3D 최대-최소 LD 로 정의된 안정적인 경로 선택 및 UAV 에너지 소비 등의 측면에서 최적의 경로를 선택하여 다중 목적 최적화를 수행한다.

마지막 연구에서는 UAV 링크 안정성, SINR, 지연시간 및 잔여 에너지를 함께 고려하여 링크 이용률이 최대화되는 경로 제어, 주파수 할당 및 라우팅 결합(JTFR) 프로토콜을 제안한다. 교차 계층 설계에서의 여러 매개 변수를 기반으로 하는 복잡한 순차적 의사 결정 과정 때문에 최적의 링크 이용률을 찾는 것은 매우 어렵다. JTFR 은 최적의 솔루션을 얻기 위해 군집 동작과 결합된 적응형 분산 다중 에이전트 심층 결정론적 정책을 사용한다. 각

UAV 에 대해, 동적 토폴로지를 채택하기 위해 2 홉 이웃 정보를 포함하는 단기 메모리 기반 상태 표현 계층을 활용하여 행위자 네트워크(actor network)를 설정한다. 이후 확장 가능한 다중 헤드 기반 비평가 네트워크(critic network)가 설정되어 이웃 노드와 협력하여 각 UAV 의 행위자 네트워크 정책을 적응적으로 조정한다.

제안된 각 프로토콜의 성능을 기존 프로토콜들과 비교 평가하기 위해 광범위한 컴퓨터 시뮬레이션을 수행한다. 우리의 성능 평가에 따르면, 제안된 JTCR 은 기존 프로토콜에 비해 34% 향상된 추적 범위 속도, 9.5% 향상된 연결 속도, 7–21% 향상된 평균 패킷 전송 비율(PDR), 9–37% 감소된 평균 종단 간 지연(AE2ED) 및 15-23% 감소된 에너지 소모를 보여준다. 이는 주로 제어 오버헤드와 감소와 적은 수의 재전송에 따른 UAV 군집의 현실적인 이동성 제어에 기인한다. 제안된 QRIFC 는 AE2ED 를 21–40% 줄이고 재전송 횟수를 줄이면서 평균 PDR 을 9–23% 더 높여 기존 라우팅 프로토콜을 능가한다. 또한, 제안된 JTFR 은 AE2ED 30–60%, 평균 PDR 15–32%, 에너지 소비 20–46%까지 기존 라우팅 프로토콜을 능가한다.

# 1. Introduction

Unmanned aerial vehicles (UAVs) equipped with various types of sensors have numerous applications in military and civilian fields, including in search and rescue operations, surveillance, wildfire monitoring, agricultural remote sensing, relay networks, providing wireless coverage to ground users (GUs) as aerial base stations (ABSs), and post-disaster relief operations [1], [2]. The rapid development of network technologies is envisioned to enable the autonomous operation of multi-UAV networks for any type of mission. It includes advanced sensors [3], control and battery technologies, global positioning systems (GPS) or GPS-denied positioning techniques, incorporations of various artificial intelligence [4], machine learning techniques [5], obstacle avoidance techniques [6], and advanced routing protocols [7]–[13].

Compared to a single-UAV system with limited energy, a limited computational capacity, poor functionality, fixed sensor field of view, and poor survivability, a collaborative UAV swarm networks (UAVSNs) provides a wide range of advantages such as wider coverage, high survivability, high flexibility, efficient task allocation, and adaptability. Owing to the unique features of UAVSNs such as their self-organizing and self-healing distributed autonomous sensing, three-dimensional (3D) positional adjustment, and low cost, the integration of UAVSNs into other applications is becoming popular, such as data collection in wireless sensor networks (WSNs) [14], [15] or from internet of things (IoT) devices [16], data ferrying in delay tolerant networks [17], mobile edge computing services to low power IoT devices [18], UAV-aided vehicular ad hoc networks (VANETs) [19], energy harvesting for low-power IoT devices [20], [21], and ABSs [22], [23]. UAVSNs can serve as ABSs to provide better network coverage, as they have a higher probability of acquiring a line-of-sight (LoS) to GUs.

UAVSNs can form a multi-hop network consisting of flying nodes and a few fixed base stations (BSs); these are known as flying ad hoc networks (FANETs), and do not require any fixed infrastructure. In a FANET, using a hop-by-hop relaying method, a UAV can perform near real-time data delivery to a BS or other UAVs. In FANETs, owing to the high mobility and limited transmission range, the state of the UAV swarm (position, velocity, and direction) changes frequently. UAVSNs also changes the formation topology in a dynamic

environment, owing to mission requirements such as evenly monitoring a particular area, tracking mobile GUs, and leaving or joining the aerial network before and after replenishing energy through a charging scheduling algorithm (CSA) [24]. This situation imposes many challenges on the control process of UAVSNs, such as developing a formation control law, maintaining stable links, selecting a multi-hop path for relaying data, guaranteeing the quality of service (QoS), and optimizing energy consumption.

Owing to changes in the relative speeds of UAVs to meet the mission performance requirements such as maximizing the coverage rate of the mission area [25], [26], tracking high-density areas of GUs [23], tracking mobile targets [27], [28] and ensuring motion fairness among UAVs [29], [30], there are massive challenges to the communication performance of a UAVSN. To meet the mission performance requirements, the key challenges in a UAVSN topology are in maintaining the communication performance, such as providing a stable link duration (LD) in the aerial links to reduce the frequent link breakages, retransmissions, and high latency. The LD between two neighboring UAVs is a function of the UAV transmission range, relative distance, and relative velocity with neighboring UAVs, as shown in Figure 1.1 [31], [32] The LD parameter defines how long a neighboring UAV will stay with another transmission range, and it widely used to construct topologies [33], optimize the control overhead [34], find the fitness of a node to identify a cluster head (CH), a dominating set [35], and make routing decisions [31], [34], [36].

Through the efficient utilization and optimization of the constrained resources of a UAVSN, a topology control algorithm (TCA) can be used to support both medium access control (MAC) and routing protocols. TCA allows to develop a comprehensive FANET by performing the joint optimization on control and communication to balance the mission and communication performance. The control sections mainly consider (i) the trajectory or mobility control to maintain trade-off between aerial coverage and connectivity by maintaining optimal node density and avoiding inter-UAV collision, (ii) topology construction and adjustment, and (iii) adaptive control of hello interval (HI) to address the trade-off between topology prediction accuracy and control overhead. The communication sections mainly include not only the allocation of resources (i.e., timeslot, frequency, and power) for transmission scheduling but also optimal relay selection. Thus, TCAs can optimize the energy consumption of UAVs, reduce inter-UAV interference and control overhead, maximize the network throughput, and ensure stable LD.

## 1.1 Components of UAV Swarm Networks

Here, we briefly discuss the UAVSN, its components, and functionalities. UAVSN imitates the behavior of swarm intelligence (SI) and collaborates with the terminal BS/IoT devices/sensors/GUs to form an autonomous self-organized multi-UAV communication system known as the FANET. In FANETs, the terrestrial devices are replaced by the UAVs, and can establish communication on-demand or in any type of emergency scenario without requiring any fixed infrastructure. Each UAV in the swarm is capable of sensing, executing a computationally intensive task locally or in an edge/cloud server, performing communication, caching data, and working as a router to forward remote UAVs sensing data to a BS for further processing. Thus, UAVSNs have two sections: a terrestrial section and non-terrestrial section, as illustrated in Figure 1.1.

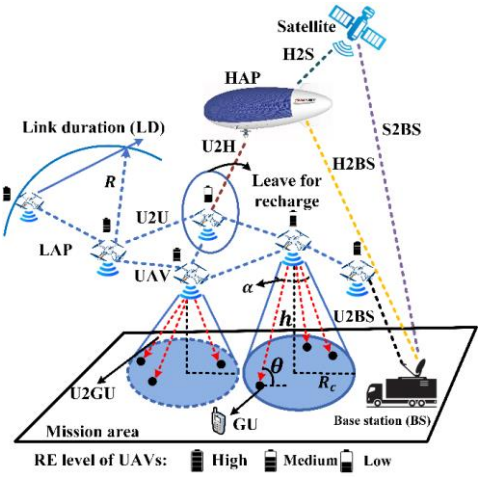



Figure 1.1 An on-demand UAV swarm network and its different components.

The terrestrial section usually comprises mobile ground vehicles, fixed BS, mobile or fixed charging stations (CSs), edge server, GUs, sensors, and IoT devices. UAVSNs can collaborate with existing terrestrial BS or an emergency vehicle on an on-demand basis to extend the network capacity. The BS can be equipped with an edge server [37] and CSs [38]. UAVs can offload their computationally intensive tasks to the edge server [39]. When the residual energy (RE) of a UAV reaches the minimal threshold, the UAV can leave the aerial network utilizing a CSA to get energy replenishment at a particular wireless CS [38].

The non-terrestrial section consists of a set of homogenous or non-homogenous UAVs working collaboratively at different altitudes; these are known as low altitude platforms

(LAPs) and high-altitude platforms (HAPs). Usually, each UAV has four major modules such as flight control, energy management, computation, and communication module. The mobility information of UAVs in 3D space can be described by six degree of freedom such as surge, heave, sway, pitch, yaw, and roll [40].

There are a variety of communication links in UAV networks according to the mission planning and control method. These communication links can be classified as air-to-ground (A2G) links and air-to-air (A2A) links. In general, the A2G links include UAV to BS (U2BS), UAV to GU (U2GU or GU2U), HAP to BS (H2BS), and satellite to BS (S2BS). Similarly, the A2A links are UAV to UAV (U2U), UAV to HAP (U2H), and HAP to satellite (H2S). UAVs can directly communicate with the satellite, especially with the GPS to localize themselves in global coordinates. Usually, LAPs can communicate with the BS using U2BS downlinks. U2BS links have low cost in terms of transmission power, latency, and path loss in LoS cases. However, the quality of the U2BS links significantly degrades in the no line of sight (NLoS) cases. To this end, depending on the signal quality and type of mission planning, LAPs can also utilize the U2H and H2S uplinks to communicate with the BS [31].

In a LAP, the radius of the disk size sensor coverage of the UAVs to the ground terminal $R_c$ is a function of the UAV altitude $h$ and FoV $\alpha$, as shown in Figure 1.1. With an increasing altitude $h$, $R_c$ increases and the probability of getting the LoS also increases. However, simultaneously, the path losses are also increased, as a result, the UAV altitude should be controlled within an optimal altitude range depending on the mission environment, distribution of the GUs, and application [41].

In UAVSNs, the communication traffic comprise sensing data collected by each UAV and the control message (i.e., mobility information of the UAVs collected by onboard GPS, inertial measurement unit (IMU) and LiDAR sensors). UAVs can exchange control messages among themselves or with BS to maintain a local neighbor list, control the mobility of the swarm, generate multi-hop routing paths for communicating with BS and relaying sensing data, maintain cooperative mission planning, and execute task assignment [42], [43].

## 1.2 Design Issues of Routing Protocols in UAVSNs

Controlling the WSN topology is less complex, as sensor nodes are mostly static, or have very little two-dimensional (2D) mobility. In [44], [45], the TCAs for WSNs were studied by classifying them into four categories: transmission power adjustment,

transmission power mode switching, clustering, and hybrid mode. In general, power mode switching is not suitable for FANETs, as UAVs require energy to stay in the air and to communicate with neighboring nodes to plan a collaborative mission. Moreover, the UAV flying energy consumption is sufficiently larger than the communication energy consumption [35]. In VANETs, node's mobility is constrained by roads; as a result, predicting the topology is much easier than in FANET, and the node energy is not constrained [46]. FANETs differ from other ad hoc networks in terms of the node density, 3D mobility [47], inter-UAV collision, restricted trajectories owing to collaborative motion planning and constraint mission boundary, limited energy of UAVs, wind disturbance, and frequent topology alterations for meeting mission performance [48], [49]. The conventional TCAs and routing protocols related to MANETs, VANETs, and WSNs are not suitable for high-speed UAVs, because the sensors or ground vehicles in these networks make horizontal 2D movements with less mobility, UAVs have 3D mobility in horizontal and vertical directions [40]. The key issues to design the TCA and routing protocols for UAVSNs or FANETs are briefly discussed.

### 1.2.1 Connectivity

In UAVSNs, the data collected by UAVs need to be transmitted to a BS by relaying through an optimal reliable multi-hop path that gives the optimal delay, highest link survival time [31], and produces balance in energy consumption for all of the UAVs. In addition, in cooperative missions, UAVs need to exchange information for mission coordination. To cope with the dynamic topology and limited energy, the UAVs should establish a stable formation by maintaining relative distances and velocities. To maintain strong connectivity, UAVs should not frequently fly away from each other, and they should consider a few communication constraints such as signal-to-interference-plus-noise ratio (SINR) level by maintaining acceptable relative distance and transmission power, maintaining a minimum safe distance, and maintaining a certain maximum attainable speed under the maximum attitude angles to adjust the direction.

### 1.2.2 Coverage

The designed algorithm should maintain a proper SINR level to achieve an acceptable data rate for all wireless links. It should consider the trade-off between coverage rate and QoS in aerial connectivity according to the application of UAVSNs. Owing to the fixed communication range and relatively high costs of UAVs, it is impractical to deploy enough

UAVs to cover a large target area over. Therefore, UAVs need to move around to confirm that each area is covered sufficiently well, which is known as dynamic coverage. The static coverage provides fixed coverage density, which may give a constant mission performance, i.e., GUs detection [24]. However, it is not desirable to cover a particular target area most of the time while leaving the remainder only poorly covered. Additionally, the density of mobile GUs may not be equal in the mission area [50]. As a result, UAVs need to move slowly to monitor the mission area with fairness [29], detect the maximum mobile targets [24], or serve the maximum number of GUs as ABS [51]. For obtaining maximum coverage with QoS in connectivity of U2U and U2BS links, the UAVSN deployment depends on a few important parameters such as the proper assessment of GUs distribution and their mobility model [52], [53] and 3D positions adjustment of UAVs to serve maximum GUs.

### 1.2.3 Distributed Algorithm

A centralized algorithm requires global information, and all of the UAVs must send data to the central controller. Therefore, this approach consumes a high bandwidth in the backbone network and incurs high computational cost. It also encounters scalability issues. In contrast, in a distributed algorithm, each UAV maintains continuous awareness of its one or two-hop neighbors and cooperates collaboratively to achieve the desired goal. Hence, this method requires less computational complexity. However, significant theoretical challenges arise when controlling UAVSNs based on partial information.

### 1.2.4 Tolerance to Communication Delay and Localization Error

The optimal allocation of resources (such as UAV transmission power, frequency, and timeslots) can significantly enhance SINR and delay in inter-UAV communication. Time delays, while broadcasting control packets among UAVs may cause the UAVs to record inaccurate locations for their neighbor UAVs. This may affect the formation controller performance [54]. It can also produce an error in topology prediction; this sequentially affects the MAC and routing protocol performance, owing to instability in the links. The major causes of time delays in UAVSNs are the inter-UAV distances, node densities, sizes of control packets and data packets, transmission power, and effects of the multi-path fading wireless channel [55]. GPS-equipped UAVs may have localization errors of 10−30 m [56]. Therefore, the designed control model should be sufficiently tolerant to GPS errors to maintain updated routing path [57]. A stable routing path can avoid link breakages in dynamic UAVSN, which can significantly reduce unnecessary retransmissions.

### 1.2.5 Collision Avoidance and Tolerance to UAV Failure

During flocking, UAVs should maintain a minimum distance from one another to avoid internal collisions [58]. However, in many cases, a UAV swarm may require partitioning in the network topology, especially when they flock in a complex-obstacle environment or track any moving target. The designed algorithm should adaptively update the network topology and perform formation reconfiguration as soon as the obstacle is passed and the UAVs re-enter each other's communication ranges [34].

### 1.2.6 Optimal Control Overhead and Number of Transmissions

The designed algorithm should be simple and should generate optimal control messages to maintain the dynamic time-varying topology. In FANETs, each UAV senses topological changes by actively monitoring a neighbor set and periodically exchanging hello packets with each other so as to share the updated mobility information. A shorter HI enables higher accuracies in real-time neighbor discovery and ensures a more up-to-date routing path. Consequently, it increases the control overhead as a penalty. Moreover, it is suggested that the sensing data are logically aggregated to minimize the number of transmissions [59].

### 1.2.7 Link Bidirectionality

The edges in the constructed topology should be bidirectional or symmetrical. Considering the MAC and network layers, the UAVs need to communicate bidirectionally. For instance, many MAC protocols such as the IEEE 802.11 standard protocols require an undirected graph topology to send a clear to send or request to send message before data transmission, to avoid data collisions and solve the hidden and exposed terminal problem [60].

### 1.2.8 Redundancy

The designed algorithm should be aware of all types of failure issues, such as UAV failures caused by any type of hardware failure, software failure, or even energy limitation. In such cases, the self-healing UAV swarm should recover the neighboring connectivity as soon as possible without creating any partition in the network during flocking.

### 1.2.9 Stability and Scalability of Dynamic UAVSNs

One key research challenge is in maintaining the formation stability and autonomous scalability for a dynamic UAVSN. The designed controller should achieve the formation

and adaptively maintain the formation stability by performing the formation reconfiguration as UAVs are joining or leaving the swarm due to mission requirement or energy replenishment [34], [61]. A stable formation can ensure link stability.

### 1.2.10 Optimizing UAV Energy Consumption

To prolong the network lifetime, UAV energy consumption optimization is required. The flight control, communication, and computation module are the major sources of UAV energy consumption. The optimal trajectory planning with motion fairness [29], energy-efficient MAC, and routing protocol [62], and offloading the computationally intensive tasks to the edge servers [63] can help to minimize the energy consumption of UAVs. The TCAs can control the optimal node density by adjusting the inter-UAV distances according to the transmission range to ensure the connectivity rate and less interference. Additionally, the optimal node density reduces the competition in MAC layer channel access, which can significantly reduce number of retransmissions [35]. The propulsion energy consumption by the UAV rotors is proportional to the distance it travels during flocking. Hence, to reduce energy consumption, a UAV swarm should have a collaborative mobility control scheme so that each UAV gets an optimal distance to travel [23], [29], [50], [57].

### 1.2.11 Convergence Time

The convergence time of the designed algorithm should be as fast as possible. Because the nodes have high mobility, the algorithm should return the optimum solution as quickly as possible to maintain the up-to-date topological changes [35].

## 1.3 Organization of Thesis

Rest of the thesis is organized as follows: In Chapter 2, existing topology control algorithms and routing protocols in UAVSNs will be reviewed and qualitatively compared along with their advantages and limitations. In Chapter 3, the joint topology control and routing (JTCR) for UAVSN is proposed and evaluated to execute a crowd surveillance mission. In Chapter 4, the Q-learning-based routing inspired by adaptive flocking control (QRIFC) is proposed to design collaborative mobility models and routing protocols in FANETs. In Chapter 5, the joint trajectory control, frequency allocation, and routing (JTFR) for UAVSN is proposed based on cross layer design and evaluated.

# 2. Related Works

In UAVSNs, the relative mobility, link delay, and path stability are highly coupled with each other. TCA can ensure aerial coverage and network connectivity by controlling the relative mobility of UAVs by adopting swarming behavior. Thus, it can improve the performance of routing protocols significantly by ensuring appropriate node density, smooth trajectory of UAVs, and enhancing the topology prediction accuracy. In this Chapter, the related literature review for the existing TCAs and routing protocols in UAVSNs are briefly discussed along with their advantages and limitations. Based on their limitations and identified research gaps, the key open issues and research challenges is summarized.

## 2.1 Topology Control for UAVSNs

A TCA is a mechanism for coordinating and optimizing the position, relative velocity, direction, and orientation of UAVs in 3D space according to the transmission range of each UAV that can generate a network with certain properties to achieve mission and communication goals. The main objectives of TCAs are to provide stable connectivity in high-speed FANETs, while ensuring a safe distance for avoiding collisions and meeting SINR constraint to ensure QoS in the U2U links, to maximize the coverage to perform the mission successfully, and to optimize the energy consumption of UAVs and network delays. In FANETs, the optimal positioning of the UAVs depends on mobility control parameters, such as the speed, direction, transmission power, and density of UAVs.

The topology control problem is an iterative process, in which the first step comprises topology construction (TC) and the second step comprises topology adjustment (TA). Typically, the TC has high computational complexity, as it generates the FANET topology from scratch. Therefore, this construction process requires a powerful algorithm that may be computationally expensive, as it must satisfy all of the constraints to provide a feasible topology. After obtaining the optimal reduced topology, a less computationally expensive TA should begin operating, so that it can rapidly adjust the topology in real-time to maintain the optimal form according to the dynamic conditions. This algorithm can also trigger a new topology construction phase by running the TC when any constraints are violated in the existing network topology over a certain period. More details discussion about TCA is given in [40]. In the next section, the relation with TCA, MAC protocol, routing protocol, and formation control is briefly discussed.

### 2.1.1 TCA Interaction with MAC Protocol

During flocking in a dynamic environment, the node density and inter-UAV distance changes frequently. Adaptive adjustment of UAV trajectory according to inter-UAV distance and physical layer transmission power can ensure optimal node density and reduces the possibility of interference [64]. Even though the MAC layer can control the transmission power, it cannot consider packet-level power control by itself. This is because this layer does not include information on the exact power required for each hop according to the inter-UAV distances at each time interval of the mobility updates [65], [66]. Therefore, according to the inter-UAV distances obtained from the TCA during flocking, the UAVs can set optimal transmission power as shown in Figure 2.1, so as to simplify the network topology by removing the redundant longest edges and minimize the interference in the inter-UAV communication.

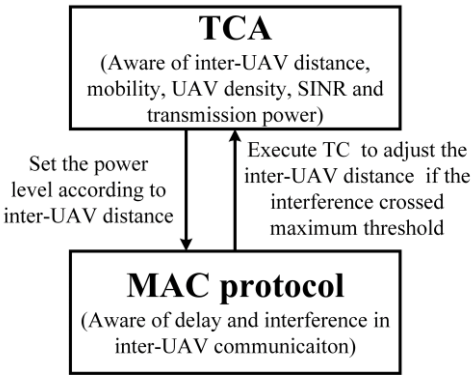



Figure 2.1 Collaboration between TCA and MAC protocol.

A TCA can support the network layer by offering a more efficient neighbor list and reducing the possibility of data collision and interference at the MAC layer. Similarly, the MAC layer can trigger the execution of the TCA to adjust the density of UAVs and contention window in case it discovers the interference crossed a maximum threshold during neighboring UAVs communication [67]. The proper assessment of inter-UAV distances according to the physical layer transmission range and the updated neighbor list given by a TCA helps to maintain optimal node density. Additionally, adaptively adjusting the size of contention window according to the node density helps to precisely estimate the data packet collision and successful packet transmission probability in UAVSNs. In [67], the authors

investigated the relation between MAC protocol and flocking control [68] to adaptively adjust the size of the contention window according to the node density of UAVs by calculating the inter-UAV distances with neighboring UAVs and updated neighbor list. It also helps to calculate the precise MAC layer contention for carrier sense multiple access with data collision avoidance.

### 2.1.2 TCA Interaction with Routing Protocol

Routing protocol is responsible for finding and maintaining the reliable path between a source and destination UAVs. When a UAV needs to transmit a packet to another UAV or BS, it finds the reliable multi-hop routing path with the help of a routing protocol for a particular destination in terms of optimal delay, UAV RE, and LD. The TCA controls the mobility of UAVs so that a strong neighborhood is established between UAVs for each timeslot that ensures sufficient LD among neighboring UAVs to avoid frequent link breakages. Thus, it reduces the number of retransmissions and offers a stable neighbor list to the routing protocol to generate an updated routing path at each data interval. As shown in Figure 2.2, the TCA constructs the UAVSN topology to maintain an updated neighbor list for each UAV by executing the TC that detects the optimal mobility of each UAV according to the mission requirements at each timeslot. Additionally, it can also trigger the TA in case it detects considerable fluctuations in the active neighbor set over time by sensing the instantaneous degree of violation in imposed safety and SINR constraints with neighboring UAVs due to the changes in mobility during flocking.

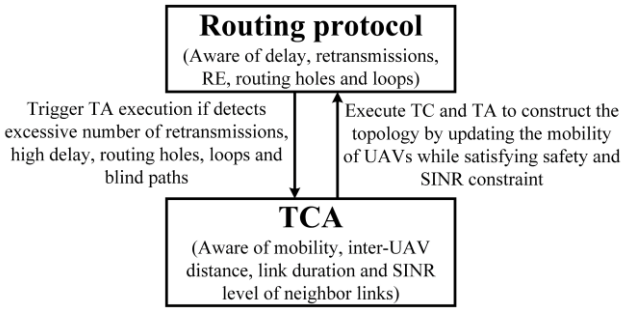



Figure 2.2 Collaboration between TCA and routing protocol.

Therefore, instead of passively waiting for the routing protocol to separately update the neighbor list, a mobility update phase can be triggered by the TCA to detect updated

neighbors if the neighboring UAVs of the source UAV stays too far or are detected within an interference zone, or even if the imposed safety distance and SINR constraints are violated. This leads to a faster response to a time-varying topology. It also reduces the packet loss and routing holes by assuring an updated neighbor list with the desired SINR levels for the respective source UAVs to perform data relaying. Similarly, the routing layer can also trigger the TA execution in the TCA, e.g., when it detects an excessive number of retransmissions, link breakages, and routing holes, as in such cases, it assumes that there have been many changes in the network topology since the last execution of the TC.

### 2.1.3 TCA Interaction with Formation Control

A formation control mechanism can maintain a relatively steady state in a UAVSN by matching the velocities and distances between neighboring UAVs and helping to avoid inter-UAV collisions [69]. The output of the UAV formation control, typically consisting of current and target WPs for all UAVs, can predict the topology of the network [34]. Proper formation control provides stability in the network topology, boosts the network's transmission efficiency and task execution capacity. In [50], the authors studied flocking control protocols for ensuring stability in maintaining a safe distance between UAVs and simultaneously preserving aerial connectivity. In general, the important tasks required to control UAV swarm formations are maintaining the relative positions, relative velocities, and directions of the UAVs, avoiding external obstacles and inter-UAV collisions, and moving the entire formation or the center of the mass of the formation along a pre-defined trajectory. It also ensures that the fleet of UAVs remains in a relatively steady-state with enough LD to exchange information for cooperative coordination and to perform data collection. Therefore, formation control techniques have a strong relationship with the network topology control. It can easily meet the requirements of the TCAs of UAV networks and cooperate with the routing protocol to generate a stable routing path.

Flocking describes the aggregated behavior of multi-agent systems, where a group of UAVs interacts to achieve common goals. This approach is a commonly used approach to control UAV swarm formations. The common properties of a swarm network include its self-organization, self-formation, and collision avoidance. The distributed self-organization and robustness of the cooperative control of UAV swarms are similar to the decentralized and self-organized characteristics of biological groups such as ant colonies, bee colonies, flocks of birds, and schools of fish [70]. Large-scale UAVSN coordination, as inspired by

these self-organized biomorphic flight pattern algorithms, enhances the efficiency of the autonomous distributed operations of UAVSN. Thus, mimicking the autonomous swarm behaviors of animals makes the complex multi-UAV coordination problem easier to address, and enhances the autonomy of UAV networks. A cooperative UAVSN flying in a complex environment can maintain a robust topology in a dynamic environment by generating collective motions adopting the three rules of the Boids flocking proposed by Reynolds in 1986 [71]–[73]. These three rules concern separation, cohesion, and alignment. Each rule produces a motion component vector. The weighted resultant of these three motion vectors determines the optimal mobility information such as the acceleration, velocity, direction, and position of each UAV in a swarm, as shown in Figure 2.3.

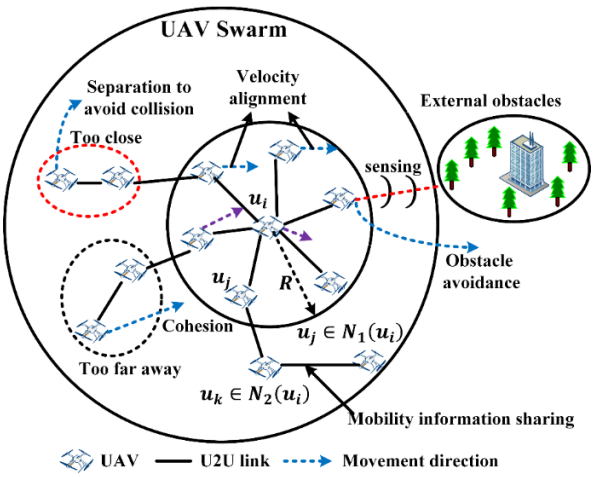



Figure 2.3 UAV swarm and its collective motion during flocking in a complex environment.

### 2.1.4 Taxonomy of TCAs

In this sub-section, we briefly discuss the existing TCA's for UAVSN topology by classifying them according to topology architectures and UAV roles in a UAVSN topology.

#### 2.1.4.1 Topology Architecture

According to the topology architecture, we classified TCA's in UAVSN as centralized, distributed, hybrid, and hierarchical structures, as shown in Figure 2.4.

In centralized TCAs, a central controller (i.e., a UAV in LAP/HAP or BS) has global knowledge regarding all of the UAVs and the mission environment. The global knowledge

includes the mobility information of each UAV in the swarm, locations of static/dynamic obstacles, shapes of the obstacles, and locations of tracked/untracked targets collected by the onboard sensor of each UAV. The central controller utilizes all this global knowledge for intelligent decision-making to control the swarm topology formation, perform obstacle avoidance, make a routing decision, execute CSA, and so on by sending the control commands to each UAV. The advantages of centralized topology are the design is simple and implementation is easy. The limitations of centralized topology are lack of robustness and high computational power because it may cause a single point of failure as the entire topology is coordinated by a single controller. It also offers low scalability due to the limited transmission range of the controller. The complexity of the central controller increases with the number of UAVs. The software defined network (SDN) [74] and deep reinforcement learning (DRL) [75] are the widely used centralized TCA for UAVSN.

In a distributed TCAs (also known as cooperative control), each UAV exchanges the mobility information and environmental data of a subset of UAVs, usually one or two-hop members of each UAV group. Each UAV has its own controller that can make decisions by independently utilizing locally collected information. For instance, each UAV can retain a similar velocity and constant relative distance with its neighbor UAVs according to the velocity and location information from its neighbors. The advantages of the distributed control are its high network scalability with low computational cost. UAVs can share their computation and communication burdens with neighbors. Thus, it has more flexibility and robustness. The limitations and challenges concern avoiding local optimality owing to a lack of global knowledge during decision-making. The virtual force [76], [77], virtual spring [78], artificial potential field (APF) [64], graph theory-based consensus [61], distributed model predictive control [79], [80], reinforcement learning (RL) [81], and game theory-based [82] algorithm are widely used TCA's for UAVSN.

In a hybrid TCAs, UAVSN has both a central controller and sub-controllers. Initially, the central controller (i.e., BS or a UAV in aerial network) constructs the UAVSN topology by collecting global knowledge. Then, each UAV makes decisions based on their local knowledge and adjusts their mobility. After a certain interval, the central controller collects mobility information from all UAVs and checks if the optimality (inter-UAV distance) of the network topology persists or not. According to that, if necessary, it reconstructs the UAVSN topology again from scratch. The advantages of the hybrid control architecture are

its global awareness, better scalability, and lower computational complexity compared to purely centralized control. The challenges in hybrid control are in minimizing the computational complexity and number of control packets. In [83], [84], the authors applied hybrid control methods to coordinate UAVSN missions.

In a hierarchical TCAs, the topology is controlled by selecting a set of sub-controllers, and each sub-controller considers an equal number of member UAVs for constructing a set of clusters. The sub-controller is also termed as the dominating node in the connecting dominating set (CDS) [85] or the CH in cluster-based TCAs. This type of control is similar to decentralized control. Usually, each sub-controller manages one cluster, considering its one-hop or two-hop members. In some design cases, the sub-controllers can act under the control of a root central controller (BS/HAP) that makes decisions and provides mission commands to the sub-controllers [86]. Then, the sub-controllers exchange the mission commands, but only with their respective cluster members (CMs). The CM UAVs execute the mission commands and give feedback to the sub-controllers, and the sub-controller also gives feedback to the central controller. However, considering the robustness and scalability, many design algorithms choose a distributed clustering process without the intervention of any central controller [87].

In homogenous UAVSNs, the sub-controllers are selected based on the fitness at each round of mobility updates. The key parameters used to select the fittest UAVs as the sub-controllers are the RE level, average LD with the neighbors [35], distances between the neighbors, distance from the BS, and degree of the node. Some algorithms create a multi-objective function considering multiple parameters to select the fittest UAV as the sub-controller [35], [59]. The sub-controller collects mobility information and sensing data from the CMs. Then, it performs data aggregation and compression, and makes a routing decision [88]. Hierarchical control is very suitable to large-scale UAVSNs, as it creates load balancing in energy consumption and minimizes the complexity in selecting the next forwarding node; in particular, routing decisions are transferred to a set of sub-controllers at each round of data transmission. The performance of the hierarchical control depends on the sub-controller lifetime, stability, mobility control, optimal number of sub-controllers, response time to construct the cluster topology, and number of control packets. The CDS-based [89], coalition game theory (CGT) [88], [90]–[93] and meta heuristic SI (i.e., particle swarm optimization (PSO) [56], grey wolf optimization (GWO) [59], and glowworm swarm

optimization (GSO) [94]) are the widely used hierarchical control methods to coordinate UAV swarm missions. More details explanation about each TCA along with their advantages and limitations are given in [40].

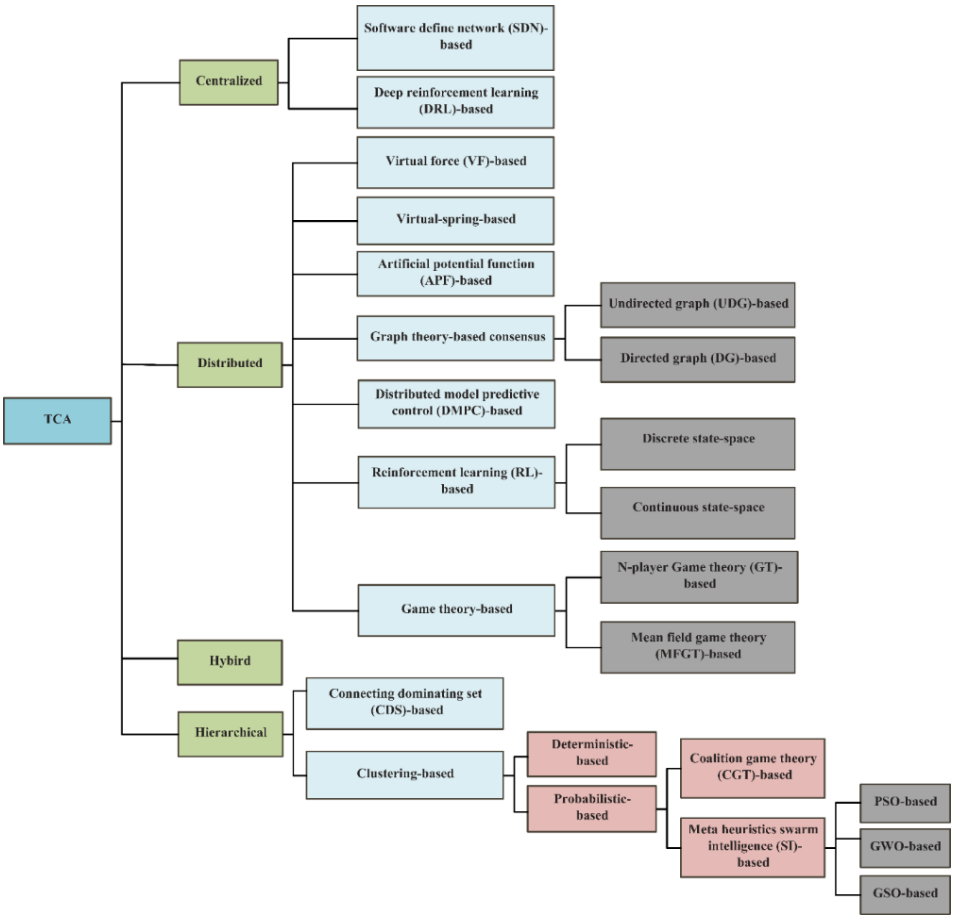



Figure 2.4 Taxonomy of TCAs in UAVSNs.

### 2.1.4.2 UAV Roles in UAVSN Topology

According to the UAV roles, TCAs can be classified as dynamic flat aerial mesh network (AMN) and leader follower (LF) topologies. In dynamic flat AMN topology, each UAV has the same role, and they work collaboratively to explore the mission area. Each UAV within the swarm communicates with one or two hop members and produces the mobility information to maintain AMN that can satisfy both connectivity and coverage needs.

In [38], [95], the authors utilized the virtual spring-based mobility model for UAV swarm to maintain strong flat AMN. The spring has both attractive and repulsive nature to produce resultant swarm mobility. The repulsive forces help to avoid inter-UAV collision and maintain inter-UAV distance to reduce the overlapping in sensor coverage to the ground terminal. The attractive forces help to maintain QoS in communication. The repulsive forces also can be used to avoid external obstacles. Similarly, in [23], [50], the authors utilized the VFs to maintain the AMN simultaneously UAVs serve as the ABS to the GUs.

One of the most common topology formation types is the LF strategy, owing to its flexibility and controllability. In UAVSNs, a single UAV designated as the leader flies independently to perform the assigned mission with the help of its own control mechanism, and other UAVs assist the leader by following its trajectory or a neighbor of the leader to form a specific formation. The leader controls the route of the entire swarm. The formation and stable topology are maintained through continuous adjustments of the distances, angles, and speeds between the leader and follower UAVs. Several LF strategies have been proposed, including single leader multiple followers (SLMF) [96], [97] multiple leaders and multiple followers (MLMF) [91], [98], and virtual leader-follower (VLF) [79], as shown in Figure 2.5.

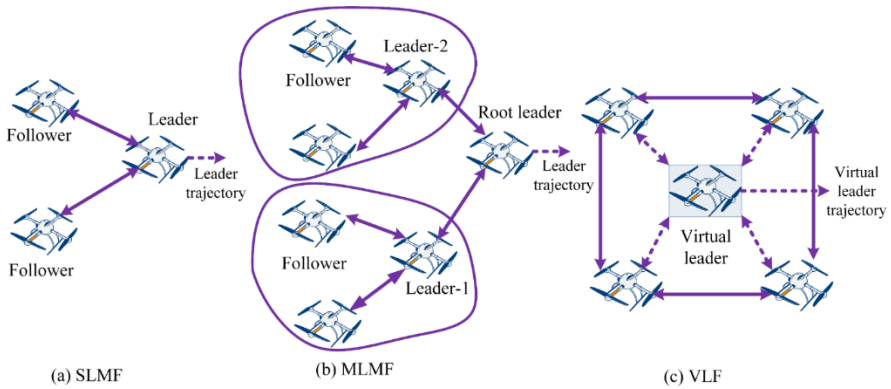



Figure 2.5 Different leader-follower (LF) topology. (a) Single leader multiple follower (SLMF), (b) Multiple leader multiple follower (MLMF), (c) Virtual leader follower (VLF).

In the SLMF strategy as shown in Figure 2.5(a), a single leader has multiple followers. In the MLMF strategy as shown in Figure 2.5(b), the swarm topology has multiple leaders, and each leader has multiple followers. An associated leader and follower can flock at different heights to execute the mission. The LF topology can consist of homogenous UAVs

or heterogeneous UAVs. The leaders can be predefined or can be selected on-demand basis. In a heterogeneous network, the leaders are pre-defined. Here, the leaders usually have higher power and computational capacities, as they collect mission data from the follower UAVs and perform mission coordination. In contrast, in a homogenous UAV network, a leader can be selected on an on-demand basis, such as according to the hierarchy index during flocking, or can be determined according to the fitness of the UAV [57].

The MLMF topology can maintain a better topology formation than that based on a single leader. In a VLF formation strategy as shown in Figure 2.5(c), a virtual leader can be considered at the center of the mass of the formation, as a moving reference point for the entire swarm formation. The virtual leader has a preplanned trajectory; this is also the desired trajectory of the entire UAV formation. In [79], the barycenter of the formation shape was considered as the virtual leader to track the reference trajectory, and the follower UAVs maintained the required distances, velocities, and angles with respect to the virtual leader to track the formation trajectory, thereby overcoming a weakness of the traditional LF flocking strategy.

The LF topology formation is very helpful for performing obstacle avoidance without partitioning the swarm. During flocking, if any UAV senses the presence of an obstacle within its sensing zone, it can declare itself as a local leader and guide neighboring UAVs to perform obstacle avoidance [57]. The LF topology simplifies the UAVSN topology control problem as the entire swarm trajectory depends on a single leader, but this model's lack of robustness in case of the failure of the explicit leader may destroy the formation of the entire swarm. Moreover, because all of the UAVs follow one leader, the convergence speed is slow.

### 2.1.5 TCA for Connectivity and Coverage

A significant amount of research has been performed on TCA in UAVSNs, while addressing the need for coverage and connectivity. In [99], proposed a coverage-efficient clustering algorithm (CECA) for large-scale FANETs that maximizes the area coverage by optimizing the number of CHs, the positions of UAVs, and their transmission power under delay constraints using gradient descent optimization. The CECA outperforms the mobility control-based clustering algorithm (MOOC) [100], which utilizes virtual forces to maximize coverage and maintain connectivity. However, the CECA encounter a slightly higher delay related to MOOC. In [56], an energy-efficient clustering for FANET was proposed by

optimizing a multi-objective CH selection fitness function using PSO. They also proposed a multi-hop routing technique to send data packets from the CHs to the BS. In [59], a multi-UAV clustering method was proposed by solving a multi-objective CH selection function jointly considering UAV RE, node degree (ND), and inter-UAV distances. They also proposed a compressed sensing-based inter-cluster routing scheme that reduces the number of transmissions. Because the positions of UAVs change frequently, a fitness calculation relying on inter-UAV distance without mobility prediction causes instability in FANETs. Moreover, in their inter-cluster routing, mobility prediction is not taken into consideration, which is very crucial in FANETs. In [92], a distributed stable clustering method utilizing the coalition game theory for FANETs was proposed, and the clustering minimizes delay in intra-cluster communication by using link subsistence probability. However, to achieve the Nash equilibrium, coalition game theory requires a long convergence time and needs to perform cluster switching operation frequently in high mobility.

In [83], the authors proposed a mission-critical FANET operation (MCFO) that jointly optimizes mission assignments and network topologies in a dynamic environment to enhance the mission and network performance. They constructed the FANET topology by optimizing the position of relay UAVs to ensure strong SINR among mission UAVs. They utilized two centralized high computational algorithms (i.e., PSO and role switching) and adjusts the topology using a gradient descent method under safety-distance and SINR constraints. In [95], the authors proposed a virtual spring-based mobility control for FANETs that maintains an aerial mesh network for post-disaster operations and addresses ground-user coverage, ensuring QoS in connectivity. In [76], the authors studied distributed topology control via virtual forces to efficiently control a FANET topology, enhancing both connectivity and coverage. In [34], the authors proposed a proactive distributed topology-aware routing method based on the relationship between the TCA and the routing layer by calculating the LD of neighboring UAVs. However, the UAV energy issue is not considered by them. In [87], the authors proposed a TCA for FANET by constructing a CDS-based topology and iteratively solving joint optimization to adjust the transmission power and position of UAVs using PSO. However, in the dominating set selection, they only considered the ND as a primary metric. According to the above review, we can say that joint consideration of mobility prediction, ND, and RE can give better UAV fitness evaluation and stability in CH lifetime in highly dynamic FANETs.

## 2.2 Existing Mobility Models and Routing Protocols

In this section, the existing mobility models, and routing protocols for UAVSNs are reviewed along with their advantages and limitations. The necessity of adopting a realistic mobility model and the advantages of a RL-based routing protocol for UAVSNs are also discussed briefly.

### 2.2.1 Existing Collaborative Mobility Models

In [9], [11], several mobility models for UAVSNs were reviewed including random direction, random waypoint, reference point group mobility, and Gauss-Markov. However, none of these mobility models adopt the distributed autonomous cooperative coordination of UAV swarms because they do not consider aerodynamic constraints and are mostly proposed for MANETs [34], [35]. In UAVSNs, adopting an appropriate mobility model that jointly addresses distributed collaborative coordination, coverage, connectivity, collision avoidance, and link stability is important for obtaining realistic simulation results. Realistic mobility, inspired by behavioral flocking control, can construct a stable UAVSN topology by following three flocking rules [101]. Here, each UAV independently makes decisions by interacting with its neighbors and can precisely define the group motion property with inter-UAV collision avoidance. Through distributed collaboration, the entire UAV swarm can iteratively maintain stable connectivity and coverage by matching distances and velocities with neighboring UAVs. Moreover, the prediction of the future 3D position and velocity of UAVs can help the routing protocol to predict the topology and select a better relay UAV.

In the literature, several behavior-based mobility models have been applied for UAVSNs, such as Boids flocking [35], virtual force [76], virtual spring [95], [102], [103], and APF [98]. Wang *et al.* [3] and Zhao *et al.* [9] utilized the attractive and repulsive virtual force to control UAVSN mobility to maximize coverage to the ground user to serve as ABS while simultaneously maintaining bi-connectivity in the UTU links. Trotta *et al.* [38], [95] designed a virtual spring-based mobility model inspired by Hooke's law to produce attractive and repulsive forces that can maintain a strong UAVSN topology, ensuring both aerial connectivity QoS and coverage to perform surveillance and ABS services. In [24], APF was utilized to control UAV swarm mobility to track a mobile crowd of humans while simultaneously maintaining aerial connectivity. Wang *et al.* [35] developed a Boids-flocking-based mobility model for UAVSNs to adaptively control the UAVSN topology by selecting stable minimal connecting dominating sets, thereby incurring less control overhead.

To prevent swarm partitioning and improve topology management, they added three additional flocking rules: centripetalism, consistency, and synchronization along with separation, cohesion, and alignment. However, these three new rules assume that each UAV knows the position and velocity of the entire swarm. This assumption restricts the distributed execution and requires a higher overhead. Chen *et al.* [104] developed a formation consensus law using two-hop information, which provided a faster formation consensus and better connectivity rate while consuming higher bandwidth.

Therefore, to maintain link stability, avoid swarm partitioning, maintain connectivity with the BS, and ensure uniform node distribution with minimal control overhead, the adaptive adjustment of flocking rules and their weights by using the two-hop neighbor's position and velocity is necessarily required.

### 2.2.2 Existing Routing Protocols

According to previous studies [7]–[9], [11], [13], the routing protocols in UAVSNs are classified as topology-based, position-based, and RL-based.

#### 2.2.2.1 Topology-Based Routing Protocols

Topology-based routing protocols are classified as proactive, reactive, and hybrid routing protocols.

Proactive routing protocols produce a large overhead to maintain the updated routing table for a dynamic topology. Thus, they consume higher bandwidth and energy, which is not suitable for resource constrained UAVSNs. Additionally, they exhibit a slow reaction to a highly dynamic topology, which causes delays, routing loops, link breakages, and blind paths [33]. A loop-free property is essential for dynamic UAVSNs to prevent data packets from being continually routed through similar nodes or paths. Blind path challenges occur in UAVSNs when the neighboring UAVs leave the transmission range of the corresponding source UAV within the intermediate time of topology update because of several reasons such as sudden changes in relative mobility, requirements for energy replenishment, and UAV failure [33]. Additionally, UAVSNs may encounter frequent link breakages if the selected relay UAV leaves the transmission range of the corresponding source UAV during data transmission. Both the blind path and link breakage phenomena produce high retransmissions, delays, and energy consumption.

In [105], [106], the authors studied the optimized link-state routing protocol (OLSR), which encounters higher overheads and routing loops and has a slow reaction in highly dynamic networks. Similarly, in [107], the authors studied the destination sequenced distance vector (DSDV), which consumes a large portion of the network bandwidth and provides a very high overhead owing to periodic updates in UAVSNs. Hong *et al.* [34] first introduced the path stability metric defined by the LD to overcome the link breakages and trade-off between topology prediction accuracy and control overhead by adaptively maintaining the hello interval in OLSR for UAVSNs. However, the uncertainty in UAV communications caused by delay and limited energy are not considered. Moreover, LD is calculated in a two-dimensional scenario. Grag *et al.* [108] proposed mobility and congestion aware OLSR (MCA-OLSR) for UAV networks by leveraging a cross layer design. In MCA-OLSR, each UAV makes a routing decision based on multiple link quality metrics, such as LD, hop count, delay, and number of interfacing links. Owing to priority-aware packet queue management and multi-metric routing decision, MCA-OLSR outperforms existing OLSR protocols.

Reactive routing protocols result in higher latency and delays owing to the on-demand route-discovery process. Additionally, in large-scale UAVSNs, the network overhead increases for reactive routing owing to an increase in the header size of the routing table [109]. In [13], the authors reported that dynamic source routing (DSR) provides a comparatively lower overhead at the cost of delays in route discovery. However, for large-scale UAVSNs, DSR routing encounters an extremely high overhead owing to an increase in the routing discovery table header [109]. Li *et al.* [109] proposed a routing protocol for large-scale UAV communications by leveraging the cross-layer design and introducing a link quality metric in DSR jointly considering link SINR, relative velocity, and queuing delay. To obtain the optimal transmit power, they utilized mean field game theory and optimally allocated the frequency resource by traversing in the available frequency that maximized link SINR. However, reactive DSR produces a large overhead and delay because of the proportional increase in the routing discovery header along with a long path. Similarly, ad-hoc on-demand distance vector (AODV) routing encounters route failures, higher delays, and higher bandwidth consumption in large-scale UAVSNs [107].

Hybrid routing protocols encounter higher computational complexities and overhead owing to the complex clustering, CH selection, and cluster maintenance processes [59].

Wang *et al.* [35] proposed a low complexity two-hop CDS-based topology management for UAVSNs by adopting a Boids flocking-based mobility model. In [87], a CDS-based dynamic topology management was proposed to maximize the throughput in the backbone network by jointly optimizing transmission power, CDS number, and UAV position using page rank and PSO. In [56], joint single-shot localization, clustering, and multi-hop routing techniques were proposed using a bounding box and PSO. However, these types of CDS and cluster-based dynamic topology management require frequent topology construction and management, which triggers a higher overhead because of their frequent cluster head or minimal CDS selection and advertising head declaration, cluster joining, leaving, and merging messages [110]. Moreover, all these methods assume MAC layer resources (i.e., timeslots or frequency) are allocated optimally to prevent interference.

Therefore, all these traditional topology-aware proactive, reactive, and hybrid routing protocols encounter several limitations in highly dynamic UAVSNs owing to the high control overhead and large delay in neighbor and path discovery [111]. All these conventional routing protocols trace the shortest path, which can trigger energy holes and severe network congestion. Additionally, they do not support adaptability to the dynamic topology to discover the efficient routing path autonomously. The limitations of topology-based proactive, reactive, and hybrid routing protocols are summarized in Table 2.1.

Table 2.1 Summary of topology-based routing protocols and their limitations in UAVSNs.

| **Protocol type** | **Limitations to adopt in UAVSNs** |
|---|---|
| Proactive | • Higher control overhead<br>• Higher bandwidth consumption to maintain an updated neighbor table.<br>• Slow reaction to rapid topology changes |
| OLSR [105] | • High control overhead<br>• Routing loop, and link breakage |
| DSDV [107] | • Requires periodic updates, high bandwidth, and control overhead |
| Reactive | • Higher delay in on-demand routing discovery<br>• No link quality assessment |
| DSR [111] | • Produces higher overhead during route discovery in large-scale networks.<br>• Higher delays |
| AODV [107] | • Higher delay, high bandwidth consumption, and link breakage |
| Hybrid [59] | • High computational complexity to construct and maintain the cluster, cluster head, and cluster member. |

#### 2.2.2.2 Position-Based Routing Protocols

In position-based routing, each UAV node utilizes the GPS for localization. In addition, UAVs can use range-free and range-based cooperative localization in a GPS-denied environment. Position-based routing protocols utilize local knowledge, often one- or two-hop information, to make routing decisions. UAVs make forwarding decisions based on their current position, the position of the destination, and the position of their neighbors. In [10], [112], the authors studied several position-based routing protocols in UAVSNs by classifying them into two categories based on path strategy: single-path and multipath strategies. Under the single-path strategy, they reviewed deterministic progress-based, randomized progress-based, and hybrid position-based routing protocols. Deterministic progress-based routing protocols have several relay node-selection strategies, including greedy forwarding, compass forwarding, and most forwarding [10]. Multipath strategies include restricted direction flooding, random directional flooding, and simple flooding of data packets [10].

According to their study, considering the dynamism in network topology in the 3D space, inter-UAV collision, high overhead, and delay, position-based routing protocols are attracting the interest of researchers. However, position-based routing protocols encounter several challenges in UAVSNs, such as maintaining the link quality [113], controlling the hello interval to predict up-to-date topology [34], localization errors, link breakages, blind paths, the presence of routing loops, and energy holes [33]. Additionally, to prolong the lifetime of UAVSNs, it is necessary to achieve a proper load balance in terms of energy and delay while determining the optimal routing path [62]. Tracing the shortest routing path may be initially beneficial, but it cannot be an optimal routing path as it depletes the energy of a few selected UAVs, and the shortest paths can be extremely congested by traffic over time [33]. It also creates energy holes in UAVSNs because selecting the shortest path always drains the energy of a few selected UAVs.

Greedy forwarding cannot ensure optimal performance in terms of energy consumption, delay, and link quality, as it always seeks progress in the transmission distance toward the destination. Additionally, owing to the selection of relay nodes at the edge of the transmission range of the source node, greedy forwarding encounters blind path and link-breakage problems. The compass and most forward techniques have higher possibilities of trapping in routing loops and local minimum [10]. The term local minimum (routing holes)

in position-based routing is defined as the selected relay UAV with no further neighbors to relay toward the target destination node. Flooding techniques in multipath forwarding produce excessive overhead, high MAC layer contention, high bandwidth, and energy consumption. The limitations of only position-based routing protocols for UAVSNs are summarized in Table 2.2.

Table 2.2 Summary of position-based routing protocols and their limitations in UAVSNs.

| Path strategy | Protocol | Limitations to adopt in UAVSNs |
|---|---|---|
| Single-path strategy | Greedy forwarding | • Always seeks progress in transmission distance; thus, it cannot ensure the desired link quality.<br>• Encounters link breakages, blind paths, and routing loops.<br>• Not energy efficient |
| | Compass forwarding | • High possibility to trap in routing loops.<br>• Not energy efficient |
| | Most forwarding | • Trapped in local minimum (no further node within transmission range to forward toward the destination)<br>• Encounters higher link breakages and blind paths.<br>• Not energy efficient |
| Multipath strategy | Restricted directional flooding | • Deterministic decision to select the direction of broadcasting packets.<br>• Broadcast multiple copies of the same packet to the selected direction.<br>• Provides excessive overhead and is not energy efficient |
| | Randomized directional flooding | • Randomized decision to select the flooding direction.<br>• Provides excessive overhead and high contention.<br>• Not energy efficient |
| | Simple Flooding | • Excessive overhead and high contention<br>• Not energy efficient |

### 2.2.2.3 Reinforcement Learning-Based Routing Protocols

In UAVSNs, link quality depends on several parameters, such as inter-UAV distance, node density, SINR, delay, relative mobility, and RE of relaying UAVs. The optimal node density and link SINR can be achieved by jointly optimizing the UAV mobility (position, velocity, and acceleration) and transmitting power according to the inter-UAV distance by

adopting a suitable TCA [34], [64]. The link delay includes MAC-layer channel access, queuing, propagation, and transmission delays. The optimal resource allocation in resource-constrained UAVSNs, such as physical-layer UAV transmission power, MAC-layer timeslots, or frequency resources, can significantly improve the SINR level in aerial links. Thus, this sequentially improves the network-layer performance (relay selection) as they are highly coupled.

Owing to the above advantages, researchers have jointly considered the MAC layer delay, link SINR, relative mobility, position progress to the destination, and RE level of neighboring UAVs, to design a multi-objective reward function in reinforcement learning (RL)-based algorithms [7], [8]. RL is an area of machine learning concerned with how intelligent agents ought to take an action from a specific state by interacting with a dynamic environment to maximize long term cumulative reward. Through the iterative state transitions, an agent learns how to choose an optimal action. Thus, RL-based action can be formulated as a Markov decision process (MDP) tuple consisting of state, action, and reward. The state represents the consequences that an agent faces in a dynamic environment by taking actions according to the learning policy. Through sequential action and utilizing previous experience, RL agents can make wiser decisions to reach a common objective. Owing to the advantages of less modeling difficulty, RL method can be used efficiently to solve complex multi-objective optimization problems by designing multi-objective reward function and treating optimization constraints as the penalty terms. Thus, RL-method does not require the convexity requirement and adaptively learn optimal policy by interacting with dynamic environment without requiring any central controller.

In UAVSNs, RL is applied in many scenarios such as trajectory control [114], [115], channel modeling [116], and resource allocation [37]. Recently, RL has been widely used in UAVSNs to design the smooth collaborative trajectory planning for UAV swarms with collision avoidance, and routing protocol design [117]. The combination of RL and deep learning also known as deep reinforcement learning (DRL) is getting attention to solve complex optimization problems due to the advantages of extracting important features, dealing with large state-action dimensionality, and utilizing recent historical information in time-varying dynamic topology. The recurrent neural network (RNN) such as long short-term memory (LSTM) and gated recurrent unit (GRU) [118] can efficiently track the temporal correlation in the sequential time series data (i.e., UAV trajectory, relative mobility,

and channel state), which is very effective to deal time-varying dynamic topology of UAVSNs.

According to our earlier discussion in Section 2.1, we can say that the TCA iteratively updates the mobility of each UAV within a swarm by using the mobility information of its neighbors. Additionally, the output of the TCA decides the topology of the UAV swarm by predicting the present and future mobility information for each UAV (acceleration, velocity, position, and flying direction) [21]. Thus, we can say that relative trajectory knowledge given by the TCA and link stability is highly coupled [29]. It can ensure stable connectivity between UAVs during flocking. The TCA updates the mobility information for each UAV in the next timeslot based on the most recent historical mobility information in the current timeslot, which indicates the similarity with the Markov property. This is because the Markov property states that the next states of the process depend only on the current state of the process. As a result, RL/DRL-based MDP formulation can be adopted to make routing decisions to obtain the most stable path [119].

Owing to this relationship, researchers have used the RL technique to select the optimal relay nodes for forwarding data in UAVSNs by designing a multi-objective reward function. Because the reward function reinforces the action policy of an RL agent and accelerates the algorithm convergence for optimal decision making, a good reward function considering multiple objectives (i.e., delay, SINR, relay node energy, and distance progress toward the destination node) gives better routing performance. Consequently, this joint consideration of multiple objectives significantly improves the packet delivery ratio (PDR), throughput, end-to-end delay, and balances the energy consumption in UAVSNs. The important features supported by RL-based routing protocols compared to the existing topology-based and position-based routing protocols in UAVSNs are summarized in Table 2.3.

Table 2.3 Important features supported by RL-based routing protocols in UAVSNs.

| Routing protocol type | Key features in routing | | | | | |
|---|---|---|---|---|---|---|
| | Link quality assessment | Routing loops avoidance | Routing holes avoidance | Energy holes avoidance | Delay optimization | Mobility prediction and link breakage avoidance |
| Topology and position-based | × | × | × | × | × | × |
| RL-based | √ | √ | √ | √ | √ | √ |

Note: "×": The corresponding feature is not supported; "√": The corresponding feature is supported.

The widely used RL/DRL methods are Q-learning (QL) [120], deep Q-network (DQN) [121], [122], and deep deterministic policy gradient (DDPG) [37], [115], [123]. Among them QL and DQN are value-based RL method and can handle only small-scale discrete action space. In contrast, DDPG is an RL algorithm that combines ideas from DQN and policy gradient methods to enable learning of continuous control problem in high-dimensional state and action spaces. DDPG is an actor-critic algorithm, meaning it maintains both an actor network to approximate the deterministic policy and a critic network to approximate the action-value function.

QL is a model-free value-based off-policy RL approach, which can obtain an instant optimal policy based on historic experiences even without prior information of the environment or even without the intervention of any central controller [124]. Here, each agent makes an optimal decision based on its neighbor state information, which can be treated as partial MDP. Considering the high mobility, constraint energy, and memory resources of UAVs, the QL method is more suitable for UAVSNs routing decision making than DRL because it is computationally more expensive and requires a large memory to store training samples and a history of action–reward pairs. In UAVSNs, QL is more suitable to make online decisions by adaptively addressing the trade-off between exploration and exploitation. Nevertheless, in large scale UAVSNs specially in cross-layer design QL encounters complexity to deal with large state-action dimensionality.

Considerable research has been conducted to improve the performance of position-based forwarding by integrating it with QL. Jung *et al.* [113] proposed a QL-based geographic routing (QGeO) for FANETs to overcome the limitations of position-based

forwarding. Rather than solely seeking progress in the transmission distance to forward data toward the destination, QGeO introduces the concept of packet travel speed (PTS), which considers the distance progress toward the destination, localization error, link error, and link delay to select the relay UAV. However, in QGeO, the UAV energy is not considered, and the discount factor is adjusted only for two different inter-UAV distances. In [113], [125], the authors showed that accounting for the mobility dynamism in FANET position-based routing with QL reduces broadcast storms and minimizes the delay in communication. Sliwa *et al.* [126] proposed a QL-based routing protocol for a UAV-aided network in which each UAV adaptively updates the Q-values by exchanging hello packets. Additionally, to cope with the dynamic FANET topology, the QL model adaptively adjusts the discount factor based on the LD and cohesion value [126]. Owing to the utilization of the predictive LD according to the relative UAV trajectory knowledge, a high PDR and lower latency were achieved. However, UAV energy was not considered in their design.

Liu *et al.* [62] proposed a QL-based multi-objective optimization (QMR) routing protocol for FANETs to minimize the delay and UAV energy. To address the FANETs dynamism, the QL module adaptively adjusts the value of the learning rate and discount factor according to the delay and similarity in the neighbor set. Furthermore, to address the exploration-exploitation trade-off in QL, the QMR routing protocol selects a relay UAV that provides a higher PTS value. Arafat and Moh [33] proposed a QL-based topology-aware routing (QTAR) in which each UAV maintains a two-hop neighbor list by imposing constraints on the PTS. Using the two-hop neighbor mobility information, each UAV can extend its local view and make better optimal decisions. However, the connectivity control and link SINR conditions were not considered for both the QMR and QTAR. Similarly, Luis *et. al.* [127] proposed an improved QL-based routing protocol for FANETs by integrating QMR and Q-noise+ to minimize the delay and jitter. Based on the availability of candidate neighbor UAVs, their algorithm selects the neighbor UAV according to the maximum PTS or random exploration by utilizing the $\in$-greedy method. In [128], the queuing theory was applied to perform neighbor discovery and adaptively adjust the hello interval to adopt the dynamic topology. Subsequently, they utilized QL to trace the optimal path for UAV communication in terms of the minimal delay and communication energy consumption. They predicted the mobility of neighboring UAVs by precisely calculating the LD using a simple Kalman filter under RWP mobility. However, the simple Kalman filter-based

neighbor UAV mobility prediction only ensures accuracy in linear Gaussian motion, whereas realistic UAV motion is mostly non-linear [129].

Without monitoring the connectivity for a particular time interval to control the relative mobility of the UAVs, it is very challenging to satisfy the imposed PTS constraint with neighboring UAVs. Both QMR and QTAR may face challenges in maintaining an adequate LD greater than the PTT required for successful packet delivery. This problem becomes more complex if UAVSN needs to maintain coverage efficiency by tracking mobile ground targets (MGTs) because it brings more instability to the FANET topology. Thus, to ensure both mission and communication performance in a real UAVSN application scenario, joint topology control and routing are required.

Additionally, we notice that neither QMR nor QTAR consider the path stability metric in the reward function. Because the reward function reinforces the QL agent's action, a good reward function can help QL agents to achieve better decision making. In addition, neighbor selection only based on the PTS metric cannot ensure better path stability because PTS only considers the inter-UAV distance. Thus, to cope with the high mobility of UAVs in 3D space, the relay UAV selection (state exploration) by predicting the mobility of UAVs given by 3D LD provides better stability in routing. In [130], a piecewise linear 2D mobility model was utilized to control connectivity among UAVs. Based on the mobility, the model controls the value of the temperature parameter in the simulated annealing optimization, which determines the exploration rate in QL to make energy-efficient routing decisions. However, in their QL model, the UAVs randomly selected a relay UAV [130]. Exploration based on random actions produces higher retransmissions and detours. According to the above comparative review of the existing protocols, the comparison of the QL-based routing protocols is summarized in Table 2.4.

Table 2.4 Comparison of QL-based routing protocols in UAVSNs.

| Reference | Optimization parameters | Reward parameters | Neighbor information | Path stability consideration based on link duration and PTT | Exploration-exploitation strategy | Adaptive hello interval | Adaptive learning rate and discount factor | Mobility model |
|---|---|---|---|---|---|---|---|---|
| [113] | Delay | PTS | 1-hop | × | × | × | ∂ | Gaussian Markov |
| [126] | Delay | – | 1-hop | × | × | × | ∂ | Random waypoint |
| [62] | Delay and energy | Delay and RE | 1-hop | × | One-hop PTS | × | √ | Random waypoint |
| [33] | Delay and energy | PTS, delay, and RE | 2-hop | × | Two-hop PTS | √ | √ | Gaussian Markov |
| [127] | Delay, and Jitter | Positive number | 1-hop | × | One-hop PTS and ∈-greedy | × | × | Random waypoint |
| [130] | Delay and energy | – | 1-hop | × | Simulated annealing | × | √ | 2D piecewise linear mobility |
| [128] | Delay, and energy | Link, neighbor, and distance | 1-hop | × | One-hop LD | × | √ | Random waypoint |

Note: "–": Not mentioned; "√": Supported; "×": Not supported; "∂": Partially considered.

Zang *et al.* [131] proposed a centralized data-driven adaptive routing protocol for UAV communications, where a weighted link quality metric is jointly defined by considering the inter-UAV distance, packet arrival rate, queue backlog length, and the number of hops. To avoid network congestion, they predicted the packet arrival at each UAV by leveraging the LSTM. Nevertheless, mobility prediction solely based on the inter-UAV distance in a centralized server may not provide an optimal solution. In [132], an extended MDP formulation was proposed by considering the state of both the current node and its one-hop neighbor, to select a relay UAV for minimizing delay. Owing to the large state space and discrete next-hop selection action space, they adopted DQN to make a routing decision. Additionally, to mitigate the online training problem in QL, they trained the DQN model offline using the concept of a generative adversarial network. In [122], adaptive hello interval adjustment techniques were proposed using DQN to improve the link reliability in dynamic UAVSNs. However, all of the above-mentioned QL and DQN-based algorithms ignore UAV trajectory optimization and MAC layer frequency or timeslot allocation, which is a critical requirement to improve the routing protocol performance.

Ding *et al.* [133] utilized a multi-agent deep Q-mixing network to solve a multi-hop packet routing problem from UAV to BS by jointly considering the trajectory design, frequency resource allocation, and next-hop selection. Their objective was to minimize transmission delays. However, the Q-mixing network computes the global Q-value for the actions taken without properly considering the observation collected from the neighboring UAVs [134]. Moreover, they discretized the UAV movement to simplify the action space, whereas the realistic action space for the UAV trajectory should be continuous [29]. Qiu *et al.* [135] applied MA-DDPG-LSTM to make routing decisions in UAVSNs by jointly considering link SINR, LD, and queuing delay, which involves adopting centralized training and distributed execution.

To cope with the dynamic topology, they considered the LSTM-based actor and critic network. However, trajectory control according to the physical layer transmission range was not considered, and they assumed that frequency resources are allocated optimally in the MAC layer. In a multi-agent scenario, critic network solely based on LSTM cannot provide adaptive attention to the neighbors' policy and LSTM does not support parallelization in the critic value function computation, which can trigger slow convergence and unstable training. Moreover, in the fully centralized training, the state-action dimensionality in centralized

critic network becomes excessively large with increasing number of UAVs, which can cause higher computational complexity and less scalability. DQN, double DQN (DDQN), and DDPG encounters to behave optimally during online execution after the offline training. Thus, further research is required to overcome above challenges.

All of the above-mentioned routing protocols consider generic mobility models, such as RWP and Gauss-Markov. Such mobility models cannot adopt the properties of UAVSNs and their aerodynamics [40]. Moreover, all RL/DRL-based algorithms control the UAV trajectory by defining UAV movements in the discrete action space (left, right, forward, backward, and hover) without any collaboration with the neighboring UAVs [133], [136]. Such discrete trajectory control cannot provide realistic trajectory because of the reduced degree of freedom in movement. Additionally, it requires a long time to train [136]. In UAVSNs, the swarming behavior-based mobility models can autonomously maintain optimal node density, coverage, connectivity, stable LD, and inter-UAV collision avoidance in both U2U and U2BS links [120].

Therefore, it is necessary to design swarming behavior coupling adaptive distributed multi-agent DDPG with two-hop neighbor information to control UAV trajectory in continuous action space, allocate frequency resources, and select relay UAVs. To generate realistic trajectory of UAVs in a distributed manner a behavior-based motion model needed to design under the sensor noise, wind disturbance, and communication uncertainties. Subsequently, the key observed state of the time-varying topology, such as the motion rules generated by the relative distance and velocity, link SINR, frequency state, queue backlog size, and LD up to two-hop neighbors are fed into the actor LSTM-based state representation layer (SRL). LSTM-based SRL forwards a better state to the actor fully connected layer (FCL) by mining temporal correlations between the current state and a finite amount of the previous historical state. Moreover, multi-head attentional critic networks can be utilized to generate action value function and adaptively adjust the actor policy by paying attention to its neighbors in a sorted weighted manner. Here, each agent state-action can be treated as a query, and the neighbor's state-action spaces can be considered as both key and value. The normalized attention weights given by the scaled dot product guides each agent to which neighbor it should pay more attention to produce a more precise Q-value.

## 2.3 Issues and Challenges of Routing in UAVSNs

In this section, the key open issues and research challenges to design RL-based routing algorithms for UAVSN are summarized.

### 2.3.1 Joint TCA and Routing

In FANETs, the relative mobility and the path stability are highly coupled with each other. TCA can ensure aerial coverage and path stability by controlling the relative mobility of UAVs, while performing the collaborative mission. TCA also minimizes the number of transmissions by performing the data aggregation at each elected CH and offers a stable topology to the inter-cluster multi-hop routing protocol. Owing to the high mobility of UAVs, it is very difficult to maintain communication from one CH to another. Thus, an alternative approach is required to perform inter-cluster routing so that we can deal with multiple issues such as traffic congestion, energy holes, routing holes, loops, and link quality assessment in inter-cluster routing. QL-based position-aware routing is suitable to perform multi-objective optimization in FANETs, which can significantly improve the inter-cluster routing performance.

As a result, the joint consideration of TCA and QL-based routing protocol enhances the performance of FANETs, because TCAs control the mobility of UAVs by controlling the relative distance, relative velocity, and direction according to the neighbor UAVs' movements to ensure sufficient LD. Additionally, TCA maintains a strong neighborhood with the neighbor UAVs, and offers a relatively stable state of the UAVs to the routing protocol at each time slot of the mobility update while incurring minimum overhead.

### 2.3.2 Realistic Mobility Model

According to our survey and reviewed protocol, all routing protocols except PARRoT [126] consider generic mobility models, such as RWP and Gaussian Markov mobility models. However, according to the discussion in Section 2.1 and 2.3.2, the mobility models in UAVSNs should be application-dependent and should adopt the behavior of SI to achieve realistic results in the simulation environment. Mobility control algorithms, such as Boids flocking [35], virtual force [50], [57], [76], virtual spring [38], [95], and APF [24], [64], [98] produce a realistic mobility model for a UAV swarm in a software simulation environment considering the type of mission. Thus, designing and evaluating routing protocols that consider a realistic mobility model can be an interesting research concept.

### 2.3.3 Multi Objective Reward Function Design

Because the reward function reinforces the algorithm convergence, designing a good reward function is very important to improve the routing performance. The QMR [62] and QTAR [33] jointly consider the link delay and RE level of UAVs in their multi-objective reward function and achieved significant performance improvement for PDR, end-to-end delay, and balance in energy consumption. However, designing the reward function considering path stability, delay, and UAV residual energy may provide more better routing performance in FANETs.

### 2.3.4 Trade-off Between Exploration and Exploitation

Exploration is an attempt to discover a new state in the search space that may provide a better reward compared with the existing experience of an RL agent. Exploitation refers to performing the best action according to existing experience. Exploration aids in determining the global optimal solution. However, during exploration, the action performed might be good or bad because excessive exploration may produce unnecessary detours, retransmissions in UAVSNs, and delay the convergence of the algorithm. Therefore, in UAVSNs routing decision making using RL, a strategy is required to balance the trade-off between exploration and exploitation to attain the global optima.

Some RL algorithms consider $\in$-greedy [127] and upper confidence bound (UCB) [137] strategies to control the exploration rate. However, in the $\in$-greedy strategy, the exploration rate depends on the parameter $\in$, which is frequently approximately 10 %, resulting in a very low exploration rate. The UCB strategy can control the exploration rate by jointly considering the sum of the average cumulative reward and number of times a specific action is selected within a specific time. In [62], the authors reported that the exploration rate should be controlled according to the network condition and degree of mobility changes in UAVSNs, instead of exploration based on time. This is logical because, when the relative neighbor state is stable, UAVs can exploit according to the existing Q-value. Otherwise, when the relative neighbor state is not stable, UAVs can perform exploration according to the predicted link duration with the neighbor links to achieve a more stable routing path.

### 2.3.5 Precise Calculation of UAV Energy Consumption

In UAVSNs, the energy consumption cost of UAVs depends on the power consumption for propulsion and communication to transmit and receive data with neighboring UAVs,

GUs, and BS [138]. However, the propulsion power of UAVs consumes significantly more energy than the communication energy cost [35]. All routing protocols only consider the communication energy when calculating the energy cost. For a realistic performance, the energy cost should be obtained by considering both propulsion and communication power. An appropriate energy consumption cost defines the presence of UAVs in the aerial network and defines the accurate node density, which is directly related to communication performance. The propulsion power is proportional to the UAV trajectory. Thus, during a collaborative mission, the trajectory should be optimized and smooth, and all UAVs should travel approximately the same distance to execute the mission [29]. Additionally, the propulsion energy cost depends on the type of UAV deployed to execute the mission. A recent survey discussed the propulsion energy model according to the type of UAV [139].

### 2.3.6 Cross Layer Design

In UAVSNs, the link delay, SINR level, link reliability, UAV RE level, and relative mobility prediction defined by 3D LD are the key factors in defining link quality. The trajectory control according to the physical layer transmission power and optimal resource allocation (i.e., frequency or timeslots, and MAC queue management) in the MAC layer, control the link SINR, data rate, and network congestion. Joint consideration of trajectory control, resource allocation, and relay selection according to above mentioned multiple link quality parameters can significantly improve the performance of the routing protocol performance because they are highly coupled. Thus, designing such cross-layer routing protocol in UAVSNs can be an interesting research direction.

### 2.3.7 Neural Network Architecture

In conventional DDPG, both actor, critic, and their target networks are constructed solely depending on FCL. FCL cannot extract the important features based on temporal continuity of sequential time series data of dynamic time-varying topology, which may be useful for obtaining better policy and value function approximation. Additionally, in multi-agent inter-active environment, each agent needs to adjust its policy according to the policy changes of the neighboring agent by adaptively paying attention to the nearby agent according to their degree of influence to avoid environmental non-stationarity and achieve faster convergence. Moreover, to support scalability, and reduce computational complexity in large scale UAVSN, critic network should utilize state-action features by using multi-head attention network only considering the nearby agents.

### 2.3.8 Model Training and Adaptive Learning

The training method of multi-agent DRL considers fully centralized, centralized training and decentralized execution, and fully cooperative training [140]. Considering the distributed execution of UAVSN and huge state-action dimensionality in large scale UAVSN fully cooperative training-based DRL algorithm design is highly required. It is because in fully centralized training each agent needs to transmit its observation-action to the central critic, which may cause higher computational complexity, bandwidth consumption, less scalability, and dealing with outdated mobility state of UAVs. Additionally, considering the fixed communication range of UAVs, some UAVs may stay very far away, and their observation-action has very less impact on current UAV's reward. Thus, training each UAV's actor network based on global Q-value generated by centralized critic without paying adaptive attention to the neighboring agent may not generate optimal policy. Nevertheless, avoiding the local optimal decision is a challenging issue in fully cooperative training, which requires research attention.

## 2.4 Comparison Between Proposed Routing Protocols

Based on the identified open issues and research challenges discussed in Section 2.3, we proposed three state-of-the-art routing protocols for UAVSNs to meet different objective and target application scenarios. The comparison of contributions between proposed routing protocols are summarized in Table 2.5.

Table 2.5 Comparison of contributions between proposed routing protocols.

| Parameter | Proposed protocol | | |
|---|---|---|---|
| | JTCR | QRIFC | JTFR |
| Key contribution | • Proposed a hierarchical routing protocol to perform crowd surveillance consisting of three modules.<br>• We consider practical mission driven mobility models inspired by virtual force to construct hierarchical UAVSN topology. | • Proposed an adaptive 3D mobility model inspired by behavior-based flocking model.<br>• Designed a new reward function for QL-based routing protocol using maximum-minimum LD up to two-hop neighbor information. | • Designed a fully cross-layer routing protocol using multi-agent DRL algorithm.<br>• A link utility maximization problem is designed under several practical constraints in UAVSN environment.<br>• We modified the actor and critic neural network architecture to adopt the dynamic time varying topology. |

| | | | |
|---|---|---|---|
| Topology type | • Leader-follower | • Flat AMN | • Flat AMN |
| Advantages | • Owing to two phase topology control and hierarchical routing, it provides less MAC layer contention for both intra-cluster and inter-cluster routing.<br>• Balances the requirement between mission performance and communication performance. | • Faster swarm cohesion and topology formation using two-hop neighbor information.<br>• The proposed mobility model maintains both U2U and U2BS links.<br>• Local optima avoidance using two-hop neighbor information.<br>• Proposed a new exploration and exploitation strategy for routing decision making to obtain better average reward. It outperforms the existing method, and benchmark method. | • MDP is formulated considering multiple key features in multi-agent environment, which enhances decision making.<br>• UAVs can make decisions using historical information of time-varying topology.<br>• Critic network with attention mechanism helps to focus on relevant information with less computational complexity and overcomes the environmental non-stationarity.<br>• Distributed model training mechanism coupling with swarming behavior-based motion model in the presence of Gaussian noise helps to adopt high fidelity behavior and optimal policy for online execution. |
| Limitations | • May trap in local minimum as only utilize one-hop neighbor information.<br>• Required higher control overhead to maintain two-phase topology control. | • Cannot utilize the historical information of time-varying topology.<br>• Support only limited state-action features in MDP formulation as QL suffers curse of dimensionality. | • Although it has less computational complexity compared to fully centralized or centralized training and distributed execution, it has higher computational complexity to train the model. |
| Target application | • Crowd surveillance, aerial base station deployment, and proving edge computing service to ground devices. | • 3D realistic mobility model for routing protocol simulation in UAVSN environment. | • Post-disaster mapping and surveillance |

# 3. Joint Topology Control and Routing

## 3.1 Introduction

Recently, the rapid development of UAV technology has made UAV swarms commercially viable. Advanced sensors [141], vision-based target localization [142], [143], battery improvements, ultra-wideband indoor and outdoor localization [144], GPS-based localization [56], obstacle avoidance techniques [6], integration of various artificial intelligences [59], and machine-learning techniques [145] are used together to provide autonomous operation of a UAV swarm. Low-altitude UAVs and drones have shown considerable potential to mitigate pandemic disease outbreaks, especially during COVID-19. This is accomplished via large-scale crowd surveillance and public announcements to enforce social distancing. The UAVs have been used to spray disinfectants into contaminated areas and deliver emergency medical supplies. UAVs equipped with infrared cameras for large-scale temperature measurements in crowds have also been deployed [146].

It has been observed that LiDAR sensors with $360^{\circ}$ field of view equipped on a UAV can track MGTs, i.e., mobile crowd of human, while preserving individual privacy and monitoring social distancing. Moreover, LiDAR-based 3D UAV mapping is functional and provides high accuracy during harsh weather [147]. Today, cities are dense population centers driven by economic motives, resource availability, and social standards. As a result, next generation video surveillance systems are expected to incorporate UAV swarms [24], [148].

The deployment of UAV swarms for persistent crowd surveillance poses several research challenges. To localize the ground targets utilizing the onboard vision sensors of each UAV is a challenge [149]. Another challenge is the topology control of a UAV swarm, which adjusts UAV positions in 3D space periodically according to each UAV's transmission range not only to maximize coverage but also to maintain the high connectivity in UAV-to-UAV (U2U) links with the desired SINR [23], [24], [50], [84]. Efficient energy management is also a challenge. It can be achieved by an energy-efficient routing protocol that delivers the sensed data such as the captured video of MGTs, 3D LiDAR mapping, and thermal images to the BS or the mobile edge computing server with minimum delay and high PDR in a real-time basis. Though the energy-efficient routing prolongs the lifetime of FANETs significantly, the energy replenishment technique is required to perform persistent surveillance [38].

In this study, we focus on joint topology control and routing to ensure mission performance while improving communication performance. Energy efficiency is also considered in routing. Similar to [149], we consider the vision-based localization of MGTs

within a mission area. Some state-of-the-art object detection and categorization methods can be used to visually distinguish MGTs from other background objects [142]. UAV swarm deployment for optimal MGT coverage over mission areas is very challenging owing to several constraints such as the limited number of available UAVs, energy limitations, limited communication ranges, desirable SINRs of U2U links, MGT mobility, trade-off between coverage efficacy, and aerial connectivity [100]. Surveillance using a UAV swarm requires maximizing the coverage of MGTs while transmitting the sensed data to BSs, which demands high QoS in connectivity with acceptable delays [99]. To meet the mission performance, UAVs should be placed as wide as possible, which affects the QoS in the U2U links. To preserve strong connectivity, UAVs should not frequently fly away from each other's communication range by maintaining the three principles of flocking: cohesion, separation, and alignment [98]. Owing to the relatively high cost of UAVs, it is infeasible to deploy enough UAVs to cover a large mission area. Therefore, UAVs require to move to track maximum MGTs as dynamic coverage. The static coverage gives a fixed coverage density, but it is not appropriate to sense a particular area for most of the time while leaving the remainder, and the density of MGTs may not be equal [29].

In [13], [111], the authors studied the topology-aware proactive, reactive, and hybrid routing protocols that may produce not only high control overhead and long delay but also a routing loop. It is because they have a slow reaction to the highly dynamic topology. Finding the shortest path may be good for the fastest delivery during the initial phases, but it cannot be an optimal routing path because it may trigger energy holes as it drains the energy of a few selected UAVs, and the shortest paths can be extremely congested [62]. In contrast, by considering the 3D dynamic time-varying topology, higher control overhead, and the possibility of inter-UAV collision, the position-based routing protocols are expected to be a valuable option for FANETs [10], [112]. However, because they only look for progress in transmission distance to reach the desired destination without both predicting the relative mobility and considering the link quality (LQ), they face higher link breakages in highly mobile FANETs [150]. They also encounter some other challenges with FANETs, including the hello interval for up-to-date topology prediction, the presence of routing holes, routing loops, and balanced energy consumption [33].

Thus, to deal with multiple problems, an intelligent algorithm is required to perform multi-hop routing in FANETs. Recently, RL is widely exploited to enhance the communication performance in FANETs, by predicting channel conditions and by jointly optimizing the UAV trajectory and communication performance [138]. By iteratively taking actions in a dynamic environment and exploiting previous experiences, RL agents can make wiser decisions to maximize the reward. QL is a value-based model-free off-policy RL algorithm, which is one of the simplest and most practiced RL algorithms [150]. Owing to the advantages of multi-objective optimization, the position-based routing protocol

incorporating QL is a lucrative solution for resource-constrained FANETs [62]. QL can be used to avoid the routing holes and loops by assigning minimum rewards. Nevertheless, the QL process can result in higher retransmissions that drain UAV energy. This is mainly due to the insufficient training samples, the imbalance between exploration and exploitation strategy, and random actions lacking proper guidance. In this paper, to overcome the above limitations, we propose an integrated scheme of two-phase topology control and position-based Q-routing with a new state exploration strategy, which is named as joint topology control and routing (JTCR).

To maintain a stable FANET topology, strong neighboring relationships should be maintained by controlling the relative distance, resultant direction, and velocity so that the link longevity (LG) among neighboring UAVs is maximized. To accomplish this, we propose JTCR to jointly investigate topology control and routing in FANETs. The major contributions of this study are summarized as follows:

- Virtual force-based mobility control (VFMC): The VFMC is the first module of JTCR and utilizes two different virtual forces: the MGT discovery force (MGT-DF) to maximize coverage toward MGTs and the adaptively weighted topology formation force (TFF) to ensure the desired SINR level in U2U links under the minimum separating distance. By leveraging these virtual forces at each timeslot, each UAV can estimate a net virtual force (NVF) to determine the optimal mobility information. The VFMC optimizes the hello interval to obtain topological changes faster and minimizes the control overhead according to the minimum link longevity found within the one-hop vicinity of each UAV.

- Energy-efficient mobility-aware fuzzy clustering (EMFC): The EMFC is the second module of JTCR and utilizes the mobility information provided by VFMC to divide the topology into multiple stable clusters for data aggregation. The EMFC clustering concept reduces the number of agents, as data packets are aggregated at each selected cluster head (CH). This helps the next Q-routing effort to relay data traffic to the BS with less transmission and less MAC layer contention, and it gives a high PDR compared with the individual UAV transmissions to the BS.

- Topology aware Q-routing (TAQR): TAQR is the third module of JTCR and achieves the multi-objective optimization for inter-cluster position-based multi-hop routing. The TAQR offers the source CH UAVs to transmit the aggregated data packets (ADPs) to the BS by selecting an optimal routing path that avoids congestion, energy holes, loops, and link breakages in FANETs.

QL requires exploration to converge to an optimal route. During exploration, uncertainties may produce unnecessary detours, resulting in a larger number of retransmissions and more energy consumption. To overcome this problem, we design a new state exploration and exploitation strategy for FANETs based on the relationship between the average neighbor intimacy (ANI), packet travel time (PTT), packet travel speed (PTS), and the link longevity. This strategy meets the trade-off between exploration and exploitation to avoid local optima. It also helps to avoid unnecessary random exploration and detours, which accelerates the convergence and reduces the number of retransmissions in FANETs.

We design a new multi-objective reward function based on the one-hop delay, path stability defined by neighbor intimacy (NI), and the RE of UAVs. Our designed reward function achieves a better average reward compared to the existing routing protocol. The TAQR can avoid routing holes, routing loops, failure-state, and link breakages by introducing a penalty mechanism and topology adjustment triggering method.

## 3.2 System Model

We consider a set of quadrotor UAVs $U = \{u_i\}_{i=1}^{|U|}$, equipped with sensors (e.g., LiDAR, thermal and normal cameras, GPS, IMU, and wireless communication interfaces) deployed in a 3D mission area, $D$ $(length = x, width = y, height = h)$. The UAVs perform surveillance operations that track a randomly distributed set of MGTs, $M = \{m_k\}_{k=1}^{|M|}$, as shown in Figure 3.1. Here, $|.|$ represents set cardinality. The entire surveillance operation time $T$ is divided into equal $n$ discrete timeslots $T = \{t_n\}_{n=1}^{|T|}$, where $\Delta t$ is considered as the length of each timeslot $t_n$. At each $t_n$, each UAV periodically senses the mission area $D$, and transmits data (e.g., videos and 3D LiDAR mapping of MGTs) to the associated leader CH UAV. The data are then transmitted to the BS for further processing with the help of an edge server . We assume that the duration $\Delta t$ is sufficiently small, and the location of UAVs are fixed within this interval. At each $t_n$, the UAVs can leave the aerial network for energy replenishment via a charging scheduling algorithm and thereafter rejoin in the aerial network [38]

Each UAV $u_i$, can localize itself in the global frame at each time instant $t$, by utilizing its GPS, whose coordinates are $\boldsymbol{p}_{u_i}(t) = (x_i, y_i, h_i)$ and has a fixed communication range $R$ with transmission power $P_{tx}$. UAVs utilize their on-board sensors to localize MGT positions or dense areas (DAs) of MGTs within their disk-size sensor-coverage radius $R_C$, at each $t_n$. All UAVs are aware of the location of the BS and dimension $D$. Our proposed JTCR, which is used to perform crowd surveillance in terms of channel and delay models, topology construction model, and routing model is discussed below.

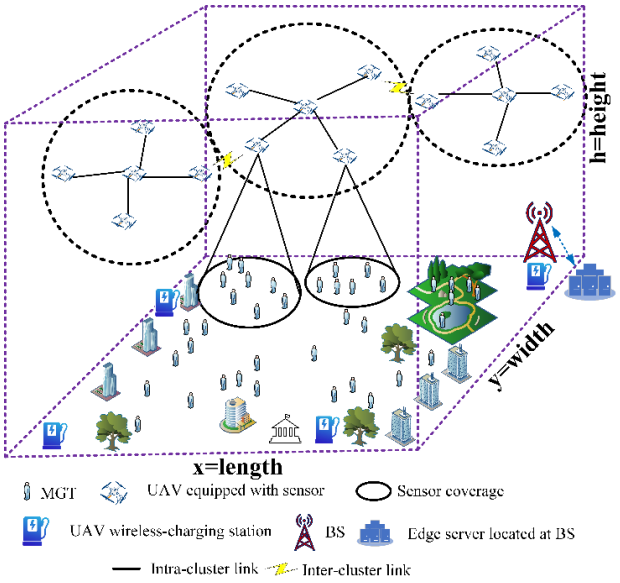



Figure 3.1 UAV swarm network for persistent crowd surveillance.

The frequently used notations in this study are summarized in Table 3.1.

Table 3.1 Notations used in this study (JTCR).

| Notation | Description |
|---|---|
| $D$ | Three-dimensional (3D) mission area |
| $U = \{u_i\}_{i=1}^{\|U\|}$ | Set of $u_i$UAVs |
| $M = \{m_k\}_{k=1}^{\|M\|}$ | Set of MGTs |
| $T = \{t_n\}_{n=1}^{\|T\|}$ | Entire surveillance time |
| $\Delta t$ | Length of each timeslot $t_n$ |
| $G(t_n)$ | FANET topology graph at each $t_n$ |
| $\boldsymbol{p}_{u_i}$ | Position vector of each UAV $u_i$ |
| $\boldsymbol{v}_{u_i}$ | Velocity vector of each UAV $u_i$ |
| $\boldsymbol{a}_{u_i}$ | Acceleration vector of each UAV $u_i$ |
| $R_{sd}$ | Separating distance range |
| $R$ | UAV communication range |
| $N(u_i)$ | One-hop neighbor of UAV $u_i$ |
| $\boldsymbol{d}_{u_{ij}}$ | Distance between two neighboring UAVs |
| $R_C$ | Sensor coverage to the ground terminal |
| $\boldsymbol{F}_{DF}^{u_i}$ | MGT discovery force (MGT-DF) vector |
| $\boldsymbol{F}_{TFF}^{u_i}$ | Topology formation force (TFF) vector |

| | |
|---|---|
| $\boldsymbol{F}_{NVF}^{u_i}$ | Net virtual force (NVF) vector |
| $LG_{u_{ij}}$ | Link longevity for link $u_{ij}$ |
| $HI$ | Hello interval for each UAV $u_i$ |
| $LQ_{u_{ij}}$ | Link quality for link $u_{ij}$ |
| $NI_{u_{ij}}$ | Neighbor intimacy |
| $ND_{u_i}$ | Node degree of UAV $u_i$ |
| $RE_{u_i}$ | Residual energy of UAV $u_i$ |
| $N_{\max}$ | Maximum cluster size |
| $R_N$ | IF-THEN Fuzzy rules |
| $delay_{u_{ij}}$ | One-hop delay on link $u_{ij}$ |
| $PTT_{u_{ij}}$ | Packet travel time for UAV $u_i$ |
| $PTS_{u_{ij}}$ | Packet travel speed for UAV $u_i$ |
| $r_{u_{ij}}$ | Multi objective reward function |
| $\alpha_{u_{ij}}$ | Learning rate in Q-learning |
| $\lambda_{u_{ij}}$ | Discount factor in Q-learning |
| $ANI_{u_i}$ | Average neighbor intimacy for UAV $u_i$ |
| $PFC_{hop1}$ | One-hop potential forwarding candidate set |

### 3.2.1 Channel and Delay Model

Owing to the open 3D spaces, communications among high-altitude UAVs (U2U links) and UAV-to-BS links are dominated by the LoS [92]. At time instant $t$, the channel power gain $\mathcal{G}_{ij}(t)$ between a source UAV $u_i$, and a receiver UAV $u_j$, or a BS in the free-space path loss model can be expressed as follows [151]:

$$\mathcal{G}_{ij}(t) = \rho_o d_{u_{ij}}^{-\zeta}(t), \tag{3.1}$$

where $\rho_0$ represents the channel-power gain at a particular reference distance of 1 m, $\zeta$ is the path-loss exponent, and $d_{u_{ij}}(t)$ represents the distance between two UAVs. In our study, we utilize the time division multiple access-based MAC to ensure that each UAV gets a dedicated timeslot for broadcasting with interference avoidance. We assume that some UAVs transmit data with a common probability of $\phi$ independently at each timeslot. Therefore, the expected interference $\bar{I}_{ij}(t)$ on link $u_{ij}$ is expressed as $\bar{I}_{ij}(t) = \sum_{\hbar \in U, \hbar \neq i,j} \phi \mathcal{G}_{\hbar j}(t) P_{tx}^{\hbar}$, where $\phi$ represents the interference rate with a set of $\hbar \neq i, j$ active neighboring UAVs of receiving UAV $u_j$ [99]. Thus, the approximate SINR $\Psi_{ij}(t)$ in dB between two UAVs under $\bar{I}_{ij}(t)$ is expressed as follows:

$$\Psi_{ij}(t) = 10log\frac{\mathcal{G}_{ij}(t)P_{tx}}{\bar{I}_{ij}(t)+N_o} = 10log\frac{\mathcal{G}_{ij}(t)P_{tx}}{\sum_{k\neq i,j}\phi\mathcal{G}_{kj}(t)P_{tx}^{k}+N_o}, \quad (3.2)$$

where $N_o$ is the assumed additive white Gaussian noise. According to the $\Psi_{ij}(t)$, the per-hop packet error rate $PER_{ij}(\Psi_{ij}(t))$between transmitter and receiver is estimated as follows [152], [153]:

$$PER_{ij}\left(\Psi_{ij}(t)\right) \approx \begin{cases} 1, & \Psi_{ij}(t) < \Psi_{th} \\ a_n \exp\left(-\mathcal{g}_n\Psi_{ij}(t)\right), & \Psi_{ij}(t) \geq \Psi_{th} \end{cases}, \quad (3.3)$$

where $\Psi_{th}$ is the SINR threshold. Additionally, $a_n$ and $\mathcal{g}_n$ are transmission-mode-dependent parameters whose values are stated in [153]. The average one-hop delay $t_{u_{ij}}^{mac}$ between two UAVs is expressed as follows [99]:

$$t_{u_{ij}}^{mac} = \frac{\tau_{ij}}{[1-PER_{ij}(\Psi_{ij}(t))]}, \quad (3.4)$$

where $\tau_{ij} = t_{ACK} - t_{tx}$ represents the round-trip time of one-hop transmission. Here, the $t_{tx}$ and $t_{ACK}$ represents the packet transmission time and acknowledge (ACK) reception time. For system bandwidth B, the transmitted data rate $R_{u_{ij}}$ is estimated as $R_{u_{ij}} = B\log_2[1 + \Psi_{ij}(t)]$.

### 3.2.2 Topology Construction Model in FANETs

We design a two-phase topology control to construct the topology for a large-scale FANET and above the TC, a position-based multi-hop routing incorporated with the QL is applied to perform surveillance.

In the first phase of topology control, UAVs use the distributed VFMC algorithm to construct the initial FANET topology $G(t_n) = (V, E, M)$, where $V \in \{U \cup BS\}$ represents the vertices consisting of UAV $u_i \in U$ and BS. Here, each UAV $u_i \in U$, communicates with neighboring UAVs within the communication range $R$, and senses $m_k \in M$, MGTs within the $R_C$. A wireless link $E$, between two UAVs $u_{ij}$ exists if the distance between two UAVs $d_{u_{ij}} < R$. The objective of the VFMC is mathematically illustrated as follows. We find the mobility information $MI \in (\boldsymbol{p}_{u_i}, \boldsymbol{v}_{u_i}, \boldsymbol{a}_{u_i})$ for each UAV $u_i$ at each $t_n$ so that

$$\omega\max_{MI}\sum_{k=1}^{M} m_k + (1-\omega)\max_{MI} LG_{u_{ij}}, \quad u_j \in N(u_i), \quad (3.5)$$

subject to the following constraints:

$$v_{min} \leq v_{u_i} \leq v_{\max}, \qquad \forall u_i \in U, \quad (3.5a)$$

$$a_{min} \leq a_{u_i} \leq a_{max}, \qquad \forall u_i \in U, \tag{3.5b}$$

$$h_{min} \leq h_{u_i} \leq h_{max}, \qquad \forall u_i \in U, \tag{3.5c}$$

$$R_{sd} \leq d_{u_{ij}}(t) < R. \qquad \forall u_i \in U, \tag{3.5d}$$

According to (3.5), the VFMC requires to determine the $MI$ at each $t_n$ that consist of the position, velocity, and acceleration $(\boldsymbol{p}_{u_i}, \boldsymbol{v}_{u_i}, \boldsymbol{a}_{u_i})$ for each UAV $u_i \in U$ so that the maximum number of MGTs $m_k \in M$ can be tracked, and the LG among one-hop neighboring UAVs $N(u_i)$is maximized for that particular $t_n$. In (3.5), the first component determines the mission performance as MGTs coverage, and the second component determines the communication performance by maximizing the LG with $N(u_i)$ UAVs. The weighting parameter $\omega \leq 1$, determines the balance between mission and communication performance, whose value is adaptively set according to the node density by performing the topology adjustment. Constraints (3.5a), (3.5b), and (3.5c) indicate that the velocity of each UAV must be within $v_{u_i} \in [v_{\min}\, v_{\max}]$, the acceleration should be within $a_{u_i} \in [a_{\min}\, a_{\max}]$, and the flying height should be adjusted within $h_{u_i} \in [h_{\min}\, h_{\max}]$, respectively. According to (3.5d), the relative distance between two neighboring UAVs $d_{u_{ij}}(t)$ must be greater than the separating distance $R_{sd}$ to avoid inter-UAV collisions and to minimize the overlap in $R_C$ of adjacent UAVs. Additionally, $d_{u_{ij}}(t)$ should be less than the communication range $R$ to maximize the LG and maintain the SINR level in U2U links. The details of the VFMC algorithm is given in Section 3.3.1.

In the second phase of topology control, the UAV swarm utilizes the given $MI$ to divide the whole swarm network into multiple stable clusters to perform data aggregation under the cluster-size constraint. The EMFC obtains the fittest UAVs as the CH by calculating the priority index (PI) in association with the fuzzy logic. The fuzzy logic blends a few parameters such as NI that is computed from LG and LQ, ND, and RE within $N(u_i)$ of each UAV. The fuzzy logic is an appropriate tool for blending the above parameters for better decision-making. The fuzzy logic system has three steps: fuzzification, rule-based fuzzy inference, and defuzzification. During fuzzification, the above four parameters are mapped into normalized crisp values representing input fuzzy sets utilizing two widely used fuzzy membership functions to determine the degree of each fuzzy input via three predefined linguistic values. During the second step, the predefined IF–THEN rule is combined with each fuzzy input, which gives an aggregated fuzzy output, PI. During defuzzification, the aggregated fuzzy output is converted into an output crisp PI for each UAV utilizing the center of gravity (CoG) [154]. The UAV with the highest PI within $N(u_i)$ works as a leader CH that carries the ADPs. ADPs are considered learning agents in the next TAQR. The details of the EMFC algorithm is given in Section 3.3.2.

### 3.2.3 Q-Learning-Based Inter-Cluster Routing Model

The third module of JTCR is TAQR and it performs inter-cluster routing in FANETs to route ADPs from CH UAV to BS, as shown in Figure 3.2. In this sub-section, the relation between two-phase topology control and TAQR multi-hop routing is briefly discussed. QL evaluates the expected value of the cumulative multi-objective reward and achieves the instant optimal policy according to the historical experience in an unknown environment without having any central controller. According to the local view of the agent, the decision-making process using the QL can be expressed as a partial Markov decision process (PMDP) tuple $(s, a, p, r)$, where $s$ represents the finite set of states, $a$ represents the finite set of actions, $p$ represents the state transition probability, and $r$ represents the reward that evaluates an action. We found QL more suitable in our resource constraint dynamic FANET environment compared to other RL techniques. This is because deep learning algorithms require higher computational complexity, large training samples, and large memory to preserve the history of state-action pairs in replay buffer.

The VFMC predicts the mobility of each UAV in the next timeslot based on the mobility state in the current timeslot by interacting with both one-hop neighboring UAVs and the detected MGTs, which implies the similarity with the Markov property. This is because Markov property states that the next state of the process depends on the current state of the process. The ADPs are forwarded from a CH UAV (source state) to the BS (final state) by selecting a relay UAV located toward the direction of the BS. Such a forwarding includes a new state transition probability to select an optimal routing path in terms of delay, path stability, and balanced energy consumption. Owing to these relationships, the FANET routing decision can be formulated as a PMDP. The PMDP process in QL gives limited action space to each agent because it is only defined by its one-hop neighboring UAVs.

In TAQR, as shown in Figure 3.2, the ADPs initially held by all source CH UAVs act as learning agents, and the entire FANET topology is the environment. The current state of the ADPs is the location of the source CH UAV, which is routed to the BS through the state transition from one UAV to another until it is delivered to the BS. During the state transition, the next state of the ADPs can be the BS or neighboring UAVs (relay state) that are considered in the PFC set. When a CH UAV $u_i$ transmits the ADPs to its one-hop neighbor UAV $u_j$, this is defined as an action $a_{u_{ij}}$ and the corresponding link is $u_{ij}$. Through action $a_{u_{ij}}$, the state of the ADPs moves from $s_{u_i}$ to $s_{u_j}$, and the corresponding $a_{u_{ij}}$ is evaluated by a multi-objective reward $r_{u_{ij}}$ consisting of a one-hop delay defined by PTT, NI, and RE of the selected relay UAV. For each relay-link selection, the UAV receives a reward or a penalty. Progressively, each UAV collects a Q-value that contributes to an optimal policy in which the cumulative reward is maximized over time.

The Q-values are updated at each forwarding UAV according to the following equation:

$$Q^n(s_{u_i}, a_{u_{ij}}) \leftarrow Q^p(s_{u_i}, a_{u_{ij}}) + \alpha_{u_{ij}}[r_{u_{ij}} + \lambda_{u_{ij}} \max_{a'_{u_{ij}}} Q(s_{u_j}, a'_{u_{ij}}) - Q^p(s_{u_i}, a_{u_{ij}})], \quad (3.6)$$

where $Q^n(s_{u_i}, a_{u_{ij}})$ and $Q^p(s_{u_i}, a_{u_{ij}})$ represents the new and previous Q-value. The term $\max_{a'_{u_{ij}}} Q(s_{u_j}, a'_{u_{ij}})$ represents the expected maximum Q-value in the next state $s_{u_j}$ when the agent selects the best learned action $a'_{u_{ij}}$. The $\alpha_{u_{ij}}$ and $\lambda_{u_{ij}}$ represent the learning rate and discount factor, respectively, whose values are within [0 1]. The $\alpha_{u_{ij}}$ indicates the degree to which newly obtained information overrides the old information, and this parameter controls the convergence of the QL. The value of $\lambda_{u_{ij}}$ controls the importance of future rewards and defines how much the QL learns from its previous mistakes. As a result, to estimate the precise Q-value, the $\alpha_{u_{ij}}$ and $\lambda_{u_{ij}}$ should be adjusted adaptively to cope with the dynamism of FANETs. The details of the TAQR algorithm will be given in Section 3.3.3. The relationship stack of the two-phase topology control and routing in JTCR is illustrated in Figure 3.2.

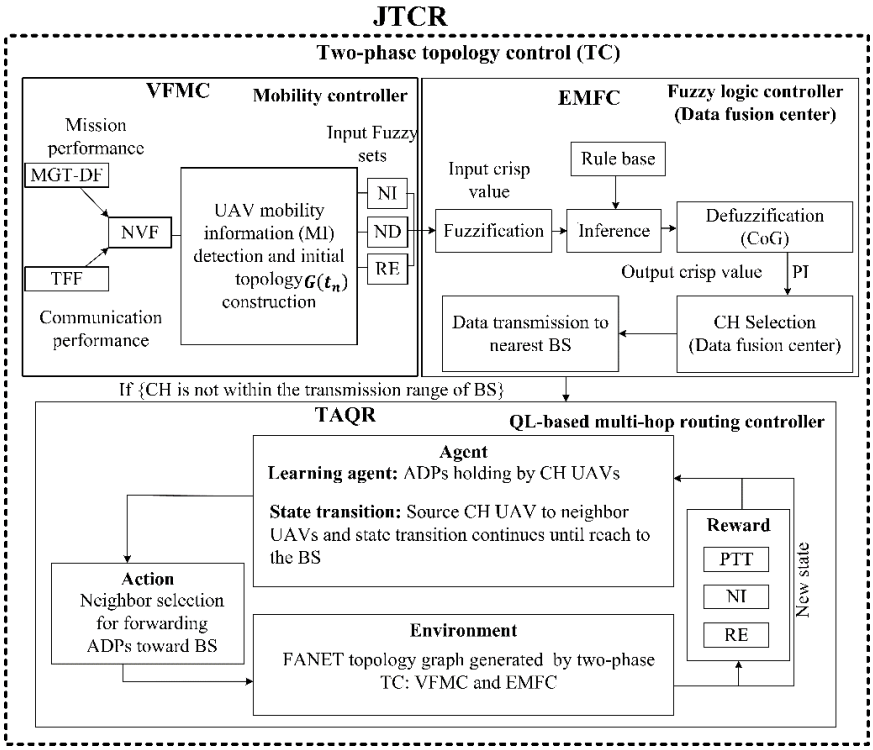



Figure 3.2 The relationship stack of the two-phase topology control (VFMC and EMFC) and QL-based multi-hop routing (TAQR) in JTCR.

## 3.3 Topology Control and Routing Algorithms

In this section, we first present the topology construction of the JTCR via a two-phase topology control to perform the surveillance. Then, we derive the multi-hop routing algorithm (i.e., TAQR) to route the ADPs to the BS.

### 3.3.1 Distributed VFMC Algorithm

The distributed VFMC is the first module of our JTCR, and it constructs the initial topology for FANET by detecting the $MI$ for each UAV. We assume that the position of the MGTs is unknown to all UAVs and that they are tracked by the onboard target tracking sensor (e.g., LiDAR) within the $R_C$ when they fly over the MGTs. They also exchange hello packets with one-hop neighbor UAVs within their communication range, $R$, at certain hello interval to maintain aerial connectivity with updated topology prediction. Low hello interval provides better positioning accuracy of neighboring UAVs while increasing the control overhead in turn. A high hello interval reduces the hello packet but forces UAVs to deal with inaccurate positionings of neighboring UAVs, leading to incorrect topology prediction. Therefore, we obtain an optimal hello interval for each UAV to significantly reduce the control overhead and to control the FANET topology with the updated mobility information.

The VFMC has two virtual force vectors, and each has two force components that create a balance between mission and communication performance, as given in (3.5). The first virtual force is the MGT-DF, which ensures coverage efficiency by tracking the maximum MGTs within *D*. The MGT-DF has two force components and obtained with the help of onboard LiDAR data, which are used to localize DAs or isolated MGT locations within the $R_C$. The magnitude and direction of the virtual forces are computed based on the Euclidean distance between the UAVs and the DAs of MGTs represented as $\boldsymbol{d}(u_i, DA_i)$ or the isolated MGTs represented as $\boldsymbol{d}(u_i, m_k)$ considering the uneven distribution of $m_k$ MGTs, as shown in Figure 3.3 (a)-(b).

The first force component of the MGT-DF is the attractive force toward the high DAs of MGTs $\boldsymbol{F}(u_i, DA_i)$, within the $R_C$ of each UAV $u_i$, as shown in Figure 3.3(a). It is computed based on Coulomb's law:

$$\boldsymbol{F}(u_i, DA_i) = \sum_{i=1}^{DA_i} K_{DA_i} \times \frac{1}{\boldsymbol{d}(u_i, DA_i)^2}, \tag{3.7}$$

where $K_{DA_i}$ represents the attractive force constant toward high DAs of MGTs and depends on the size of the DAs. If a UAV $u_i$ tracks two different-sized DAs within its $R_C$, the resultant force direction will be slightly closer to the large DA (DA-2 in Figure 3.3(a)) and will be adjusted following the line between two DAs to efficiently maintain the travel distance of each UAV while maximizing coverage, as shown in Figure 3.3(a).

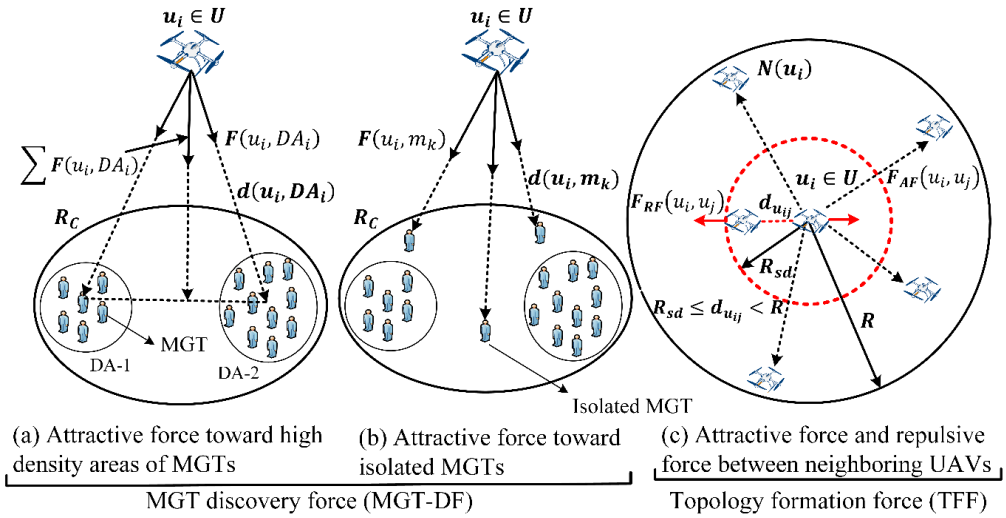



Figure 3.3 Geometric diagram of virtual forces and their motion components that act on each UAV in a UAV swarm.

The second force component is attractive force toward the isolated MGTs $\boldsymbol{F}(u_i, m_k)$ within the $R_C$ of each UAV $u_i$, as shown in Figure 3.3(b). It is also computed based on Coulomb's law as given below:

$$\boldsymbol{F}(u_i, m_k) = \sum_{k=1}^{m_k} K_{m_k} \times \frac{1}{\boldsymbol{d}(u_i, m_k)^2}, \tag{3.8}$$

where $K_{m_k}$ represents the attractive force constant toward the isolated MGTs, and $K_{m_k} \ll K_{DA_i}$ to give attraction priority to the DAs of the MGTs. The attraction values of the force decrease with the increasing distance between the UAVs and DAs or MGTs. As a result, we adjust the attractive force value by controlling the UAV height within $[h_{\min}\ h_{\max}]$. Therefore, the MGT-DF $\boldsymbol{F}_{DF}^{u_i}$ for each UAV $u_i$ is computed as follows:

$$\boldsymbol{F}_{DF}^{u_i} = \boldsymbol{F}(u_i, DA_i) + \boldsymbol{F}(u_i, m_k), \tag{3.9}$$

The $\boldsymbol{F}_{TFF}^{u_i}$ is the second virtual force that maintains the separating distance to avoid collisions while reducing overlap in $R_C$ among neighboring UAVs. It also maintains communication QoS by controlling the relative distance between UAVs to maximize the LG as much as possible, as given in (3.5) and (3.5d). The $\boldsymbol{F}_{TFF}^{u_i}$ has two force-vector components: the attractive force $\boldsymbol{F}_{AF}(u_i, u_j)$ and the repulsive force $\boldsymbol{F}_{RF}(u_i, u_j)$, as shown in Figure 3.3(c). Both $\boldsymbol{F}_{AF}(u_i, u_j)$ and $\boldsymbol{F}_{RF}(u_i, u_j)$ are obtained according to inter-UAV distance to satisfy the imposed flocking constraints given in (3.5d)**.** As UAVs are dragged toward the DAs of MGTs, it is necessary to maintain a separating distance $R_{sd}$, among the $N(u_i)$ of

each UAV to avoid inter-UAV collisions and reduce overlapping in $R_C$. $\boldsymbol{F}_{RF}(u_i, u_j)$ is computed as follows:

$$\boldsymbol{F}_{RF}(u_i, u_j) = \begin{cases} \sum_{u_j \in N(u_i)} K_R \left( \frac{1}{d_{u_{ij}}^2} - \frac{1}{R_{sd}^2} \right), & 0 < d_{u_{ij}} < R_{sd} \\ 0, & R_{sd} \leq d_{u_{ij}} < R \end{cases}, \tag{3.10}$$

The $\boldsymbol{F}_{RF}(u_i, u_j)$ is activated only when the separating distance constraint is violated ($d_{u_{ij}} < R_{sd}$). The value of the repulsive force constant is always $K_R \gg K_A$ to strictly maintain the separating distance, where $K_A$ is the attractive force constant. When UAVs flock by satisfying the relative distance constraints given in (3.5d), the $\boldsymbol{F}_{RF}(u_i, u_j)$ is zero. However, due to the uncertainty in UAV flocking (such as uneven distribution of MGTs in *D*), and the UAVs may frequently fly away from each communication range $R$. As a result, to maintain the strong neighbor relationship, we control the relative distance between two UAVs within the range given in (3.5d) by applying an attractive force $\boldsymbol{F}_{AF}(u_i, u_j)$. The $\boldsymbol{F}_{AF}(u_i, u_j)$ is exponentially increases when the relative distance is increased within $R_{sd} \leq d_{u_{ij}} < R$, and it becomes zero otherwise. The $\boldsymbol{F}_{AF}(u_i, u_j)$ is computed as follows:

$$\boldsymbol{F}_{AF}(u_i, u_j) = \begin{cases} \sum_{u_j \in N(u_i)} K_A \left( R_{sd} - \boldsymbol{d}_{u_{ij}} \right) \exp \left[ \frac{\left( \boldsymbol{d}_{u_{ij}} - R_{sd} \right)^2}{R} \right], & R_{sd} \leq d_{u_{ij}} < R \\ 0, & otherwise \end{cases}, \tag{3.11}$$

Therefore, the $\boldsymbol{F}_{TFF}^{u_i}$ is computed as follows:

$$\boldsymbol{F}_{TFF}^{u_i} = \omega_1 \boldsymbol{F}_{AF}(u_i, u_j) + \omega_2 \boldsymbol{F}_{RF}(u_i, u_j), \tag{3.12}$$

where$\omega_1 + \omega_2 = 1$ is the force weight, which values are adaptively adjusted by applying topology adjustment according to the node density within neighboring UAVs to meet safety distances and QoS in communication requirements to maintain a sufficient LG and the desired SINR.

The topology adjustment is performed by sensing the changes in neighboring distances represented as $TA_1^{tt'}$ and $TA_2^{tt'}$ for each UAV $u_i$, at two different times $t$ and $t'$, where $t' > t$. This is computed as follows:

$$TA_1^{tt'} = \exp \left[ \eta_1 \left\{ R_{sd} - \min_{u_j \in N(u_i)} d_{u_{ij}} \right\} \right], \tag{3.13}$$

$$TA_2^{tt'} = \exp\left[\eta_2\left\{\max_{u_j \in N(u_i)} d_{u_{ij}} - R\right\}\right], \tag{3.14}$$

where $\eta_1$ and $\eta_2$ are sensitivity parameters. The $TA_1^{tt'}$ measures the degree of violation in imposed safety-distance constraints, $d_{u_{ij}} > R_{sd}$, and immediately increases exponentially if $d_{u_{ij}} < R_{sd}$. If the value of $TA_1^{tt'}$ is greater than the threshold $\delta_1$, the weight becomes $\omega_2 > \omega_1$ to increase the effect of repulsive force given by (3.10). The $TA_2^{tt'}$ measures the degree of violation for the imposed QoS constraints, $d_{u_{ij}} < R$. It immediately increases exponentially if $d_{u_{ij}} > R$. If the value of $TA_2^{tt'}$ is greater than the threshold $\delta_2$, the weight becomes $\omega_1 > \omega_2$ to increase the effect of attractive force given by (3.11).

Finally, the NVF $\boldsymbol{F}_{NVF}^{u_i}$ acting on each UAV is computed via vector addition as follows:

$$\boldsymbol{F}_{NVF}^{u_i} = \boldsymbol{F}_{DF}^{u_i} + \boldsymbol{F}_{TFF}^{u_i} \tag{3.15}$$

According to the Newton's second law of motion, at each $t_n$, each UAV $u_i$ utilizes the $\boldsymbol{F}_{NVF}^{u_i}$ as its control input (acceleration) to determine the $MI \in (\boldsymbol{a}_{u_i}, \boldsymbol{v}_{u_i}, \boldsymbol{p}_{u_i})$ and it can be computed as follows:

$$\boldsymbol{a}_{u_i}(t_n) = \left(\frac{\boldsymbol{F}_{NVF}^{u_i}}{\left\|\boldsymbol{F}_{NVF}^{u_i}\right\|}\right) \times \tan^{-1}\left(\left\|\boldsymbol{F}_{NVF}^{u_i}\right\|\right) \times \frac{2}{\pi} \times a_{\max}, \tag{3.16}$$

$$\boldsymbol{v}_{u_i}(t_{n+1}) = \boldsymbol{v}_{u_i}(t_n) + \boldsymbol{a}_{u_i}(t_n) \times \Delta t, \tag{3.17}$$

$$\boldsymbol{v}_{u_i}(t_{n+1}) = \begin{cases} \boldsymbol{v}_{u_i}(t_{n+1}), & \left\|\boldsymbol{v}_{u_i}(t_{n+1})\right\| < v_{\max} \\ \left[\frac{v_{u_i}(t_{n+1})}{\left\|v_{u_i}(t_{n+1})\right\|}\right] \times v_{\max}, & \left\|\boldsymbol{v}_{u_i}(t_{n+1})\right\| \geq v_{\max} \end{cases}, \tag{3.18}$$

$$\boldsymbol{p}_{u_i}(t_{n+1}) = \boldsymbol{p}_{u_i}(t_n) + \boldsymbol{v}_{u_i}(t_n)\Delta t + \frac{1}{2}\boldsymbol{a}_{u_i}(t_n)\Delta t^2, \tag{3.19}$$

where $\boldsymbol{v}_{u_i}(t_{n+1})$ and $\boldsymbol{p}_{u_i}(t_{n+1})$ represents UAV velocity and position in the next time slot, respectively. Here, $\|.\|$ represents the magnitude of a vector. According to (3.16)−(3.19), at each $t_n$, UAV $u_i$ utilizes $\boldsymbol{F}_{NVF}^{u_i}$ to compute its acceleration $\boldsymbol{a}_{u_i}$, velocity $\boldsymbol{v}_{u_i}$, and position

$\boldsymbol{p}_{u_i}$. As the magnitude of $\boldsymbol{F}_{NVF}^{u_i}$ varies from 0 to $+\infty$, to set the acceleration within $[a_{\min}\ a_{\max}]$, we apply a trigonometric function in (3.16). Similarly, to keep the velocity within $[v_{\min}\ v_{\max}]$, we use equation (3.18). According to the $MI$, FANET topology $G(t_n)$ is constructed.

We derive the LG between two adjacent UAVs and determine the adaptive hello interval to optimize the number of hello packets. The LG defines the link subsistence time between two adjacent UAVs, which is a function of the relative distance, relative velocity, and communication range $R$ of the UAVs. Let UAV $u_i$ receive two consecutive hello packets at time $t$ and $t'$ $(t' > t)$ from the neighbor UAV $u_j \in N(u_i)$, as shown in Figure 3.4.

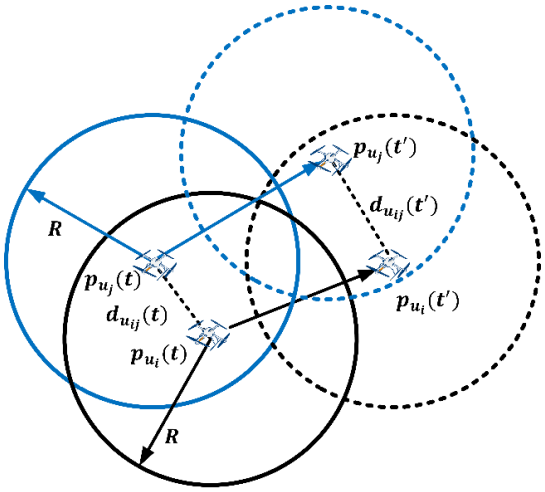



Figure 3.4 The LG and hello interval estimation between two neighboring UAVs (receding scenario).

According to the hello packet, UAV $u_i$ obtains the position of UAV $u_j$ at times $t$ and $t'$ as $p_{u_j}(t)$ and $p_{u_j}(t')$, respectively. Similarly, from $MI$, UAV $u_i$ detects its own position as $p_{u_i}(t)$ and $p_{u_i}(t')$. The change in relative distance $\Delta d$ between two UAVs from $t$ to $t'$is computed as $\Delta d = d_{u_{ij}}(t') - d_{u_{ij}}(t) = \left\|p_{u_i}(t') - p_{u_j}(t')\right\| - \left\|p_{u_i}(t) - p_{u_j}(t)\right\|$. When $\Delta d > 0$, receding motion occurs between two UAVs. From $t$ to $t'$, the predicted relative velocity between the two is $\Delta d/(t' - t)$. As a result, the expected time needed to move away from each other's communication range $R$, is estimated as $\frac{\left\|R - d_{u_{ij}}(t')\right\|}{\Delta d/(t'-t)}$. When $\Delta d \leq 0$, the approaching motion occurs between the two UAVs. The expected time to cross distance $R$

at a minimum relative speed $v_{\min}$, is computed as $\Gamma_{\max} = {}^{R}/_{v_{min}}$. Finally, $LG_{u_{ij}}$ is estimated by each UAV $u_i$ for $u_j \in N(u_i)$, as given below:

$$LG_{u_{ij}} = \begin{cases} \frac{\left\| R - d_{u_{ij}}(t') \right\|}{\frac{\Delta d}{(t'-t)}}, & if\ \Delta d > 0 \\ \frac{d_{ij}(t')}{R} \times \Gamma_{\max}, & if\ \Delta d \leq 0 \end{cases} \tag{3.20}$$

Instead of using a fixed hello interval, it should be optimized with topological alterations to offer an updated neighbor table to the routing protocol. As a result, we set the interval for each UAV equal to the minimum LG found within $N(u_i)$ of UAV $u_i$. The adaptive hello interval $HI$ is estimated as follows:

$$HI = \sigma \times \left[ \min_{u_j \in N(u_i)} LG_{u_{ij}} \right], \tag{3.21}$$

where $\sigma$ represents the frequency factor whose value is within $\sigma \in [0\ 1]$, and the default value is 0.5. A UAV $u_i$, senses the changes in $N(u_i)$ set from time $t$ to $t'$ represented as $N^t(u_i)$ and $N^{t'}(u_i)$ respectively, and the value of $\sigma$ is adaptively controlled as follows:

$$\sigma = 1 - \left[ \frac{\left| N^{t'}(u_i) \cup N^t(u_i) \right| - \left| N^{t'}(u_i) \cap N^t(u_i) \right|}{\left| N^{t'}(u_i) \cup N^t(u_i) \right|} \right], \tag{3.22}$$

To estimate the $LQ_{u_{ij}}$ of a bidirectional link, we use the expected transmission count [155], which is computed by counting the number of transmitted hello packets and those receiving ACKs for a particular hello interval, as given in (3.21) and computed as follows:

$$LQ_{u_{ij}} = \frac{fd_{u_{ij}}}{rf_{u_{ij}}}, \tag{3.23}$$

where $fd_{u_{ij}}$represents the forward delivery ratio of successfully sending hello packets to the receiver, and $rf_{u_{ij}}$ represents the reverse delivery ratio of successfully receiving the ACK for each hello packet from the receiver.

Each UAV shares the hello packets with $N(u_i)$ UAVs using the hello interval given in (3.21), including a unique UAV ID, a hello packet sequence number, an $MI$ with parameters of delay, an LG, an LQ, an ND, and an RE, which are utilized in the phase-2 of topology control, and the routing for better decision making. The VFMC algorithm is described in Algorithm 3.1.

---

**Algorithm 3.1: VFMC**

---

**Input**: Locations of DAs and isolated MGTs $m_k \in M$, UAVs $u_i \in U$ candidate position $p_{u_i}$, and predefined thresholds $\delta_1$ and $\delta_2$.
**Output:** Topology $G(t_n) = (V, E, M)$ with $MI(t_n) \in (\boldsymbol{p}_{u_i}, \boldsymbol{v}_{u_i}, \boldsymbol{a}_{u_i})$ and Neighbor relationship parameters: LG, hello interval (HI) and LQ
1: **Proceed** to the next time slot $t_{n+1}$
**Step 1:** Broadcast HELLO Packets (HPs) with current position
2: **for** each $u_i \in U$ **do**
3: Broadcast HPs to one-hop neighbor
4: **end for**
**Step 2:** Information update for neighbor discovery
5: **for** ∀ received *HPs* at UAV $u_i$ from neighbor $u_j$ **do**
6: Get originator $u_j$ unique UAV ID
7 **if** $\left[u_j \in N(u_i)\right]$ **then**
8: **if** (*received HP sequence > record HP sequence)* **then**
9: Update the position of $u_j$
10: **end if**
11: **else**
12: Add a new record for $u_j$ to neighbor set $N(u_i)$
13: **end if**
14: **end for**
**Step 3:** Mobility information $MI(t_n) \in (\boldsymbol{p}_{u_i}, \boldsymbol{v}_{u_i}, \boldsymbol{a}_{u_i})$ detection
15: **for** each UAV $u_i$ having one-hop neighbor $u_j \in N(u_i)$ **do**
16: Calculate the MGT-DF $\boldsymbol{F}_{DF}^{u_i}$ using (3.7)−(3.9)
17: **if** $\left(TA_1^{tt'} > \delta_1\right)$ **then** // Violation of the safety constraint
18: Set the force weight as $\omega_2 > \omega_1$ in (3.12)
19: **else if** $\left(TA_2^{tt'} > \delta_2\right)$ **then** //Violation of the SINR constraint
20: Set the force weight as $\omega_1 > \omega_2$ in (3.12)
21: **else**
22: Set the force weight as $\omega_1 = \omega_2$ in (3.12)
23: **end if**
24: Calculate the TFF $\boldsymbol{F}_{TFF}^{u_i}$ using (3.10)−(3.12)
25: Compute the NVF $\boldsymbol{F}_{NVF}^{u_i}$ using (3.15)
26: Compute $MI(t_n) \in (\boldsymbol{p}_{u_i}, \boldsymbol{v}_{u_i}, \boldsymbol{a}_{u_i})$ using (3.16)−(3.19)
27: Construct the FANET topology $G(t_n) = (V, E, M)$
28: Calculate the LG using (3.20)
29: Update the HI using (3.21)−(3.22)
30: Calculate the LQ using (3.23)
31: **end for**

---

### 3.3.2 EMFC Clustering

The EMFC is the second phase of the JTCR that divides the first-phase topology $G(t_n)$, into a set of stable clusters to perform data aggregation. The EMFC clustering process has

three steps: UAV fitness PI calculation to select a $CH \subseteq U$ UAV to act as a local leader to perform data fusion; cluster formation via follower selection; and cluster equalization. As $G(t_n)$ is connected, cluster invitation and joining requests are not required. The PI calculation process utilizes fuzzy logic, which takes into input two link-related parameters (i.e., LG and LQ) as NI and two UAV state-related parameters (i.e., ND and RE) to obtain the output PI. Each UAV $u_i$, shares PI with $u_j \in N(u_i)$ via the hello packets, and the UAV $u_i$ having the highest PI within its $N(u_i)$ vicinity, declares itself to be the $CH_i$. The PI calculation process using fuzzy logic involves three steps. In the first step, each UAV $u_i$ normalizes the above fuzzy input sets by using the maximum value of corresponding input parameters collected through the received hello packets from one-hop neighbor $u_j \in N(u_i)$.

The UAV $u_i$ having the largest $LG_{u_{ij}}$ computed in (3.20) within its $N(u_i)$ UAVs is quite suitable for becoming a stable CH, because the highest $LG_{u_{ij}}$ gives better link subsistence probability with neighbors. A larger $LQ_{u_{ij}}$ computed in (3.23) offers better link reliability for collecting data from the other CMs. $NI_{u_{ij}}$ represents the neighbor intimacy of a UAV with its $N(u_i)$ UAVs consisting of both $LG_{u_{ij}}$ and $LQ_{u_{ij}}$. A larger $NI_{u_{ij}}$ gives better stability and better PDR during data aggregation at each elected CH. $NI_{u_{ij}}$ is computed as follows:

$$NI_{u_{ij}} = \left( \frac{LG_{u_{ij}}}{\max\limits_{u_j \in N(u_i)} LG_{ij}} \right) \times \left( \frac{LQ_{u_{ij}}}{\max\limits_{u_j \in N(u_i)} LQ_{ij}} \right). \tag{3.24}$$

As all UAVs attempt to move to the dense areas of the MGTs, those having the greatest number of neighbors defined by the ND, offer better leadership and minimize the number of CH requirements. A UAV $u_i$having node degree $ND_{u_i}$ is normalized as follows:

$$ND_{u_i} = \frac{ND_{u_i} - \min\limits_{u_j \in N(u_i)} ND_{u_j}}{\max\limits_{u_j \in N(u_i)} ND_{u_j}}. \tag{3.25}$$

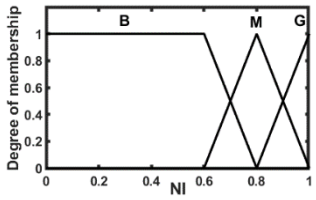
As the CH UAV performs data aggregation and compression for its cluster members (CMs), it requires sufficient energy. The UAV having the highest RE within its one-hop vicinity provides better stability for CH. The RE of a UAV $RE_{u_i}$ is normalized as follows:

$$RE_{u_i} = \frac{RE_{u_i} - \min\limits_{u_j \in N(u_i)} RE_{u_j}}{\max\limits_{u_j \in N(u_i)} RE_{u_j}}. \tag{3.26}$$

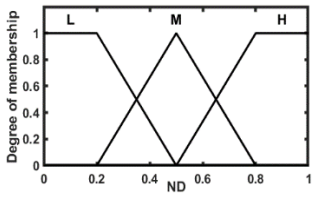
We use two fuzzy membership functions (i.e., triangular and trapezoidal) to convert these input crisp values to fuzzy values. The associated linguistic values and the ranges of fuzzy membership functions are given in Table 3.2 and Figure 3.5 (a)–(d).

Table 3.2 Input and output fuzzy sets with linguistic values.

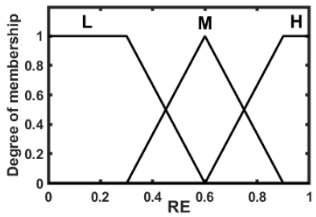
| Input/Output | Linguistic value |
|---|---|
| NI | Bad (B), Medium (M), and Good (G) |
| ND | Low (L), Medium (M), and High (H) |
| RE | Low (L), Medium (M), and High (H) |
| PI | Very low (VL), Low (L), Unpreferable (U), Medium (M), High (H), and Very high (VH) |



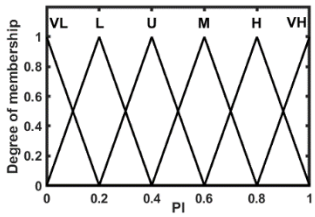

(a) Fuzzy membership of NI.




(b) Fuzzy membership of ND.




(c) Fuzzy membership of RE.




(d) Fuzzy membership of PI.

Figure 3.5 Fuzzy membership values of inputs (NI, ND, and RE) and output (PI) fuzzy sets in the EMFC.

In the second step, the states of the fuzzy input sets are evaluated by each predefined IF–THEN rule $R_N$ given in Table 3.3 by using the MIN–MAX method [154] to find the aggregated fuzzy output PI given in (3.27). The pre-defined linguistic values of output PI and associated fuzzy membership function ranges are also given in Table 3.3 and Figure 3.5(d). Because we consider three inputs with three linguistic values, the number of rules in Table 3.3 is $3^3 = 27$. The multi-objective PI is computed as follows:

$$PI = \rho_1 NI_{u_{ij}} + \rho_2 ND_{u_{ij}} + \rho_3 RE_{u_i}, \tag{3.27}$$

where $\rho_1$, $\rho_2$, and $\rho_3$ are the weighting factors of the fuzzy rules add $\rho_1 + \rho_2 + \rho_3 = 1$. Finally, the defuzzification point as the crisp output of PI is obtained by applying the CoG method in the aggregated fuzzy output of PI. In Figure 3.5(d), the corresponding centroid value of *x*-coordinate after evaluating all IF-THEN rules represents the final decoded crisp PI that gives the fitness of each UAV.

Table 3.3 Fuzzy IF–THEN rules to find the PI for UAVs.

| $R_N$ | IN | | | O | $R_N$ | IN | | | O |
|---|---|---|---|---|---|---|---|---|---|
| | NI | ND | RE | PI | | NI | ND | RE | PI |
| 1 | B | L | L | VL | 15 | M | M | H | M |
| 2 | B | L | M | VL | 16 | M | H | L | U |
| 3 | B | L | H | VL | 17 | M | H | M | M |
| 4 | B | M | L | VL | 18 | M | H | H | M |
| 5 | B | M | M | L | 19 | G | L | L | L |
| 6 | B | M | H | L | 20 | G | L | M | M |
| 7 | B | H | L | VL | 21 | G | L | H | H |
| 8 | B | H | M | VL | 22 | G | M | L | U |
| 9 | B | H | H | U | 23 | G | M | M | M |
| 10 | M | L | L | L | 24 | G | M | H | H |
| 11 | M | L | M | U | 25 | G | H | L | U |
| 12 | M | L | H | M | 26 | G | H | M | H |
| 13 | M | M | L | L | 27 | G | H | H | VH |
| 14 | M | M | M | M | | | | | |

During cluster formation phase, each elected $CH_i$ forms a cluster using its one-hop neighbor $u_j \in N(u_i)$ as the CM represented as $CH_i \in \{CM_{CH_i}^i\}$, where $i = \{1,2,\cdots\}$ is the index of each CM under the respective CH, as shown in Figure 3.6. During the cluster-size equalization process, if each $CH_i$ has a number of CMs greater than the threshold $N_{CH_i} > N_{\max}$, it releases the CMs according to the order of lowest LG until the size becomes $N_{CH_i} \leq N_{max}$. The released CMs can be borrowed by a neighbor $CH_j$ if it has fewer $CM_{CH_j}^i$. The equal cluster size produces fewer contention delays during data aggregation at each CH and creates a balance of inter- and intra-cluster data transmissions. Each $CH_i$ collects the sensing data from its $CM_{CH_i}^i$ and acts as a data fusion center to prepare the ADPs by performing data aggregation.

All $CH_i$ participate in the next TAQR routing as source nodes to deliver the ADPs to the BS via multi-hop routing. The EMFC clustering process reduces the number of transmissions in the FANET compared with each UAV individually transmitting to the BS via multi-hop routing. Some UAVs may not belong to any CH, owing to the cluster-size constraints, and they participate in the next TAQR with their own sensing data without facing any problems as topology $G(t_n)$ is connected. The above process of the EMFC algorithm is described in Algorithm 3.2.

---

**Algorithm 3.2: EMFC**

---

**Input**: Topology $G(t_n)$, mobility information $MI(t_n)$, neighbor relationship parameters: LG, HI, LQ, ND, RE, and $N_{max}$.
**Output:** Leader CH set in $G(t_n)$ and their associated follower as cluster formation
**Step 1:** PI calculation using fuzzy logic to select CH as local leader.
1. **for** each round ***do***
2. **for** each $u_i \in U$ with one-hop neighbor $u_j \in N(u_i)$ **do**
3: Received hello packets from $u_j \in N(u_i)$ and extract it.
4: **Get max** $(LG_{u_{ij}}, LQ_{u_{ij}}, ND_{u_i}, RE_{u_i})$ within $\left[u_j \in \{N(u_i) \cup u_i\}\right]$
5: **Get min** $(ND_{u_j}, RE_{u_j})$ within $\left[u_j \in \{N(u_i) \cup u_i\}\right]$
6: Mapping crisp inputs $\left[NI_{u_{ij}}, ND_{u_i}, RE_{u_i}\right]$ to fuzzy membership function using Equation (3.24)−(3.27) and Table 3.2
7: $PI \leftarrow null$ // Initialization of aggregated fuzzy output PI
8: **for** $N \leftarrow 1$ to $3^3$ **do**
9: Evaluate input states using fuzzy rules $evaluate(R_N)$ given in Table 3.3 using fuzzy MIN-MAX method
10: $PI \leftarrow PI \cup evaluate(R_N)$ // Aggregate the fuzzy output PI
11: **end for**
12: Calculate crisp output of PI using CoG method
13: Include PI value in hello packet and transmits to $u_j \in N(u_i)$
**Step 2:** Cluster formation via follower CM selection for each CH
14: **if** $\left(PI(u_i) > \left[PI(u_j) \in N(u_i)\right]\right)$ **then**
15: Set $u_i$ as $CH_i$ and construct cluster using $N(u_i)$ UAVs
16: **else**
17: Follow the nearest CH according to the maximum PI in $N(u_i)$
18: **end if**
**Step 3:** Cluster size equalization
19: **if** $(N_{CH_i} > N_{\max})$ **then**
20: Release CMs according to the order of **min** LG value until satisfy $N_{\max}$ constraint
21: Set cluster size restriction *flag==true*
22: **else if** $(N_{CH_i} < N_{\max})$ **then**
23: Borrow CM from neighbor $CH_j$ according to the order of **max** LG until satisfying the $N_{\max}$ constraint
24: Set cluster size restriction *flag==true*
25: **else** // all CH restriction flag==*true*
26: Remaining UAVs declare as self CH
27: **end if**
28: **end for**
29**:** round++ //Go to next round
30**: end for**

---

### 3.3.3 TAQR Learning

The TAQR is a position-based multi-hop routing protocol incorporated with QL, where each CH carrying an ADP act as an RL agent and adaptively learns how to reach the BS to deliver ADPs for further processing by finding an optimal routing path in terms of delays, reliable links, and UAV energy, as explained next.

#### 3.3.3.1 State Exploration for Forwarding Node Selection

We derive an initial state exploration strategy to avoid unnecessary exploration and detours during initial decision-making for selecting the next relay UAV by the source CH or the respective intermediate source by defining a potential forwarding candidates (PFC) set considering few one-hop neighbors of the respective source UAVs according to the distance progress toward the BS. According to Figure 3.6, the source $CH_i$ selects a one-hop PFC set as $PFC_{hop1}\{CH_i\} \in \{CM^1_{CH_i}, CM^2_{CH_i}\}$. If $CH_i$ selects the UAV $CM^1_{CH_i}$ to relay the ADPs as it shows better distance progress toward the BS, the $CM^1_{CH_i}$ has the PFC set represented as $PFC_{hop2}\{CM^1_{CH_i}\} \in \{CM^1_{CH_j}, CM^2_{CH_j}\}$ to reach the destination BS. The PFC sets from the source $CH_i$ to destination BS is defined as $PFC_{hop1}\{CH_i\} = \{CM^i_{CH_i} | d(CM^i_{CH_i}, BS) < d(CH_i, BS)\}$ and $PFC_{hop2}\{CM^i_{CH_i}\} = \{CM^i_{CH_j} | d(CM^i_{CH_j}, BS) < d(CM^i_{CH_i}, BS)\}$, where $d(.)$ represents the distance between the respective source UAV and the BS. To explain the QL model, we denote the respective source UAV as $u_i$, which selects the next relay UAV $u_j \in PFC_{hop1}$ to forward the ADPs to the BS.

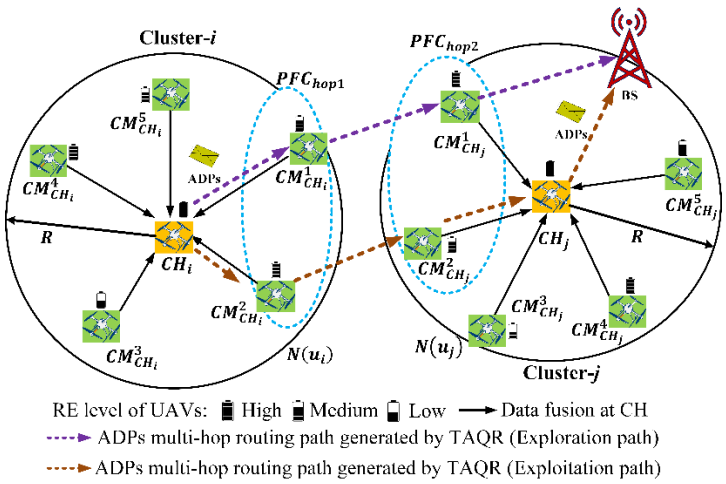



Figure 3.6 Two-phase topology control with CH associated CMs, and PFC sets for respective source UAVs to route ADPs to BS using TAQR with exploration and exploitation paths at different rounds of data transmission.

### 3.3.3.2 Link Delay and PTS for Forwarding Metrics

Considering the limitation of position-based routing protocol as it only seeks the progress in transmission distance, we calculate the $PTS_{u_{ij}}$ for each relaying UAV $u_j \in PCF_{hop1}$, that considers both distance progress and channel condition. The one-hop $delay_{u_{ij}}$ consists of mac delay $t_{ij}^{mac}$ to access the channel and M/M/1 queuing delay $t_{ij}^{que}$. We assume that the packet arrival rate $A_{u_i}$ at UAV $u_i$ follows the poison distribution. The waiting time for the packet to reach the head of the transmission queue is $t_{ij}^{que} = {}^{1}/_{(F_{u_i} - A_{u_i})}$, where $F_{u_i}$ is the forwarding rate [156]. The $delay_{u_{ij}}$ between two UAVs is updated using exponentially weighted moving average and is computed as follows:

$$delay_{u_{ij}} = \beta delay_{u_{ij}} + (1-\beta)\,(t_{ij}^{mac} + t_{ij}^{que}), \tag{3.28}$$

where weighting parameter $\beta \in [0\ 1]$. The $PTT_{u_{ij}}$ for a successful state transition from source $u_i$ to $u_j$ is computed as

$$PTT_{u_{ij}} = delay_{u_{ij}} + \frac{d_{u_{ij}}}{c} + \frac{P_{size}}{R_{u_{ij}}}, \tag{3.29}$$

where $c$ represents the propagation speed equal to the speed of light, and $P_{size}$ represents the size of the packets. From $PTT_{u_{ij}}$ we estimated $PTS_{u_{ij}}$ for link $u_{ij}$ as follows:

$$PTS_{u_{ij}} = \left[\frac{\{d(u_i,BS) - d(u_j,BS)\}}{PTT_{u_{ij}}}\right] > 0. \tag{3.30}$$

A value of $PTS_{u_{ij}} > 0$ indicates that the relay UAV shows distance progress toward the BS, and a higher value of $PTS_{u_{ij}}$ accelerates the probability of delivering the ADPs to the next relay UAV within the given deadline $PTT_{u_{ij}}$. Therefore, during exploration, the neighbor UAV $u_j \in PFC_{hop1}$, which offers a maximum $PTS_{u_{ij}} > 0$ satisfying the condition $LG_{u_{ij}} \geq PTT_{u_{ij}}$, is included in the $u_j \in PFC_{hop1}$ set to relay the ADPs toward the BS.

### 3.3.3.3 Multi Objective Reward Function

The source UAV $u_i$ evaluates its action as a relay UAV $u_j \in PFC_{hop1}$ selection, by using the multi-objective reward $r_{u_{ij}}$ to discover optimal routing paths to avoid congestion, link breakages, and energy holes. The first component of the $r_{u_{ij}}$ is $PTT_{u_{ij}}$, which helps to avoid highly congested path. The relay link $u_{ij}$having less $PTT_{u_{ij}}$provide less delay. Hence, we use the negative exponential in the first component of $r_{u_{ij}}$.

Frequent link breakages cause more retransmissions in FANETs. Owing to sudden change in relative mobility, the neighbor UAV $u_j$ may leave the communication range $R$ of the respective source UAV $u_i$ within the intermediate time of neighbor upgrade or even in the middle of data transmissions. Thus, to ensure better path stability, a UAV $u_j \in PFC_{hop1}$ is considered to be a relay that gives a higher LG given by (3.20) and a better reliable LQ given by (3.23). By blending these two parameters, we obtain $NI_{u_{ij}}$ which is the second component of $r_{u_{ij}}$.

The UAV $u_j \in PFC_{hop1}$ having more $RE_{u_j}$ in proportion to its initial energy $E_{ini}$ is more eligible to be the next forwarding node with respect to the current source UAV $u_i$ to equalize energy consumption. The energy-related cost $E_j$ is the third component of $r_{u_{ij}}$. Finally, $r_{u_{ij}}$ is computed as follows:

$$r_{u_{ij}} = \frac{1}{3}\left[A_1 e^{-PTT_{u_{ij}}} + A_2 NI_{u_{ij}} + A_3 E_j\right] = \frac{1}{3}\left[A_1 e^{-PTT_{u_{ij}}} + A_2 \frac{LG_{u_{ij}} \times LQ_{u_{ij}}}{\max\limits_{u_j \in PFC_{hop1}} LG_{u_{ij}} \times LQ_{u_{ij}}} + A_3 \frac{\frac{RE_{u_j}}{E_{ini}}}{\max\limits_{u_j \in PFC_{hop1}} \frac{RE_{u_j}}{E_{ini}}}\right], \tag{3.31}$$

where $A_1 + A_2 + A_3 = 1$ is the weighting parameter. If the next node is the BS, we allocate the maximum reward $r_{\max}$ to the link $u_{ij}$ as given in (3.32). If the taken action stuck in local optima (routing holes), meaning that the selected relay UAV $u_j$, shows distance progress to the BS but there is no potential neighbor UAV to forward further or even if it takes longer PTT for ADPs to reach the BS, we allocate minimum reward $r_{\min}$ to that relay UAV. Otherwise, when UAV $u_j$ works as a relay toward the BS, each action is evaluated by $r_{u_{ij}}$ given in (3.31). Additionally, if the relay UAV $u_j$ does not send an ACK to the source UAV $u_i$, it will consider the failure state and give a penalty $r_{\min}$ to that link. Therefore, the final reward $r_{u_{ij}}$ for updating the Q-value is computed as follows:

$$r_{u_{ij}} = \begin{cases} r_{\max} = 100, & if\ link\ u_{ij}\ lead\ to\ the\ BS \\ r_{\min} = -100, & if\ link\ u_{ij}\ is\ local\ minimum, \\ 100 \times r_{u_{ij}}, & otherwise \end{cases} \tag{3.32}$$

### 3.7.3.4 Adaptive Q-learning Parameters

As discussed in Section 3.2.3, $\alpha_{u_{ij}}$ and $\lambda_{u_{ij}}$should be controlled adaptively to generate a stable Q-value, considering the frequent topological changes. We update $\alpha_{u_{ij}} \in [0\ 1]$ for link $u_{ij}$ according to the exponential of the normalized one-hop $delay_{u_{ij}}$ as follows:

$$\alpha_{u_{ij}} = \begin{cases} 1 - \exp\left[-\left(\frac{\left\| delay_{u_{ij}} - m_{u_{ij}} \right\|}{\mu_{u_{ij}}}\right)\right], & \mu_{u_{ij}} \neq 0 \\ 0.3, & \mu_{u_{ij}} = 0 \end{cases}, \tag{3.33}$$

where $m_{u_{ij}}$ and $\mu_{u_{ij}}$ represent the mean and variance of the $delay_{u_{ij}}$ computed in (3.28). According to (3.33), if $delay_{u_{ij}}$ is higher, $\alpha_{u_{ij}}$ is larger to update the Q-value faster.

A higher $\lambda_{u_{ij}}$ value defines the stability of the expected future Q-value, and a lower $\lambda_{u_{ij}}$ gives a vulnerable Q-value expectation. As we aim to find a stable reliable link $u_{ij}$, we adaptively adjust the value of $\lambda_{u_{ij}} \in [0, 1]$ for link $u_{ij}$ according to mobility, more specifically the relative distance $d_{u_{ij}}$ intimacy with the neighboring UAVs as follows:

$$\lambda_{u_{ij}} = \begin{cases} 1 - \frac{\left\| R_{sd} - d_{u_{ij}} \right\|}{R_{sd}}, & if\ \ 0 \leq d_{u_{ij}} \leq R_{sd} \\ 1 - \frac{d_{u_{ij}}}{R}, & if\ \ R_{sd} < d_{u_{ij}} \leq R \end{cases}. \tag{3.34}$$

According to (3.34), the value of $\lambda_{u_{ij}}$ decreases when $d_{u_{ij}} < R_{sd}$, and it is maximized when $d_{u_{ij}} = R_{sd}$. Then, $\lambda_{u_{ij}}$ decreases proportionally with an increasing $d_{u_{ij}}$ and becomes zero when $d_{u_{ij}} = R$. After obtaining $\alpha_{u_{ij}}$, $\lambda_{u_{ij}}$, and $r_{u_{ij}}$, we update the Q-value using (3.6) for the corresponding link $u_{ij}$.

### 3.3.3.4 Routing Decision and Balance Between Exploration and Exploitation

Exploration is the discovery of a new state for ADPs that may provide a better reward than experience. Exploitation takes the best action according to the maximum Q-value of the corresponding link, and it helps reach the global optima. However, during exploration, the actions taken can be good or bad. Hence, excessive exploration can generate unnecessary detours. Therefore, in FANET QL-based routing decision-making, an exploration strategy is required to obtain a new state for relaying that may provide better routing paths.

Each UAV $u_i$ adaptively decides either to perform exploration or exploitation according to the value of ANI denoted as $ANI_{u_i}$. $ANI_{u_i}$ is computed based on the $NI_{u_{ij}}$

given by (3.24) to meet the balance between exploration and exploitation. The $ANI_{u_i}$ is computed as follows:

$$ANI_{u_i} = \frac{\sum_{u_j \in N(u_i)} NI_{u_{ij}}}{|N(u_i)|} < NI_{th}, \tag{3.35}$$

If the $ANI_{u_i}$ is less than the threshold value $NI_{th}$ that is set to 0.9 in our study, the UAV decides to explore and, instead of taking random action, the neighboring UAVs (whose $PTS_{u_{ij}} > 0$ and $LG_{u_{ij}} \geq PTT_{u_{ij}}$) are never selected as relay UAVs and are included in the $PFC_{hop1}$ to explore a new state. If the $ANI_{u_i} > NI_{th}$, it means the neighboring state is relatively stable. Thus, UAV decides to perform exploitation. The source UAV $u_i$ selects the neighbor UAV $u_j \in PFC_{hop1}$, that offer the maximum Q-value stratifying the constraint $LG_{u_{ij}} \geq PTT_{u_{ij}}$. When the source UAVs have less NI and hardly meet the imposed LG constraints $LG_{u_{ij}} \geq PTT_{u_{ij}}$ with neighboring UAVs, the TAQR can trigger the topology adjustment operation to adjust the weight of the attractive force in (3.12) to maintain the path stability by controlling the relative velocity and LG with one of the neighboring UAVs to forward toward the BS.

To avoid the routing loop during relay UAV selection, each source UAV must not consider any UAV that has been previously considered in the end-to-end path to the BS. During each state transition, the updated Q-value is continually tracked against previously visited UAVs so that none of the forwarding UAVs is selected more than once. Additionally, the penalty $r_{\min}$ value in the reward function (3.32) helps avoid unnecessary detours of ADPs.

The above process is described in Algorithm 3.3. Lines 15−33 represents our proposed state exploration strategy to route ADPs toward BS according to neighbor state stability condition, $ANI_{u_i} < NI_{th}$. It includes the topology adjustment triggering method to improve the neighbor intimacy to meet the condition, $LG_{u_{ij}} \geq PTT_{u_{ij}}$ (Lines 25−28). It also includes the penalty mechanism if the forwarding UAVs fail to meet the condition $PTS_{u_{ij}} > 0$, to avoid the routing holes (Lines 30−33). Lines 34−38 represent the exploitation strategy based on the maximum Q-value found in $u_j \in PFC_{hop1}$ set.

---

**Algorithm 3.3:** TAQR

---

**Input:** FANET topology generated by VFMC and EMFC, $NI_{th}$
**Output:** Leader CH UAVs transmit ADPs to the BS
1: **Proceed** to next time slot $t_{n+1}$
2: $Q - value = PTT_{u_{ij}} = PTS_{u_{ij}} = 0$ // Initialization
// **Phase-1 FANET Topology Control:** Topology $G(t_n)$ and *MI* detection
3: **for** each UAV $u_i \in U$ **do**
4: Call algorithm 1
5: **end for**
// **Phase-2 FANET Topology Control:** Local leader CH and follower CM selection
6: **for** each UAV $u_i \in U$ **do**
7: Call algorithm 2
8: CH collect data from CM and prepare ADPs
9: **end for**
// **Routing decision using Q-learning:** Each CH carrying the ADPs act as source and other UAV act as relay to transmit ADPs to BS //
10: **while** ADPs need to transmit **do**
11: **if** $(d(u_i, BS) \leq R)$ **then** //if source UAV within the communication range of BS
12: Transmit the data to BS and allocate maximum reward
13: **else**
14: Make routing decisions based on Q-learning
15: **if** $\left(ANI_{u_i} < NI_{th}\right)$ **then** //exploration
16: **for** each $u_j \in PFC_{hop1}$ of $u_i$ **do**
17: Calculate $PTT_{u_{ij}}$ using (3.29)
18: Calculate $PTS_{u_{ij}}$ using (3.30)
19: **end for**
20: **if** $(PTS_{u_{ij}} > 0)$ **then**
21: **if** $(LG_{u_{ij}} \geq PTT_{u_{ij}})$ **then**
22: Update $PFC_{hop1} \leftarrow (u_j \in PFC_{hop1})$ according to the descending order of $PTS_{u_{ij}}$
23: Select relay UAV $u_j \in PFC_{hop1}$ that offer maximum $PTS_{u_{ij}}$
24: Calculate the reward using (3.32) and update the Q-value using (3.33)−(3.34) and (3.6)
25: **else**
26: Trigger topology adjustment to adjust the weight of the attractive force in (12) to satisfy $LG_{u_{ij}} \geq PTT_{u_{ij}}$
27: Select relay UAV $u_j \in PFC_{hop1}$ that satisfy $LG_{u_{ij}} \geq PTT_{u_{ij}}$
28: Calculate the reward using (3.32) and update the Q-value using (3.33)−(3.34) and (3.6)
29: **end if**
30: **else**
31: Trigger penalty mechanism
32: Give minimum reward and update Q-value
33: **end if**
34: **else** // exploitation

35: Select the relay UAV $u_j \in PFC_{hop1}$ with maximum Q-value
36: Calculate the reward using (3.32) and update the Q-value using (3.33)−(3.34) and (3.6)
37: **end if**
38: **end if**
39: **end while**

---

### 3.3.4 Cost and Time Complexity

The three modules of VFMC, EMFC, and TAQR are executed in each UAV using one-hop neighbor information. As a result, the computational cost for one complete round depends on the degree of the UAV during the sequential updates of the FANET topology at each *HI*, as given in (3.21). Thus, the approximate computational cost for each *HI* is $O(2\Delta)$ messages, including ACKs, where $\Delta$ represents the maximum degree of a UAV over each sequential topology update. The time complexity of the VFMC is $O(\Delta M)$, and that of the EMFC is $O(\Delta 27IN) + O(Opt_{CH} X \log Y)$, where 27 is the number of rules in the fuzzy table, *IN* is the number of fuzzy inputs, $Opt_{CH} = {}^{U}/_{N_{\max}}$ represents the optimal number of elected CHs for each round, $X$ is the maximum number of shortage CM UAVs for a CH to become $N_{\max}$, and $Y$ is the minimum number of CM UAVs within a CH that must leave to satisfy the cluster constraint, $N_{CH_i} \leq N_{\max}$. Finally, the time complexity of TAQR is $O(\Delta)$ for state exploration because it requires only one-hop neighbor information.

## 3.4 Performance Evaluation

In this Section, the performance of the proposed JTCR is evaluated via an extensive computer simulation. As JTCR considers the mission and communication performance, we consider the two protocols of MOOC [100] and MCFO [83], which are suitable for comparison with our JTCR. As we adopted the QL-based geographic routing protocol, we also compared the JTCR with the recently proposed QL-based geographic routing protocol QTAR [33], which was proposed specifically to perform surveillance missions. We adopted the implementation environment for the MCFO as a few mission UAVs were set to track the MGTs with a circular trajectory, and some relay UAVs (70 % of the total UAVs) were used to create a relay path with the BS by following the mission UAV's trajectory. For routing, the shortest path was typically considered. MOOC is a clustering protocol that uses attractive and repulsive virtual forces to maximize coverage and maintain connectivity among UAVs. We adopted the MOOC simulation environment by using only the attractive and repulsive virtual forces between UAVs. The MOOC provides a fixed coverage density by hovering without tracking MGTs. For routing decisions, we adopted conventional clustering-based hierarchical routing (i.e., CM to CH, CH to another CH, and CH to BS). We implemented the QTAR environment according to topology construction and multi-hop data routing

proposed in [33]. In the following subsections, the simulation environment and performance metrics are discussed.

### 3.4.1 Simulation Environment

We implemented the proposed JTCR surveillance model using MATLAB with Mamdani fuzzy logic and a reinforcement learning toolbox. The UAVs were uniformly deployed in mission-area *D* with a topology dimension of 2,000 × 2,000 × 100 m to monitor the randomly distributed MGTs at a speed of 5 m/s. We considered the reference point group mobility model (RPGM) and random waypoint (RWP) [157] for the MGTs and UAVs to change their mobility according to the MI of the VFMC at each timeslot. The height of the UAVs varied from 70 to 100 m. The maximum allowable velocity $v_{\max}$ and acceleration $a_{\max}$ for the UAVs was set to 15 m/s and 5 m/s. The maximum communication range of the UAV was set to 250 m, and $R_{sd}$ was set to 100 m. The minimum threshold value for calculating the LG value was set to 2 s. In our simulation, during each timeslot, the data transmission of each UAV is completed in three phases: sensing the MGTs by using onboard sensors, sending the sensed data to the elected CH UAV for aggregation, and relaying the ADPs toward a single location fixed BS. Each data interval round was 10 s, and the total simulation time was $T = 1000$ s. To perform the topology adjustment, we set $\eta_1 = 0.4$ and $\eta_2 = 0.2$. Additionally, we applied the values of $\delta_1 = 55$ and $\delta_2 = 7$. The values of the force constants were set to $K_{DA_i}$=600, $K_{m_k} = 8$, $K_R = 3{,}000$, and $K_A = 1{,}000$. Initially, we set $\sigma = 0.5$ and $HI = 0.5$ s. To generate data traffic, we considered a video streaming application running on each UAV modeled at a constant bitrate (CBR). The complete parameters used in the simulation are listed in Table 3.4.

Table 3.4 Simulation parameters (JTCR).

| Parameter | Value |
|---|---|
| Topology Dimension | 2,000 × 2,000 × 100 *m* |
| Maximum number of UAVs | 100 |
| Number of MGTs | 1300 |
| MGTs mobility model | RPGM and RWP |
| UAV height range | *70–100 m* |
| UAV communication range (*R*) | 250 *m* |
| Separating distance ($R_{sd}$) | 100 *m* |
| Maximum velocity of UAVs | 12 *m/s* |
| Carrier frequency | 2.4 *GHz* |
| UAV transmit power | 5 *mW* |
| SINR threshold for U2U links | 0 *dB* |
| Propagation model | Free space |
| Path loss exponent ($\zeta$) | 3 |

| | |
|---|---|
| MAC protocol | CSMA/CA with TDMA |
| Bandwidth ($B$) | 20 MHz |
| Antenna | Omni-directional |
| UAV initial energy ($E_{ini}$) | $2 \times 10^5$ *Joule (J)* |
| Energy threshold of UAV | 2,000 *J* |
| Traffic type | CBR |
| Traffic load per video streaming | 2 Mbps |
| Transport protocol | User datagram protocol |
| Comparing protocols | MOOC, MCFO, and QTAR |

### 3.4.2 Performance Metrics

We considered performance metrics of two categories: mission and communication performance. The mission-related performance metrics are as follows:

- Tracking coverage rate (TCR): Indicates the ratio between the total number of MGTs uniquely covered by all UAVs divided by the total number of MGTs at each timeslot. The TCR metric is evaluated for different numbers of UAVs and time steps in seconds.

Communication-related performance metrics taken in our study are given below:

- Connectivity rate: Ratio between the number of connected node pairs and possible maximum number of connected node pairs in the topology with the same number of UAVs, which is a connected graph.

- Packet delivery ratio (PDR): The PDR is determined by the number of successfully delivered data packets at the BS and the number of data packets originating from the leader CH UAVs. PDR reflects the data delivery effectiveness of the routing protocol and higher PDR means better performance.

- Average number of retransmissions (ANR): The ANR represents the average number of packets needed to be retransmitted by the source CH UAV, owing to link breakages and congestion. Less ANR means better performance.

- Average end-to-end delay (AE2ED): The average time required for successful data transmission between the source CH and BS is described as AE2ED. Less AE2ED means better performance.

- Control overhead: The size of control packets per hello interval generated by the topology control and routing is defined as the control overhead. The control overhead includes hello packets that contain UAV mobility information, delay, LG, LQ, ND, RE, PI, and Q-value for each neighbor included in the packet header for constructing the FANET topology and routing decisions.

- Clustering stability: Number of CHs, number of isolated CH due to cluster size constraint, and CH lifetime. All the above parameters are observed for the different number of UAVs.

- Normalized Residual Energy (NRE): This metric consists of both the UAV propulsion energy consumption ($PE_{u_i}$) to perform flocking adjustments given in [138] and the energy consumption to perform the data communications ($CE_{u_i}$) given in [56] at each timeslot. NRE for each UAV is normalized as $\frac{E_{ini}-(PE_{u_i}+CE_{u_i})}{E_{ini}}$. NRE is observed after completing the simulation, and higher NRE indicates less energy consumption.

### 3.4.3 Simulation Results and Discussion

In this subsection, the simulation results are compared with the existing protocols and discussed in terms of the above performance metrics.

#### 3.4.3.1 Mission Performance

Because the QTAR is only routing protocol, it was not included in the mission performance evaluation. Figure 3.7 shows the TCR for different numbers of UAVs. As in the proposed JTCR, we assumed that all UAVs performed missions and simultaneously relayed data with the help of an MGT-DF vector, which tracks more MGTs and provides better TCR than others.

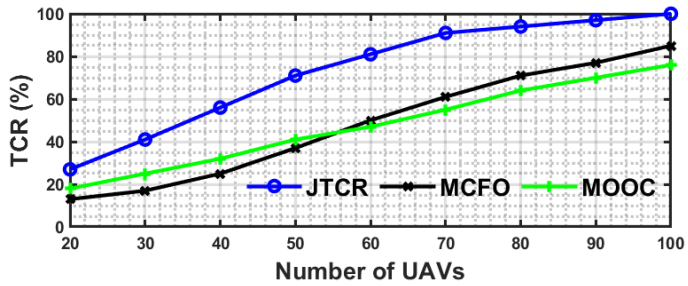



Figure 3.7 TCR for different number of UAVs.

The MGT-DF vector of the VFMC mobility controller updated the UAV position according to the mobility of the MGTs. Initially, MOOC provided a better TCR than MCFO because it maintained a constant coverage density, but with an increasing number of UAVs, MCFO obtained higher mission UAVs to track more MGTs. Thus, when the node density passed 57, the MCFO provided better mission performance than did MOOC.

Figure 3.8 represents the TCR for 80 UAVs at different time steps (seconds). The proposed JTCR outperforms others and TCR increases iteratively due to continuous mobility updates of UAVs to track maximum MGTs using the MGT-DF of VFMC mobility controller. Because MCFO is required to balance the number of mission UAVs and relay UAVs to optimize mission and communication performance, it shows less TCR compared with the JTCR, but it iteratively increases over MOOC. Initially, MOOC provides a better TCR than MCFO because it maintains a fixed coverage of the mission area, but its TCR becomes steady after few iterations as UAVs are not tracking the movement of MGTs.

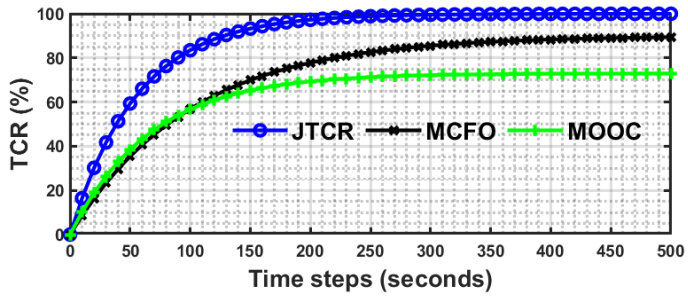



Figure 3.8 Tracking coverage rate (TCR) for different time steps (seconds) with 80 UAVs.

### 3.4.3.2 Communication Performance

In this subsection, we evaluate the JTCR for communication performance metrics with different number of UAVs. Figure 3.9 shows the connectivity rate for different number of UAVs. The proposed JTCR provide a better connectivity rate than others as it uses TFF, which has the attractive and repulsive virtual forces to construct the FANET topology. Simultaneously, it adaptively balances the force weights according to the changes in inter-UAV distance between neighboring UAVs during the mission. The adaptive hello interval also creates a strong relationship with neighbor UAVs. As QTAR does not include the connectivity maintenance mechanism, it is not included here. MOOC offers better connectivity than MCFO because it controls the UAV velocity by applying attractive and repulsive virtual forces. However, with the increased number of UAVs, the connectivity performance of MCFO increases as it obtains more relay UAVs to construct the FANET topology at a reasonable inter-UAV flocking distance.

Figure 3.10 shows the PDR for different number of UAVs. The proposed JTCR provide a better PDR than the other methods for two reasons. First, owing to the EMFC clustering concept being used instead of transmitting the sensed data straight away, each UAV selects the leader CH UAV that provides better leadership and better LG and LQ to transmit the sensed data to the BS. The EMFC clustering also offers fewer MAC contention delay, owing to the equal cluster size during data aggregation at the CH UAV. Second, the CH UAV forwards data to the ADPs by selecting the forwarding UAV that offers a higher PTS satisfying the LG constraint. Moreover, as shown in Figure 3.11, the JTCR requires far fewer retransmissions compared than others. This is a vital requirement for achieving high PDR performance.

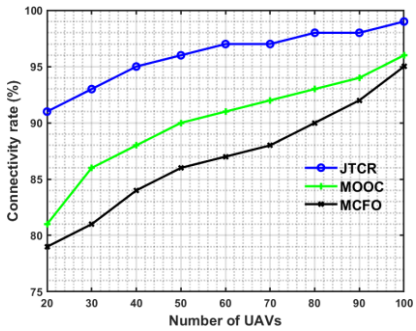



Figure 3.9 Connectivity rate for the different number of UAVs.

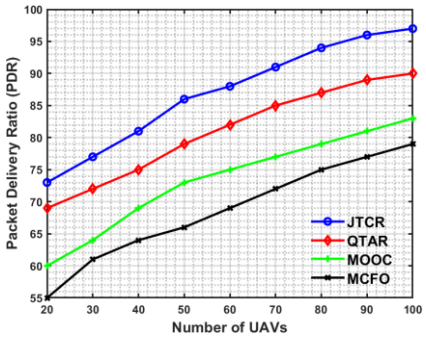



Figure 3.10 PDR for the different number of UAVs.

Figure 3.11 shows the ANR performance for different number of UAVs. The proposed JTCR requires fewer retransmissions than others, as the initial topology construction creates a strong neighbor relationship among UAVs by maximizing the LG with neighboring UAVs to avoid frequent link breakages while performing the mission. In addition, during EMFC data aggregation, the CH UAV with higher stability and LG is considered. Similarly, during relay UAV selection, the CH UAV selects the UAV offering a higher neighbor intimacy in terms of LG and LQ. The proposed JTCR can avoid link breakage, owing to its adaptive

hello interval that adjusts to the minimum LG found within its one-hop vicinity to refresh the neighbor list immediately according to the degree of topological changes. The QTAR always selects the forwarding UAV that offers a higher PTS without controlling the relative velocity. Hence, more link breakages are encountered. The MCFO provides more retransmissions because it always selects a relay that provides the shortest path toward the destination. As a result, MCFO encounters higher data congestion. MOOC provides fewer retransmissions compared with the MCFO as it controls the UAV's relative velocity to maximize the connectivity duration.

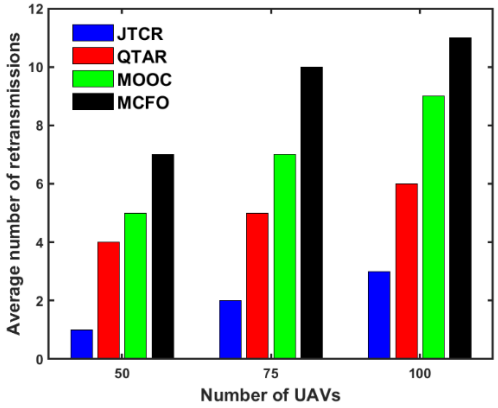



Figure 3.11 Average number of retransmissions (ANR) for the different number of UAVs.

Figure 3.12 represents the AE2ED delay for different number of UAVs, and the JTCR creates less delay compared with others due to three main reasons. First, during data aggregation, each UAV encounters less contention, owing to the equal CMs for each leader CH. Second, during relay UAV selection (state exploration), JTCR selects the relay UAV that offers higher PTS. JTCR precisely calculates the PTT, queuing, transmission, and propagation delay for each neighbor UAV. Third, in the reward function, JTCR estimates the reward jointly considering the one-hop delay and neighbor intimacy (both LG and LQ) that offer better path stability. Although the MCFO selects the shortest path, neither it nor the MOOC considers the MAC-layer assumption of selecting the optimal relay UAV that offers the least delay. However, owing to the clustering data aggregation concept, MOOC provides comparatively less delay than does the MCFO. QTAR considers both MAC and queuing delays. However, owing to each UAV transmission to the BS, the delays are higher than the proposed JTCR.

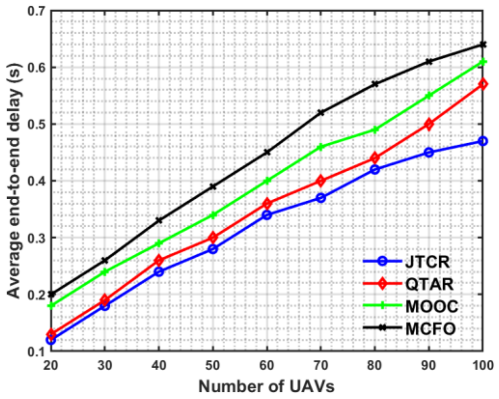



Figure 3.12 Average end-to-end delay (AE2ED) for the different number of UAVs.

Figure 3.13 shows the control overhead size per hello interval for different numbers of UAVs. QTAR gives a very high control overhead compared with others because it retains the two-hop neighbor information. The JTCR produces very less control overhead compared with QTAR because it retains only one-hop neighbor information and optimizes the number of hello packets by controlling the hello interval in terms of minimum LG and hello interval frequency by sensing the topological changes at two different times. Additionally, the adaptive force weight management of TFF in JTCR according to inter-UAV distance maximizes the LG with neighboring UAVs, which helps JTCR to significantly reduce control overhead than QTAR. However, as JTCR must store mobility information, LG, LQ, ND, PI, and Q-value information for each neighbor link, the proposed JTCR provides a slightly higher control overhead than does the MCFO and MOOC. Owing to broadcasting the hello messages at a fixed hello interval, the control overhead is increased almost linearly with the increased number of UAVs in both MOOC and MCFO.

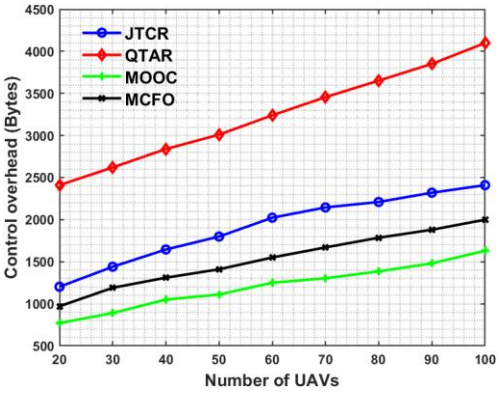



Figure 3.13 Control overhead size per hello interval for the different number of UAVs.

Figure 3.14 presents the NRE of different routing protocols for 100 UAVs. In Figure 3.14, the horizontal red line within each box represents the median of the NRE for each routing protocol. JTCR shows better NRE (less energy consumption) compared to other routing protocols owing to three major reasons. First, the VFMC module in JTCR generates an efficient travel distance for each UAV (to maximize coverage toward MGTs and maintain connectivity with neighbor UAVs). It not only reduces propulsion energy consumption but also produces balance in propulsion energy consumption during flocking adjustments. This is because propulsion energy consumption is proportional to each UAV trajectory, and it is sufficiently larger than the communication energy consumption. Second, to select the data aggregator, the EMFC module in JTCR gives priority to the UAVs having higher RE level. Third, while selecting relay UAV by TAQR module in JTCR, more reward is given to the UAVs having a higher RE. This creates proper load sharing among UAVs to avoid energy holes, resulting in the extended FANET lifetime. Additionally, owing to the data aggregation and our proposed exploration and exploitation strategy, the JTCR requires fewer transmissions than QTAR, which significantly reduces UAV communication energy consumption. Due to the above reasons, if we look at the NRE distribution of UAVs in each box, JTCR produces more balance in energy consumption. The balance in energy consumption gives better node density and topological stability in FANETs. QTAR provides higher energy consumption than the JTCR because it encounters high control overhead and more retransmissions. Both MCFO and MOOC provide less NRE because they do not consider any energy consumption metric during clustering or route selection.

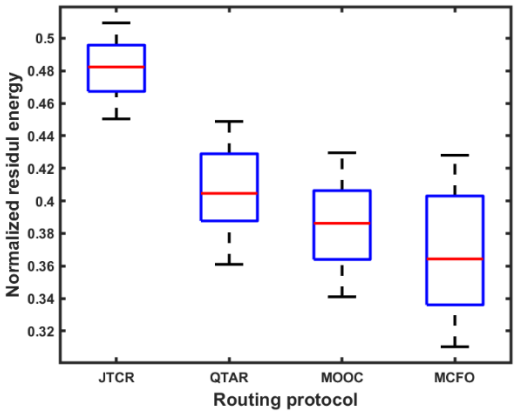



Figure 3.14 Normalized residual energy (NRE) for different routing protocols.

Next, we observe the clustering stability performance of the proposed JTCR with MOOC clustering. Figure 3.15 shows the required number of leader CH UAVs for the different number of UAVs. The EMFC module in JTCR requires more CH than MOOC because MOOC uses a multi-hop clustering in which CMs can join the CH, even if it is away from its one-hop vicinity, usually two-hop members are allowed to join. However, if we observe

the CH lifetime in Figure 3.16, the EMFC clustering of JTCR provides much better CH lifetime and stability in the FANET topology. This is because, during leader CH selection, JTCR jointly considers the NI, the leadership factor (ND), and the RE level of UAVs.

According to Figure 3.17, the JTCR provides a smaller number of isolated CHs in the clustering process compared to MOOC. It is because the EMFC clustering module in JTCR performs the cluster size equalization under the constraint of the maximum cluster size. Higher CH lifetime and the less number of isolated CHs also improve the performance during data aggregation, create load balance in inter-cluster routing, and mitigate the routing delay by adopting the QL-based optimization. In Figure 3.17, for our JTCR, we observe little variation (as the number of isolated CHs is not large) in the number of isolated CHs for different number of UAVs. This is mainly because of both the cluster size equalization under the constraint of the maximum cluster size and the distribution of UAVs distribution within the mission area. MOOC considers multi-hop CMs by only controlling the velocity of neighboring UAVs, and the RE level of CH UAVs are not taken into consideration. MOOC allows the follower CM to follow the leader CH away from its one-hop neighbors without considering the RE levels of the leader CH. Such policy in MOOC clustering produces uncertainty in the stability of the FANET topology because UAVs may leave the network for energy replenishment if they reach the threshold energy levels.

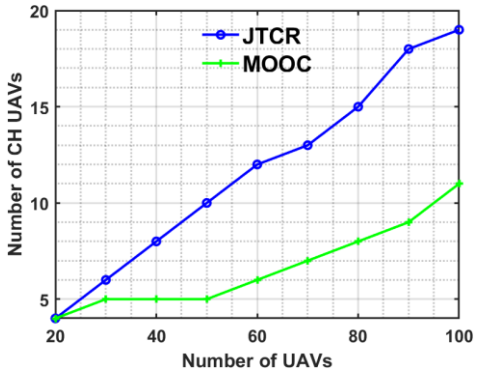



Figure 3.15 Number of CH UAVs versus the number of UAVs.

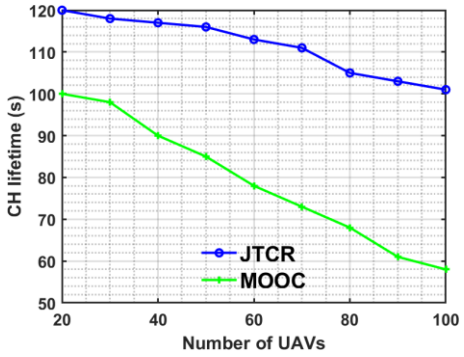



Figure 3.16. CH lifetime versus the number of UAVs.

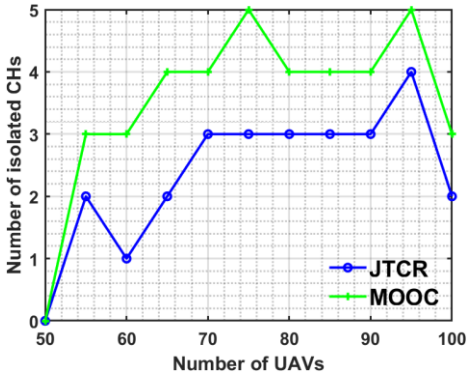



Figure 3.17 Number of isolated CHs versus the number of UAVs.

As the proposed JTCR utilizes the RL algorithm to make inter-cluster routing decisions using the TAQR module, a key concern is QL convergence. We next discuss the convergence of our TAQR inter-cluster multi-hop routing compared with QTAR. Considering the topological changes in FANETs to support adaptive learning, QTAR updates the discount factor for each neighbor link according to the degree of change in the neighboring set at two different times, which may not provide appropriate link conditions with each neighbor UAV. In contrast, the proposed TAQR updates the discount factor according to the relative distance. This provides a proper assessment of each neighbor link SINR level and produces a more precise Q-value by giving a higher discount to the links that satisfy the imposed flocking constraints (the minimum separating distance and the maximum transmission range constraint to support desired SINR).

The TAQR achieves better topology and produces suboptimal actions very quickly because a two-phased topology control is designed to provide a stable topology for each timeslot at a reasonable cost in control overhead by considering both mission and communication performance. In QTAR, each UAV stores the mobility information, delay,

and Q-value for all two-hop neighbors. As a result, owing to the extended topological knowledge, it converges slightly earlier than TAQR, as shown in Figure 3.18. Because QTAR does not consider the exploration and exploitation policy of QL, it converged with fewer rewards. In contrast, owing to the exploration strategy based on the ANI and the relation between PTS, PTT, and LG with neighbor UAVs, the TAQR in JTCR provides better average reward via exploring the new state, as shown in Figure 3.18. Because the VFMC mobility controller in JTCR controls the relative distance, the LG between neighboring UAVs is also guaranteed with a reasonable cost in control overhead at each timeslot.

Compared with the QTAR, the TAQR in JTCR is more intelligent when making routing decisions. During exploration, it selects the relay UAV that offers a higher PTS satisfying the LG constraint and ensures a longer survival time of the selected neighbor link to complete the data transmission within the given PTT. It avoids the routing loop by storing the previously visited UAV in an end-to-end path. It can also trigger the topology adjustment process to adaptively adjust the weight of the TFF if the neighbors listed in the PFC set are too far away or if they fall within an interference zone.

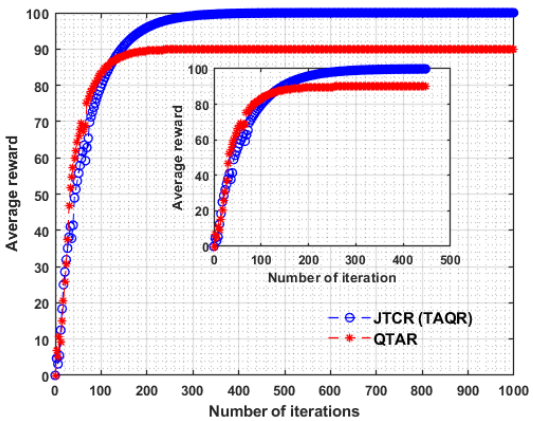



Figure 3.18 Average reward versus the number of iterations.

### 3.4.3.3 Comparative Summary

Based on our discussion in Section 3.4.3, in this subsection, we briefly discuss the mission and communication performance improvements of our proposed JTCR compared to other routing protocols. We observe that our proposed JTCR gives 34.95% and 33.74% better TCR compared to MOOC and MCFO, respectively, as mission performance. The EMFC module in JTCR gives 29.18% better CH lifetime compared to MOOC.

JTCR gives 6.14% and 9.50% better connectivity rate than MOOC and MCFO, respectively. JTCR exhibits 7.03%, 15.58%, and 21.58% better PDR performance compared to QTAR, MOOC, and MCFO, respectively. Also, JTCR provides 9.75 %, 24.04%, and 38.32% less AE2ED compared to QTAR, MOOC, and MCFO, respectively. In the case of NRE (remaining energy), JTCR exhibits 15.72%, 19.79%, and 23.46% better NRE (less energy consumption) compared to QTAR, MOOC, and MCFO, respectively. JTCR shows 65.70% less control overhead than QTAR, which is a significant reduction in control overhead while improving the communication performance. However, JTCR provides a slightly higher control overhead (30.79% and 19.94%) compared to MOOC and MCFO, respectively, because the proposed JTCR improves both mission performance and communication performance simultaneously. Nevertheless, such a slight increment in control overhead can be acceptable, thanks to the performance improvements of both mission and communication.

## 3.5 Conclusion

In this study, we jointly investigated the relationship between MAC, topology control, and routing policy to efficiently perform crowd surveillance operations using a UAV swarm. A two-phase topology control balances the requirement between mission and communication performance to meet the trade-off between coverage and aerial connectivity. It also offers a stable FANET topology at each timeslot to the routing protocol for forwarding the sensed data to the BS. Thus, it provides better PDR, fewer retransmissions, and less end-to-end delay. It also produces a balance in the energy consumption of UAVs and extends the lifetime of the FANET with reasonable control overhead. In TAQR, the strategy of exploration and exploitation helps to avoid local optima in QL and gives a better average reward. Additionally, the adaptive learning in TAQR helps to avoid routing holes, loops, and unexpected link breakages in inter-cluster multi-hop routing. Because our objective was to detect the maximum number of MGTs and transmit the sensed data to the BS, we slowly controlled the mobility of UAVs.

# 4. Q-Learning-Based Routing Inspired by Adaptive Flocking Control

## 4.1 Introduction

With the significant development of UAV technology in recent years, UAV swarms have been utilized in many applications including surveillance [100], wildfire monitoring [59], ABS [23], data collection, and providing mobile edge computing services to low-power Internet of things devices. In a UAV swarm also known as FANET, UAVs can collaboratively execute a mission through a formation control algorithm with 3D positioning, by communicating with each other in an ad hoc manner. Significant advancements in sensor and battery technologies, localization techniques based on GPS, and cooperative localization using different ranging methods in GPS-denied environments have enhanced the autonomy of FANETs [56], [95], [158].

In such a UAV swarm, cooperative coordination among UAVs is necessarily required to maximize the coverage and communication performance [159]. Regarding to communication performance, it should maintain a desirable connectivity rate with minimal delay in the UAV-to-UAV (UTU) and UAV-to-base station (UTBS) links. To achieve the above objectives, researchers are designing the self-organized, self-healing, and distributed coordination of multiple UAVs mimicking the properties of SI [50], [76], [101]. Owing to the high mobility in 3D space, time-varying topology, limited energy, fixed transmission range, and the possibility of inter-UAV collisions, FANET topologies are highly dynamic and different from MANETs and VANETs. Moreover, in FANETs, UAVs can arbitrarily leave an aerial network to obtain energy replenishment and thereafter rejoin the network [38]. In MANETs and VANETs, nodes have moderate and high mobilities in 2D space, respectively. However, in VANETs, the mobility of the nodes is constrained by the road, and the nodes are not energy-limited. Owing to these unique properties, the mobility models and routing protocols proposed for MANETs or VANETs are not suitable for direct adoption in FANETs [35]. Mobility models for FANETs should be realistic, autonomous, and mission-driven to achieve high mission and communication performances [83], [159].

To perform a collaborative mission, a mobility model for FANETs should have the following properties. First, each UAV should autonomously maintain a particular separation distance from its neighbor to avoid inter-UAV collisions while simultaneously staying adequately closer to ensure QoS in the UTU links. Second, to preserve collaborative coordination and synchronization of movements, the UAVs should continuously adjust their position, velocity, and flying direction according to the mobility of neighboring UAVs. Third, UAVs in the swarm should be self-healing to establish connectivity during the failure of a neighboring UAV and should be able to arbitrarily leave or join the FANET. Fourth,

the trajectory of each UAV should be smooth, and the moving trajectory of each UAV should maintain fairness in the travel distance to create a balance in energy consumption [29]. Finally, an optimal control overhead should be incurred to predict the updated topology.

The abovementioned properties of collaborative FANETs are similar to the distributed, stable, and self-organized characteristics of biological groups such as flocks of birds and schools of fish [101]. Thus, the large-scale FANET coordination inspired by these behavior-based self-organized swarming flights enhances the effectiveness and simplifies the autonomous distributed coordination of UAV swarms. Cooperative FANETs flying in a dynamic environment can maintain a robust topology through collective motion by adopting the three rules of flocking proposed by Reynolds in 1986 [71]. The three rules are cohesion (attraction), alignment (velocity matching), and separation (repulsion). Each rule produces a motion-component vector, and the weighted sum of the three motion vectors determines the optimal mobility of each UAV for maintaining a connected topology.

In FANETs, owing to the limited transmission power of the UAVs, a direct communication link can only be established within a limited transmission range. Thus, to transmit the data sensed by remote UAVs to the base station (BS), a reliable multi-hop path needs to be established by a series of intermediate relay UAVs. Because of the highly dynamic topology and data routing without proper awareness of the updated topology, FANETs face higher link breakages and blind-path issues. They encounter link breakages if the selected relay UAV leaves the transmission range of the corresponding source UAV in the middle of data transmission. The topology is dynamically changed by the relative mobility and failure of UAVs. Blind path occurs when the neighboring UAV leaves the transmission range of the corresponding source UAV during the topology update [33]. Both phenomena result in high retransmissions and energy consumption. To predict the updated topology, the UAVs exchange hello packets with their neighbors at a particular hello interval. Although a low hello interval provides updated mobility information to the neighboring UAVs, it simultaneously increases the overhead. Thus, to satisfy the trade-off between topology prediction accuracy and overhead, the hello interval must be controlled according to the degree of mobility changes.

In [13], [109], [111], the authors studied traditional topology-aware routing protocols in MANETs, which give slow reaction to a highly dynamic network. Thus, they encounter higher link breakages, delay, overhead, energy holes, routing loops, and blind paths. Tracing the shortest routing path may be good initially, but it cannot be an optimal routing path as it triggers energy holes by depleting a few selected UAVs' energy [62]. Additionally, the shortest paths can be highly congested. A loop-free property is very crucial for FANETs to prevent data packets from being continually routed through similar nodes. Considering the 3D dynamic topology, high overhead, and possibility of inter-UAV collision, position-based

routing protocols have attracted the attention of researchers [10]. Nevertheless, the position-based routing protocols encounter several challenges in FANETs such as maintaining the link quality with increased transmission distance and avoiding the link breakages [10]. Other challenges exist, such as controlling the hello interval to predict the updated topology, localization error, routing holes, routing loops, and energy holes. Routing holes in position-based routing are a local minimum case in which the forwarding UAV has no neighboring UAVs within its transmission range to forward data packets toward the destination.

Recently, RL has been widely applied to optimize wireless network communication performance [2]. Through sequential actions by interacting with dynamic environments and utilizing previous experiences, RL agents can make wiser decisions to maximize the reward. In FANETs, RL is applied in many application scenarios, such as network topology prediction, channel estimation, joint optimization of the UAV's trajectory and communication [160], and data routing [62]. QL is a model-free off-policy value-based RL that is suitable for performing multi-objective optimization in resource-constrained FANETs. QL evaluates the expected value of the cumulative reward and obtains the instant optimal policy based on historical experience, even in an unknown environment without a central controller.

In dynamic FANETs, the link quality of multi-hop paths depends on several parameters such as node density, link signal-to-interference-plus-noise ratio (SINR), delay, relative mobility, and RE of relay UAVs. Thus, to jointly address the above issues, researchers have designed a multi-objective reward function in adaptive QL to select the optimal relay node for forwarding data [33], [62]. This joint consideration of multiple objectives significantly improves the PDR, end-to-end delay, and energy consumption [62]. Additionally, QL can be trained to identify the link that is trapped in the local minimum in position-based forwarding by providing a minimum reward [119]. However, the QL model results in high retransmissions and detours. This is mainly due to the insufficient training samples, the strategy of exploration and exploitation, and the random relay node selection.

In FANETs, while selecting a relay, it is important to find a stable path that ensures sufficient LD for reliable data transmission. The relative mobility prediction metric LD defines a predictable time at which two neighboring UAVs stay within their transmission range [31], [34]. Thus, LD is a function of the inter-UAV distance, relative velocity, flying direction, and UAV transmission range. In a multi-hop path, the minimum LD between two adjacent nodes defines the lifetime of that path. Thus, if there are multiple paths to reach the destination from a particular source, the maximum of the minimum LD along with these multi-hop paths yields the best stable path. Additionally, the consideration of the link delay and RE of intermediate relay UAVs significantly improves the routing performance. Mobility estimation and stable 3D LD can be precisely calculated using a designed flocking

controller. Subsequently, the routing module utilizes the LD to obtain a stable multi-hop path because the relative trajectory knowledge and link stability are highly coupled in a dynamic topology.

The contributions of this study are two-fold. First, a robust distributed mobility model for FANETs is proposed to perform a collaborative UAV swarm mission inspired by behavior-based flocking control. Afterward, the relationship between the flocking-based mobility model and routing is studied to develop a novel routing protocol. The specific contributions of this study are as follows:

- Adaptive flocking control algorithm (AFCA): In AFCA, each UAV utilizes the mobility information of its two-hop neighbor to extend its local view and produce cohesion, alignment, and separation flocking rules. Owing to the wider knowledge of the time-varying topology, AFCA provides faster swarm cohesion.

  To deal with the dynamic topology, AFCA adaptively adjusts the weights of the flocking rules according to the network condition to maximize coverage under the aerial connectivity constraint. The adaptive adjustment of the weights of the flocking rules facilitates the maintenance of optimal node density in FANETs, which provides better SINR, link stability, and inter-UAV collision avoidance. Additionally, AFCA ensures fairness in the travel distance of each UAV, thereby balancing the energy consumption of the UAVs.

  To address the trade-off between topology prediction accuracy and control overhead, AFCA adaptively controls the hello interval for each UAV according to the degree of mobility changes within the neighboring UAVs, defined by the one-hop minimum LD.

- Q-learning-based routing protocol inspired by flocking control (QRIFC): A new multi-objective reward function for QL is designed to minimize delay, energy consumption, and stable path selection using the maximum–minimum 3D predictive LD up to a two-hop neighbor. This strategy extends the local view of each UAV to select a more stable path.

  QL requires exploration before converging to an optimal route. The uncertainties during exploration lead to unnecessary detours, resulting in a higher number of retransmissions and energy consumption. To address this issue, a new state exploration and exploitation strategy for FANETs is proposed on the basis of the relationship between the normalized average link duration (NALD), packet travel time (PTT), and packet travel speed (PTS). This strategy provides adaptability to QL to cope with the dynamic topology and accelerate the QL convergence to the optimal route. It outperforms existing routing protocols, UCB, and $\in$-greedy-based exploration and exploitation strategies and achieves a better average reward.

The penalty mechanism in QRIFC helps to avoid routing holes and loops. QRIFC can trigger topology adjustment (TA) to adaptively adjust the weight of the flocking rules if it detects higher PTT and link breakages.

## 4.2 System Model

There are a set of $N$ quadrotor UAVs $U = \{u_1, u_2, \cdots, u_N\}$ equipped with GPS, IMU, and wireless interface in this study. UAVs are deployed within a 3D mission area to perform a collaborative surveillance mission, as shown in Figure 4.1. The dimensions (length, width, and height) of the mission area are bounded by $(x_{\min} \le x \le x_{\max}, y_{\min} \le y \le y_{\max}, z_{\min} \le z \le z_{\max})$. The entire surveillance mission time $T = \{t_1, t_2, \cdots, t_{n-1}, t_n\}$ is divided into $t_n$ timeslots, and the length of each timeslot is a constant $\tau$. It is assumed that $\tau$ is sufficiently small, and within this time, the mobility of each UAV is fixed. Thus, the FANET topology can be represented as a time-dependent undirected graph, $G(t_n) = (V(t_n), E(t_n))$, where $V(t_n) \in \{U(t_n) \cup BS\}$ represents the vertex set consisting of $U(t_n)$ UAVs and a single localization-fixed BS. The BS is considered as a ground vehicle, which is the data collection and mission control center. The BS can serve as an edge-computing server to perform computationally intensive tasks for the UAVs. In addition, each UAV is aware of the locations of the BS, $\vec{p}_{BS}$.

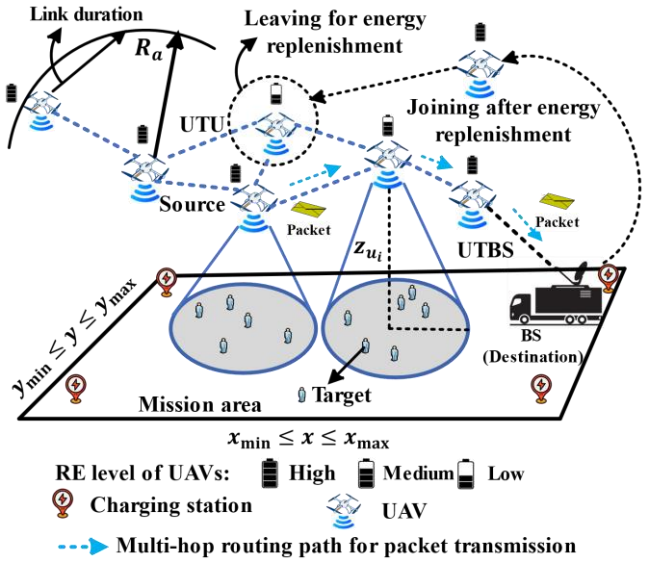



Figure 4.1 An example of collaborative UAV swarm mission.

At each $t_n$, each UAV sends important sensing data to the BS through a multi-hop routing path, as illustrated in Figure 4.1. The RE level of the UAVs is divided into three

categories: high, medium, and low. At each $t_n$,, according to the RE level, the UAVs enter a charging scheduling process to leave the aerial network for energy replenishment at the wireless charging station and thereafter rejoin the network, as shown in Figure 4.1. The transmission range of each UAV is divided into two regions: the repulsion range $R_r$ and attraction range $R_a$. Thus, to meet the safety distance and transmission range constraints, the inter-UAV distance $d_{u_{ij}}(t_n)$ must be maintained within $R_r \leq d_{u_{ij}}(t_n) \leq R_a$. If $d_{u_{ij}}(t_n) \leq R_a$, a direct edge $E(t_n)$ between two UAVs is present. Each UAV $u_i$ uses an onboard GPS to localize its 3D position $\vec{p}_{u_i}(t_n) \in (x_i, y_i, z_i)$. To avoid external obstacles, the height of UAVs is maintained within $z_{u_i}(t_n) \in [z_{\min}, z_{\max}]$. The channel, delay, energy, mobility, routing model, and problem formulation for the proposed QRIFC will be derived in the following subsections. The key notations used in this study are listed in Table 4.1.

Table 4.1 Key notations used in this study (QRIFC).

| Notation | Description |
|---|---|
| $T = \{t_1, \cdots, t_n\}$ | Operation time *T* divided into $t_n$ timeslots |
| $\tau$ | Length of each timeslot |
| $U = \{u_i, \cdots, u_N\}$ | Set of *N* UAVs and $u_i$ is index of each UAV |
| $\vec{p}_{u_i}(t_n)$ | 3D Position vector of UAV $u_i$ at timeslot $t_n$ |
| $\vec{p}_{BS}$ | 2D Position of the fixed BS |
| $\vec{v}_{u_i}(t_n)$ | Velocity of UAV $u_i$ at timeslot $t_n$ |
| $R_r$ | Repulsion range |
| $R_a$ | Attraction range |
| $N_{u_i}^1(t_n)$ | One-hop neighbor set |
| $N_{u_i}^a(t_n)$ | One-hop neighbor found in attraction range |
| $N_{u_i}^r(t_n)$ | UAV in proximity of repulsion range |
| $N_{u_i}^2(t_n)$ | Two-hop neighbor set |
| $d_{u_{ij}}(t_n)$ | Distance between two neighboring UAVs |
| $PTT_{u_{ij}}$ | One-hop packet travel time |
| $PTS_{u_{ij}}$ | One-hop packet travel speed |
| $PTT_{u_{i\to k}}$ | Packet travel time of two-hop neighbors |
| $PTS_{u_{i\to k}}$ | Packet travel speed of two-hop neighbors |
| $RE_{u_i}(t_n)$ | Residual energy (RE) of a UAV $u_i$ |
| $LD_{u_{ij}}$ | Link duration (LD) between two adjacent UAVs |
| $u_j \in PR_{1-hop}$ | One-hop potential relay of UAV $u_i$ |
| $u_k \in PR_{2-hop}$ | Two-hop potential relay of UAV $u_i$ |

### 4.2.1 Channel Model

Owing to the 3D mobility, wireless communication channels between high-altitude UAVs (UTU links) and UTBS links are dominated by line-of-sight (LoS). Thus, the channel power gain $\mathcal{G}_{ij}(t_n)$ between a source UAV $u_i$ and a receiver UAV $u_j$ or a BS in free space is considered as $\mathcal{G}_{ij}(t_n) = \rho_o d_{u_{ij}}^{-\zeta}$, where $\rho_0$ represents the LoS channel power gain at a particular reference distance, that is, $\rho_o = 1\ m$, and $\zeta$ is the path-loss exponent [35]. For a given transmission power $P_{u_i}^{tx}$ of UAV $u_i$, the SINR $\gamma_{ij}(t_n)$ at UAV $u_j$ can be estimated as follows [99]:

$$\gamma_{ij}(t_n) = 10\log\frac{\mathcal{G}_{ij}(t_n)P_{u_i}^{tx}}{I_{ij}(t_n)+\sigma^2(t)} = 10\log\frac{\rho_o d_{u_{ij}}^{-\zeta}P_{u_i}^{tx}}{\sum_{\hbar\in\mathcal{I}_i}\phi\mathcal{G}_{\hbar j}(t_n)P_{tx}^{\hbar}+\sigma^2(t_n)}, \tag{4.1}$$

where $I_{ij}$ represents the interference, $\mathcal{I}_i \in U$ represents the set of $\hbar \neq i,j$ active neighboring UAVs simultaneously broadcasting and $\sigma^2(t_n)$ represents the additive white Gaussian noise power. Here, $\phi$ is a binary variable, whose value turns into 1 if UAV $u_i$ detects a simultaneous transmission within its one-hop neighborhood, otherwise its value set to zero.

UTU links are established successfully if $\gamma_{ij}(t_n) \geq \gamma_{th}$, where $\gamma_{th}$ is a predefined SINR threshold. Hence, the maximum communication range for UAV $u_i$ to communicate with UAV $u_j$ is $d_{u_{ij}}(t_n) \leq d_{u_{ij}}^{th} = \left[\frac{\rho_o P_{u_i}^{tx}}{\left(I_{ij}(t_n)+\sigma^2(t_n)\right)10^{\frac{\gamma_{th}}{10}}}\right]^{1/\zeta}$. Each UAV is equipped with an omnidirectional antenna, and the maximum communication range of each UAV can be denoted as a sphere with radius $R_a = d_{u_{ij}}^{th}$. For the system bandwidth B, the transmitted data rate $C_{u_{ij}}(t_n)$ at UAV $u_i$ is estimated as $C_{u_{ij}}(t_n) = B\log_2[1+\gamma_{ij}(t_n)]$. For a given $\gamma_{ij}(t)$, the per-hop packet error rate $PER_{ij}$ on link $u_{ij}$ is estimated as follows [152]:

$$PER_{ij}(\gamma_{ij}(t_n)) \approx \begin{cases} a_n \exp\left(-\mathcal{g}_n\gamma_{ij}(t_n)\right), & \gamma_{ij}(t_n) \geq \gamma_{th} \\ 1, & \gamma_{ij}(t_n) < \gamma_{th} \end{cases}, \tag{4.2}$$

where $n$ is the transmission mode index. $a_n$ and $\mathcal{g}_n$ are transmission-mode-related parameters whose values are mentioned in [161].

### 4.2.2 Delay Model

In FANETs, the data transmission delay $t_{u_{ij}}$ for link $u_{ij}$ consists of one-hop MAC delay $t_{u_{ij}}^{mac}$, queuing delay $t_{u_{ij}}^{que}$, propagation delay $t_{u_{ij}}^{pg}$, and transmission delay $t_{u_{ij}}^{tx}$. Thus, $t_{u_{ij}} = t_{u_{ij}}^{mac} + t_{u_{ij}}^{que} + t_{u_{ij}}^{pg} + t_{u_{ij}}^{tx}$. The $t_{u_{ij}}^{mac}$ represents the contention delay for the source

node to access the medium utilizing a MAC protocol. The queuing delay of each data packet at UAV $u_i$ depends on the packet arrival rate $A_{u_i}$ and forwarding rate $F_{u_i}$. The M/M/1 queuing model is adopted and it is assumed that the packet arrival rate $A_{u_i}$ follows the Poisson distribution. Thus, $t_{u_{ij}}^{que}$ is updated as $t_{u_{ij}}^{que} = {1}/{(F_{u_i} - A_{u_i})}$, where $F_{u_i}$ is the service rate; $t_{u_{ij}}^{pg}$ is computed as $t_{u_{ij}}^{pg} = {d_{u_{ij}}}/{v_p}$, where $v_p$ is the propagation speed; and $t_{u_{ij}}^{tx}$ computed as $t_{u_{ij}}^{tx} = \frac{P_{size}}{C_{u_{ij}}(t_n)}$, where $P_{size}$ represents the packet size. Finally, the window mean exponentially weighted moving average on $t_{u_{ij}}$ is applied to obtain a more accurate delay as $delay_{u_{ij}}(t_n)$. Each UAV $u_i$ maintains a fixed window of length $W$ for each neighbor $u_j \in N_{u_i}^1(t_n)$ to record the $delay_{u_{ij}}(t_n)$ for the last data packets transmitted on the link $u_{ij}$ within the $W$ and, the delay is computed as follows:

$$delay_{u_{ij}}(t_n) = (1-\beta)\frac{\sum_{n=n-W}^{n-1} delay_{u_{ij}}(t_n)}{W} + \beta t_{u_{ij}}, \tag{4.3}$$

where $\beta \in [0\ 1]$ represents the weighting coefficient. Thus, the required PTT on the link $u_{ij}$, $PTT_{u_{ij}}$ for one-hop transmission is computed as follows:

$$PTT_{u_{ij}} = \frac{delay_{u_{ij}}}{\left[1-PER_{ij}\left(\gamma_{ij}(t_n)\right)\right]}. \tag{4.4}$$

### 4.2.3 Energy Model

The energy consumption cost of a quadrotor UAV comprises two major components: propulsion energy and communication energy. The propulsion power $PR_{u_i}$ produces thrust through the UAV rotors to overcome drag forces and gravity to support the mobility of UAVs in air. Similar to [138], the thrust $T_h$ and $PR_{u_i}$ for the quadrotor UAVs is obtained. In practice, $PR_{u_i}$ is much higher than the communication power. According to [138], the $T_h$ generated by each rotor is a function of the UAV velocity $\vec{v}_{u_i}$ and acceleration $\vec{a}_{u_i}$. The $PR_{u_i}$ for UAV $u_i$ is a function of $\vec{v}_{u_i}$and $T_h$. Thus, $PR_{u_i}$ is proportional to the UAV trajectory. As a result, maintaining fairness in the travel distance for each UAV while performing the collaborative mission ensures a balance in energy consumption.

The energy consumption for communication depends on the size of the transmitted and received data at each timeslot. Given the transmitting data rate $C_{u_{ij}}^{tx}(t_n)$, transmitting power $P_{u_i}^{tx}$, and packet size $P_{size}$, the transmitting energy consumption $E_{u_i}^{tx}$ for the UAV $u_i$ is computed as $E_{u_i}^{tx} = \frac{(P_{u_i}^{tx} \times P_{size})}{C_{u_{ij}}^{tx}(t_n)}$. Similarly, given the receiving data rate $C_{u_{ij}}^{rx}(t_n)$ and the

receiving power $P_{u_i}^{rx}$ the energy consumption for receiving data $E_{u_i}^{rx}$ by UAV $u_i$ is computed as $E_{u_i}^{rx} = \frac{(P_{u_i}^{rx} \times P_{size})}{C_{u_{ij}}^{rx}(t_n)}$. For a UAV $u_i$ with initial maximum energy $E_{\max}$, the RE at $t_n$ is computed as follows:

$$RE_{u_i}(t_n) = E_{\max} - \sum_{t_n=1}^{t_{n-1}} \left[ \{ PR_{u_i} \tau + E_{u_i}^{tx} + E_{u_i}^{rx} \} \right]. \tag{4.5}$$

It is assumed that, when the $RE_{u_i}(t_n)$ UAV $u_i$ reaches the threshold energy level $E_{th}$, it enters a charging scheduling process to obtain a charging slot from the charging station for energy replenishment. After energy replenishment, $RE_{u_i}(t_n)$ of UAVs are reset to the $E_{\max}$ and return to the aerial network. The charging scheduling process is an optimization process comprising several joint objectives, such as minimizing the ascending and descending costs, and maintaining the connectivity and coverage density in FANETs [40]. This issue is out of scope of this study.

### 4.2.4 Problem Formulation

Owing to the constrained transmission power of UAVs, the source UAV $u_i$ transmits its data packet to the BS by selecting a series of relay UAVs resulting in a path $(u_i, u_j, u_k, \cdots, BS)$. It is assumed that the path comprises $h$ hops. The main objective of the QL algorithm is to minimize delay in terms of maximum PTS, select the stable path in terms of a maximum of minimum LDs, and select the relay UAVs provided by the highest RE level. Thus, jointly considering these three metrics, the link quality maximization problem can be represented as

$$\max \sum_{j=1}^{h-1} (w_1 PTS_{u_{ij}} + w_2 LD_{u_{ij}} + w_3 RE_{u_{ij}}), \tag{4.6}$$

subject to the following constraints: The inter-UAV distance should be bounded by $R_r \leq d_{u_{ij}}(t_n) \leq R_a$. The acceleration and velocity of UAVs should be within $\left\| \vec{a}_{u_i}(t_n) \right\| \leq a_{\max}$ and $\left\| \vec{v}_{u_i}(t_n) \right\| \leq v_{\max}$. The path delay should satisfy $\min LD_{u_{ij}} > \max PTT_{u_{ij}}$ to avoid link breakages, and the UAV RE level should satisfy $RE_{u_i}(t_n) \leq E_{th}$. Note that each term in (4.6) is normalized by using the maximum value of corresponding parameters found within two-hop neighbor information, which will be derived further in Section 4.3.2. Here, $w_1 + w_2 + w_3 = 1$, where $w_1$, $w_2$, and $w_3$ represent the weights of the three link-quality metrics, respectively.

According to (4.6), the optimal path selection in a FANET is highly coupled with relative mobility control, path stability, delay, and available UAV RE. Thus, this study jointly considers the mobility control strategy based on AFCA and the precisely calculated the UAV RE given by (4.5). Then, relative mobility knowledge, path delay based on PTT

given by (4.4), and UAV RE are fed into the QL module to make intelligent routing decision and trigger TA.

### 4.2.5 Framework for AFCA and QRIFC

In this subsection, the relation between the mobility in AFCA and the QL in QRIFC is briefly discussed. The AFCA considers each UAV as a particle with an initial velocity and position. Owing to the limited transmission range, generating the flocking rules only utilizing the one-hop neighbor mobility may produce partition in the swarm and delay the swarm cohesion. Thus, to adopt better collaborative movement with link stability, the AFCA extends the local view of each UAV by collecting mobility information up to two-hop neighbors. Then, AFCA calculates the topology formation rule (TFR) by taking the weighted sum of the cohesion, alignment, and separation rule. According to, TFR each UAV updates the mobility to construct the FANET topology $G(t_n)$ and predict the LD to select the relay UAV. The details of the AFCA are provided in Section 4.3.1.

In QL, the agent experiences different consequences in the environment, known as states. In a particular state, an agent may select an action from a set of allowable actions and obtain a reward or penalty. Iteratively, each agent gathers an experience, represented by a Q-value, that leads to an optimal policy in which the cumulative reward is maximized over time. Thus, the QL-based decision-making process can be described as a MDP tuple $(s, a, p, r)$, where $s$ represents a set of states, $a$ represents a set of possible actions, $p$ represents the state transition probability, and $r$ represents the reward. In AFCA, the mobility of UAVs in the next timeslot is updated based on the mobility status in the current timeslot. This property helps the routing model to adopt the MDP formulation with QL to make routing decisions in FANETs. In QRIFC, the data packet carried by each UAV is a learning agent and the FANET topology constructed by the AFCA is the environment, as shown in Figure 4.2.

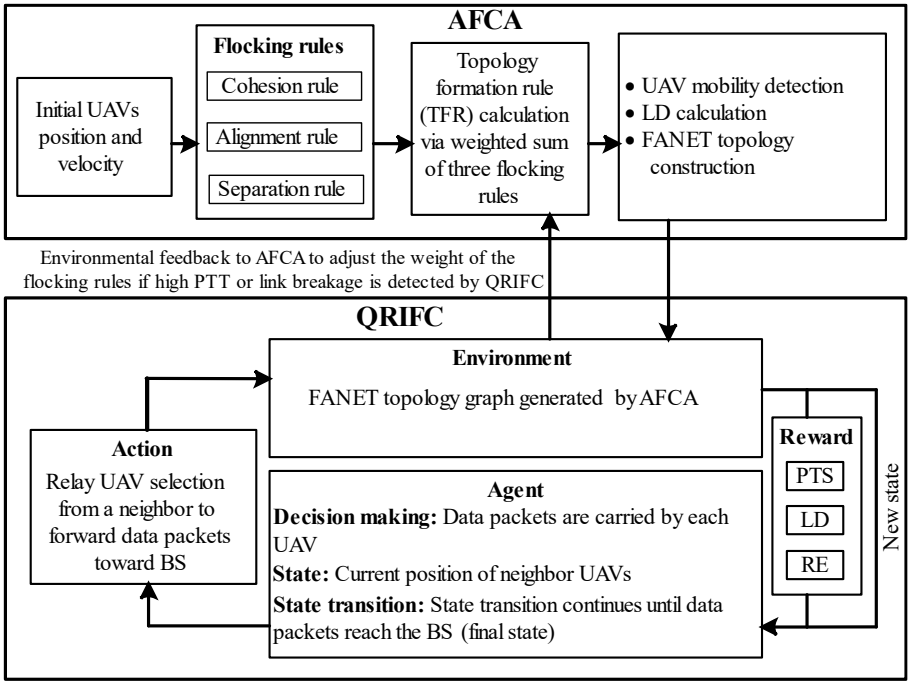



Figure 4.2 The interaction between AFCA and QRIFC.

The current state of the data packet is the location of the carrying UAV (source), which is routed to the BS (final state) through an intermediate state transition (one relay UAV to another) until it is delivered to the BS. When UAV $u_i$ transmits the data packets to its one-hop neighbor UAV $u_j$, this is defined as an action $a_{u_{ij}}$ and the associated link is $u_{ij}$. Through $a_{u_{ij}}$, the state of the data packet moves from $s_{u_i}$ to $s'_{u_i}$, and each $a_{u_{ij}}$ is evaluated using a new multi-objective reward $r_{u_{ij}}$ comprising the PTS, LD, and RE of the relay UAV. During the $a_{u_{ij}}$ evaluation via $r_{u_{ij}}$, if QRIFC detects a higher PTT and link breakage, the UAV can trigger the adjustment of the weight of the flocking rules to improve the neighbor intimacy. The Q-values for each neighbor link are updated using the following Bellman equation:

$$Q^{new}(s_{u_i}, a_{u_{ij}}) \leftarrow Q^{old}(s_{u_i}, a_{u_{ij}}) + \alpha_{u_{ij}}[r_{u_{ij}} + \lambda_{u_{ij}} \max_{a'_{u_{ij}}} Q(s_{u_i}{}', a'_{u_{ij}}) - Q^{old}(s_{u_i}, a_{u_{ij}})], \tag{4.7}$$

where $\max_{a'_{u_{ij}}} Q(s'_{u_i}, a'_{u_{ij}})$ represents the future Q-value expectation in the next state $s'_{u_i}$ after executing the best action $a'_{u_{ij}}$, $\alpha_{u_{ij}}$ and $\lambda_{u_{ij}}$ are the learning rate and discount factor, respectively, with values within [0 1]. $\alpha_{u_{ij}}$ specifies the degree to which the newly obtained

information overrides the old information and controls QL convergence. $\lambda_{u_{ij}}$ controls the importance of future rewards and identifies how much QL learns from its earlier mistakes. Thus, to estimate the precise Q-value and deal with the dynamic topology, both $\alpha_{u_{ij}}$ and $\lambda_{u_{ij}}$ should be adaptively controlled according to the PTT and mobility of UAVs. The details of QRIFC will be provided in Section 4.3.2.

## 4.3 Flocking Control and Routing Algorithms

In this section, the AFCA is derived to construct the topology of a UAV swarm and then determine the 3D LD, which is further utilized by QRIFC to make a routing decision.

### 4.3.1 Adaptive Flocking Control

The motion component, which is used to determine the mobility of each UAV using two-hop mobility, is calculated as follows:

The cohesion rule $\overrightarrow{CR}_{u_i}(t_n)$ defines the motion of each UAV attracted to the average centroid of the neighboring UAV positions. Its purpose is to keep the UAVs close to one another to avoid frequent link breakages or swarm partitions. According to Figure 4.3, the motion component is determined by $\overrightarrow{CR}_{u_i}(t_n)$ for UAV $u_i$ by utilizing the one-hop neighbor set $u_j \in N^a_{u_i}(t_n)$ located in the attraction range $R_r \le d_{u_{ij}}(t_n) \le R_a$ (e.g., $u_j \in (u_2, u_3, u_4)$), and two-hop neighbor $u_k \in N^2_{u_i}(t_n)$ (e.g., $u_k \in u_5$). $\overrightarrow{CR}_{u_i}(t_n)$ is computed as follows:

$$\overrightarrow{CR}_{u_i}(t_n) = \omega_1 \left[ \frac{\sum_{u_j \in N^a_{u_i}(t_n)} \left\{ \vec{p}_{u_j}(t_n) - \vec{p}_{u_i}(t_n) \right\}}{\left| N^a_{u_i}(t_n) \right|} \right] + \omega_2 \left[ \frac{\sum_{u_k \in N^2_{u_i}(t_n)} \left\{ \vec{p}_{u_k}(t_n) - \vec{p}_{u_i}(t_n) \right\}}{\left| N^2_{u_i}(t_n) \right|} \right], \tag{4.8}$$

where $\omega_1 + \omega_2 = 1$ represents the weight value for the one-hop and two-hop motion components. To prioritize a one-hop neighbor $\omega_1 > \omega_2$ is considered. Here, $|.|$ denotes the cardinality of a set.

The alignment rule $\overrightarrow{AR}_{u_i}(t_n)$ ensures that each UAV adopts a velocity direction according to its neighbor's average velocity. According to Figure 4.3, the motion component is determined by $\overrightarrow{AR}_{u_i}(t_n)$ for UAV $u_i$ by utilizing $u_j \in N^a_{u_i}(t_n)$, velocity $\vec{v}_{u_j}(t_n)$, $u_k \in N^2_{u_i}(t_n)$, and velocity $\vec{v}_{u_k}(t_n)$. $\overrightarrow{AR}_{u_i}(t_n)$ is computed as follows:

$$\overrightarrow{AR}_{u_i}(t_n) = \omega_3 \left[ \frac{\sum_{u_j \in N^a_{u_i}(t_n)} \left\{ \vec{v}_{u_j}(t_n) - \vec{v}_{u_i}(t_n) \right\}}{\left| N^a_{u_i}(t_n) \right|} \right] + \omega_4 \left[ \frac{\sum_{u_k \in N^2_{u_i}(t_n)} \left\{ \vec{v}_{u_k}(t_n) - \vec{v}_{u_i}(t_n) \right\}}{\left| N^2_{u_i}(t_n) \right|} \right], \tag{4.9}$$

where $\omega_3 + \omega_4 = 1$ represents the weights for the one-hop and two-hop neighbor velocity alignment rules. To prioritize the one-hop neighbor velocity, $\omega_3 > \omega_4$. Both $\overrightarrow{CR}_{u_i}(t_n)$ and $\overrightarrow{AR}_{u_i}(t_n)$ help the UAVs to maintain the transmission-range constraint $d_{u_{ij}}(t_n) \leq R_a$.

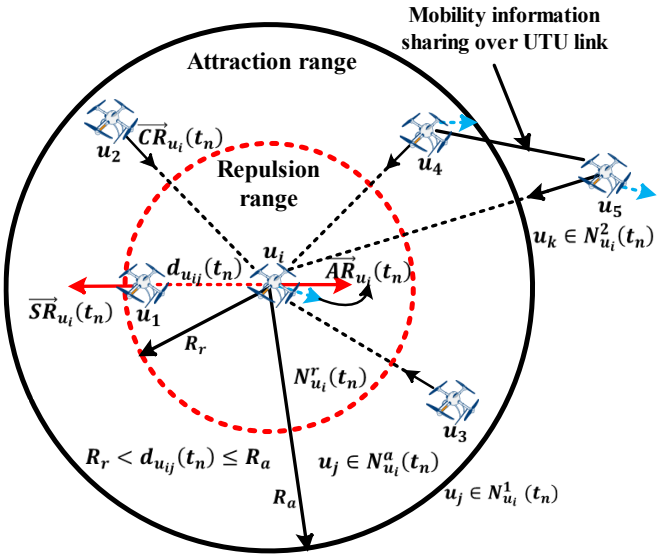



Figure 4.3 Motion components for each UAV in AFCA.

The separation rule $\overrightarrow{SR}_{u_i}(t_n)$ ensures a minimum separating distance among neighboring UAVs to avoid the inter-UAV collision. It also reduces the overlapping in the UAV sensor coverage to the ground terminal. According to Figure 4.3, the motion component is determined by the $\overrightarrow{SR}_{u_i}(t_n)$ for UAV $u_i$ by utilizing the one-hop neighbor UAVs $u_j \in N^r_{u_i}(t_n)$ (e.g., $u_j \in u_1$) located in the repulsion range $d_{u_{ij}}(t_n) \leq R_r$. $\overrightarrow{SR}_{u_i}(t_n)$ is computed as follows:

$$\overrightarrow{SR}_{u_i}(t_n) = \frac{\Sigma_{u_j \in N^r_{u_i}(t_n)} \left\{ \vec{p}_{u_i}(t_n) - \vec{p}_{u_j}(t_n) \right\}}{\left| N^r_{u_i}(t_n) \right|}. \tag{4.10}$$

Finally, the TFR, $\overrightarrow{TFR}_{u_i}(t_n)$ for UAV $u_i$ is computed by taking the weighted sum of the above three motion components:

$$\overrightarrow{TFR}_{u_i}(t_n) = \left[ \delta_1 \overrightarrow{CR}_{u_i}(t_n) + \delta_2 \overrightarrow{AR}_{u_i}(t_n) + \delta_3 \overrightarrow{SR}_{u_i}(t_n) \right] + \vec{\Phi}_{u_i}(t_n) + \mathcal{W}(t_n), \tag{4.11}$$

where $\delta_1 + \delta_2 + \delta_3 = 1$ represents the weight of each rule, and the values are adaptively determined according to the node density and inter-UAV distance relationship. Here, the

term $\overrightarrow{\Phi}_{u_i}(t_n) = \left[\vec{p}_{BS} - \vec{p}_{u_i}\right]$, is used to keep connected the swarm with the BS. $\mathcal{W}(t_n)$ represents the Gaussian noise with zero mean and limited variance to introduce the wind disturbance.

To adaptively control the rule weight, the TA is performed by sensing the changes in the one-hop neighboring minimum and maximum distances, represented as $TA_1$ and $TA_2$, respectively, for each UAV $u_i$ at $t_n$ and $t_{n+1}$. They are calculated as follows:

$$TA_1 = \exp\left[\epsilon_1\left\{R_r - \min_{u_j \in N^r_{u_i}(t_n)} d_{u_{ij}}(t_n)\right\}\right], \tag{4.12}$$

$$TA_2 = \exp\left[\epsilon_2\left\{\max_{u_j \in N^a_{u_i}(t_n)} d_{u_{ij}}(t_n) - R_a\right\}\right], \tag{4.13}$$

where $\epsilon_1$ and $\epsilon_2$ are sensitivity constants ($\epsilon_1 > \epsilon_2$). $TA_1$ determines the degree of violation in the imposed safety distance constraint $\min_{u_j \in N^r_{u_i}(t_n)} d_{u_{ij}}(t_n) \geq R_r$ and increases exponentially if $d_{u_{ij}} < R_r$. If $TA_1 > \Delta_1$, the weight becomes $\delta_3 > (\delta_1 + \delta_2)$ and the effect of $\overrightarrow{SR}_{u_i}(t_n)$ is increased. $TA_2$ determines the degree of violation of the imposed transmission range constraint $\max_{u_j \in N^a_{u_i}(t_n)} d_{u_{ij}}(t_n) \leq R_a$ and increases exponentially when $d_{u_{ij}} \geq R_r$. If $TA_2 > \Delta_2$, the weight becomes $(\delta_1 + \delta_2) > \delta_3$ and the effects of both $\overrightarrow{CR}_{u_i}(t_n)$ and $\overrightarrow{AR}_{u_i}(t_n)$ are increased. Here, both $\Delta_1$ and $\Delta_2$ are the predefined threshold constants. The adaptive adjustment of flocking rule weight given by TA helps UAV swarm to adjust its connectivity with remaining neighbor UAVs in case of neighbor UAV failure due to hardware or software malfunction. It also helps the UAV swarm to adjust the inter-UAV distance accordingly if any UAV left due to energy limitation or re-join into the aerial networks after energy replenishment.

Each UAV $u_i$ utilizes $\overrightarrow{TFR}_{u_i}(t_n)$ as its control input to determine $\vec{a}_{u_i}(t_n)$, $\vec{v}_{u_i}(t_{n+1})$, and $\vec{p}_{u_i}(t_{n+1})$ in the next timeslot $t_{n+1}$, as given below:

$$\vec{a}_{u_i}(t_n) = \left(\frac{\overrightarrow{TFR}_{u_i}(t_n)}{\left\|\overrightarrow{TFR}_{u_i}(t_n)\right\|}\right) \times \tan^{-1}\left(\left\|\overrightarrow{TFR}_{u_i}(t_n)\right\|\right) \times \frac{2}{\pi} \times a_{\max}, \tag{4.14}$$

$$\vec{v}_{u_i}(t_{n+1}) = \vec{v}_{u_i}(t_n) + \vec{a}_{u_i}(t_n) \times \tau, \tag{4.15}$$

$$\vec{v}_{u_i}(t_{n+1}) = \begin{cases} \vec{v}_{u_i}(t_{n+1}) \times [\exp\ (\eta - 1)], & \left\|\vec{v}_{u_i}(t_{n+1})\right\| < v_{\max} \\ \left[\frac{\vec{v}_{u_i}(t_{n+1})}{\left\|\vec{v}_{u_i}(t_{n+1})\right\|}\right] \times v_{\max} \times [\exp\ (\eta - 1)], & \left\|\vec{v}_{u_i}(t_{n+1})\right\| \geq v_{\max} \end{cases}, \tag{4.16}$$

$$\vec{p}_{u_i}(t_{n+1}) = \vec{p}_{u_i}(t_n) + \vec{v}_{u_i}(t_n) \times \tau + \frac{1}{2} \times \vec{a}_{u_i}(t_n) \times \tau^2, \tag{4.17}$$

For each UAV $u_i$, to keep the magnitude of acceleration within $\left\|\vec{a}_{u_i}(t_n)\right\| \leq a_{\max}$, a trigonometric function is applied in (4.14). Similarly, to maintain the velocity for each UAV within $\left\|\vec{v}_{u_i}(t_n)\right\| \leq v_{\max}$, equation (4.16) is applied, where $a_{\max}$ and $v_{\max}$ represents the maximum attainable acceleration and velocity, respectively. Here, $\eta \in [0\ 1]$ represents the velocity synchronization term with neighboring UAVs $u_j \in N^a_{u_i}(t_n)$ and computed as $\eta = \frac{\left\|\sum_{u_j \in N^a_{u_i}(t_n)} \vec{v}_{u_j}(t_n)\right\|}{\sum_{u_j \in N^a_{u_i}(t_n)} \left\|\vec{v}_{u_j}(t_n)\right\|}$. $\eta$ is utilized as an exponential term in (4.16) to ensure that each UAV can only attain the maximum velocity to fly if its neighbor's velocity is properly synchronized. Otherwise, the velocity of the UAV decreases according to $\eta$ to avoid chaotic movement in the swarm. Here, $\|.\|$ represents the absolute magnitude. Based on the above mobility, the FANET topology $G(t_n)$ is constructed.

$LD_{u_{ij}}$ is defined as the maximum link subsistence time $t$ between two neighboring UAVs [148]. It is bounded by the inter-UAV distance $d_{u_{ij}} = R_a$. Let two UAVs $u_i$ and $u_j$ with initial positions $p_{u_i} = (x_{u_i}, y_{u_i}, z_{u_i})$ and $p_{u_j} = (x_{u_j}, y_{u_j}, z_{u_j})$, velocities $v_{u_i}$ and $v_{u_j}$, and flying directions $(\theta_{u_i}, \phi_{u_i})$ and $(\theta_{u_j}, \phi_{u_j})$. After time $t$, $d_{u_{ij}}$ is estimated as follows:

$${d_{u_{ij}}}^2 = (\mathcal{X} + at)^2 + (\mathcal{Y} + bt)^2 + (\mathcal{Z} + ct)^2, \tag{4.18}$$

where $\mathcal{X} = (x_{u_i} - x_{u_j})$, $\mathcal{Y} = (y_{u_i} - y_{u_j})$, $\mathcal{Z} = (z_{u_i} - z_{u_j})$, $a = (v_{u_i} \sin\theta_{u_i} \cos\phi_{u_i} - v_{u_j} \sin\theta_{u_j} \cos\phi_{u_j})$, $b = (v_{u_i} \sin\theta_{u_i} \sin\phi_{u_i} - v_{u_j} \sin\theta_{u_j} \sin\phi_{u_j})$, and $c = (v_{u_i} \cos\theta_{u_i} - v_{u_j} \cos\theta_{u_j})$. By substituting $d_{u_{ij}} = R_a$ into (18), $t^2(a^2 + b^2 + c^2) + t(2a\mathcal{X} + 2b\mathcal{Y} + 2c\mathcal{Z}) + \mathcal{X}^2 + \mathcal{Y}^2 + \mathcal{Z}^2 - {R_a}^2 = 0$ is obtained. Then,

$$At^2 + Bt + C = 0, \tag{4.19}$$

is found where $A = (a^2 + b^2 + c^2)$, $B = (2a\mathcal{X} + 2b\mathcal{Y} + 2c\mathcal{Z})$, and $C = \mathcal{X}^2 + \mathcal{Y}^2 + \mathcal{Z}^2 - {R_a}^2$. The solution of (4.19) has one positive root and one negative root. The positive root defines the $LD_{u_{ij}}$.

To predict the updated topology with minimal control overhead, the hello interval $HI_{u_i}$ for each UAV $u_i$ is estimated by the minimum $LD_{u_{ij}}$ found within the one-hop neighbor $N_{u_i}^1(t_n)$ and is computed as follows:

$$HI_{u_i} = \sigma \times [\min_{u_j \in N_{u_i}^1(t_n)} LD_{u_{ij}}], \tag{4.20}$$

where $\sigma$ is the hello interval factor with the default value of 0.5. Each UAV exchanges the hello packet with $N_{u_i}^1(t_n)$ neighbor UAVs using the hello interval given in (4.20), including a hello packet sequence number, a unique UAV ID, mobility information (3D position, velocity, LD, PTT, and RE) of it and its neighbors. According to the received hello packets, each UAV $u_i$ updates its one-hop $u_j \in N_{u_i}^1(t_n)$ and two-hop $u_k \in N_{u_i}^2(t_n)$ neighbor table. Afterward, the UAV updates its mobility in the next time slot and makes routing decisions using the QRIFC routing module. The above process is summarized in Algorithm 4.1.

---

**Algorithm 4.1:** AFCA

---

**Input:** Neighbor UAV location and threshold $\Delta_1$ and $\Delta_2$
**Output:** Topology $G(t_n)$, UAV mobility, LD, and $HI_{u_i}$
1: **Proceed** to the next time slot $t_{n+1}$
**Phase 1:** Hello packet (HP) broadcasting
2: **for** each $u_j \in U(t_n)$ **do**
3: Transmit HPs to $u_i \in N_{u_j}^1$ UAVs with its $u_k \in (N_{u_j}^1 \cap N_{u_i}^1)$ mobility information
4: **end for**
**Phase 2:** One-hop and two-hop neighbor table update
5: **for** ∀ received HPs at UAV $u_i$ from neighbor $u_j \in N_{u_i}^1$ do
6: Get originator $u_j \in N_{u_i}^1$ unique UAV ID
7: **if** ($u_j \in N_{u_i}^1$) **then**
8: **if** (received HP sequence > record HP sequence) then
9: Update the position of $u_j \in N_{u_i}^1$ and $u_k \in N_{u_i}^2$
10: **end if**
11: **else**
12: Add a new record for $u_j \in N_{u_i}^1$ and its neighbor $u_k \in N_{u_i}^2$
13: **end if**
14: **end for**
**Phase 3:** Mobility update according to weighted flocking rules
15: **for** each UAV $u_i$ having $u_j \in N_{u_i}^1$ **do**
16: Calculate the $\overrightarrow{CR}_{u_i}(t_n)$, $\overrightarrow{AR}_{u_i}(t_n)$, and $\overrightarrow{SR}_{u_i}(t_n)$ using (4.8)−(4.10)
17: **if** ($TA_1 > \Delta_1$) **then** // Violation of the separating distance
18: Set the rule weight $\delta_3 > (\delta_1 + \delta_2)$ in (4.11)
19: **else if** ($TA_2 > \Delta_2$) **then** //Violation of the transmission range
20: Set the rule weight $(\delta_1 + \delta_2) > \delta_3$ in (4.11)
21: **end if**
22: Calculate the $\overrightarrow{TFR}_{u_i}(t_n)$ using (4.11)
23: Compute $\vec{a}(t_n)$, $\vec{v}_{u_i}(t_{n+1})$, and $\vec{p}_{u_i}(t_{n+1})$ using (4.14)−(4.17)
24: Construct the FANET topology $G(t_n)$
25: Calculate the $LD_{u_{ij}}$ by solving (4.19)
26: Update the $HI_{u_i}$ using (4.20)
27: Update the $RE_{u_i}(t_n)$ using (4.5)
28: **end for**

---

### 4.3.2 Q-Learning-Based Routing

QRIFC is a position-based multi-hop routing protocol combined with QL, where data packets carried by each UAV act as RL agents and adaptively learn how to reach the BS by utilizing the relative mobility (PTS and LD) with a two-hop neighbor given by AFCA. Routing decisions based on only one-hop neighbors' knowledge might be less optimal

because they do not consider the availability of further suitable relay UAVs for forwarding toward the BS. QRIFC enhances the routing performance by expanding the local view of the topology at the current source UAV to include two-hop neighbor information. The optimal relay UAV selection strategy for a source UAV to forward the sensed data packet to the destination BS in QRIFC is explained below:

#### 4.3.2.1 State Exploration for Relay UAV Selection

A state exploration strategy is derived to avoid unnecessary detours during the initial decision making for selecting the relay UAV. This is done by defining a potential relay (PR) set for each respective source UAV considering the PTS up to a two-hop neighbor. It also helps QL to tackle the large state space problem. To explain the state exploration strategy for the data packets to reach the BS, a source UAV $u_i$ and destination BS is considered, as shown in Figure 4.4. The source UAV $u_i$ forwards data packets to the BS by sequentially selecting relay UAV from its one-hop PR, $u_j \in PR_{1-hop}$ (e.g., $(u_1, u_2, u_3)$) , and two-hop PR, $u_k \in PR_{2-hop}$ (e.g., $(u_6, u_7, u_8)$) based on the value of $PTS > 0$.

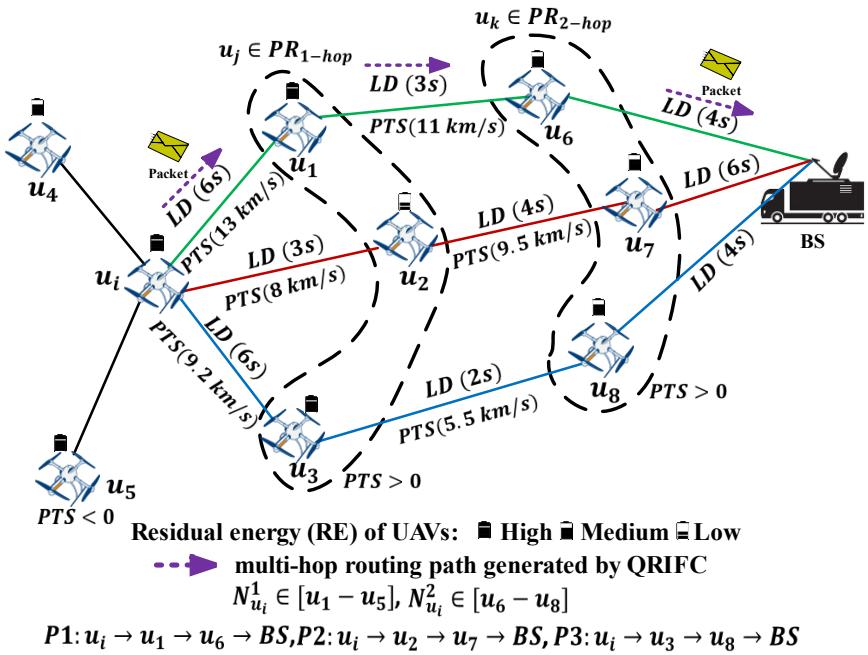



Figure 4.4 A routing example in QRIFC using PTS, LD, and UAV RE.

The PTT up to the two-hop neighbor for UAV $u_i$ $PTT_{u_{i\to k}} \in (PTT_{u_{ij}}, PTT_{u_{jk}})$ is obtained using (4.4). While selecting a relay UAV, the PTS considers the progress of the distance toward the destination BS and PTT offered by the relay UAV. The PTS up to the two-hop path $PTS_{u_{i\to k}}$ for UAV $u_i$ is computed by:

$$PTS_{u_{i\to k}} = \frac{d(u_i,BS)-d(u_j,BS)}{PTT_{u_{ij}}} + \frac{d(u_j,BS)-d(u_k,BS)}{PTT_{u_{jk}}}, \tag{4.21}$$

where $d(.)$ represents the Euclidian distance between the respective source UAV and BS. Here, $PTS_{u_{i\to k}} > 0$ indicates that the chosen relays show distance progress toward the BS, and the higher the value of PTS, the more suitable the link is, as it intends to provide less PTT.

#### 4.3.2.2 Multi Objective Reward Function

To simply explain the relay selection strategy to transmit the data packets toward the BS from the source state UAV $u_i$, three suitable paths are considered: $P1$: $u_i \to u_1 \to u_6 \to BS$, $P2$: $u_i \to u_2 \to u_7 \to BS$, and $P3$: $u_i \to u_3 \to u_8 \to BS$, as shown in Figure 4.4. The source UAV $u_i$ evaluates its action $a_{u_{ij}}$ as a relay UAV selection by $r_{u_{ij}}$ to discover optimal routing paths to minimize delay, ensure link stability, and avoid energy and routing holes. Because the reward function reinforces the action policy of an agent, a good reward function can accelerate QL convergence. Thus, to obtain the optimal path, QRIFC jointly considers three important metrics of PTS, LD, and RE up to two-hop neighbor UAVs to update the Q-value for each neighbor link as described in the objective function (4.6).

The first component of the reward $r^1_{u_{ij}}$ is the PTS and it minimizes the end-to-end delay. A relay UAV that provides a higher PTS is more suitable for the next relay to forward data packets toward the BS. Path $P1$ gives a better PTS than $P2$ and $P3$; thus, relay $u_1$ is more suitable. The $r^1_{u_{ij}}$ is normalized as follows:

$$r^1_{u_{ij}} = \frac{PTS_{u_{i\to k}}}{\max\limits_{u_j \in PR_{1-hop}, u_k \in PR_{2-hop}} PTS_{u_{i\to k}}}, \tag{4.22}$$

The second component $r^2_{u_{ij}}$ is the LD given by (4.19), which ensures the stability of the chosen link. The maximum LD of one path is equal to the minimum LD of the two adjacent UAVs along the path. This indicates that LD is not an additive metric. Therefore, it is preferable to use a maximum–minimum LD to select a more stable path. For instance, in Figure 4.4, the minimum LD for paths $P1$, $P2$, and $P3$ are 3 s, 3 s, and 2 s, respectively. If the maximum of the minimum LD is taken, $P1$ and $P2$ are suitable to forward data packets

by selecting relays $u_1$ or $u_2$ as they intend to provide a higher LD compared to $P3$. Thus, LD up to the two-hop neighbor $LD_{u_{i\to k}}$ is defined as follows:

$$LD_{u_{i\to k}} = \min_{u_j \in PR_{1-hop}, u_k \in PR_{2-hop}} (LD_{u_{ij}}, LD_{u_{jk}}), \tag{4.23}$$

Based on the above discussion, if the source UAV has multiple paths to reach the destination, the maximum of the minimum LD $\varphi_{u_{i\to k}}$ along those paths provides a better stable path and is computed as $\varphi_{u_{i\to k}} = \max_{u_j \in PR_{1-hop}, u_k \in PR_{2-hop}} [\min LD_{u_{i\to k}}]$. To normalize the two-hop stable path selection metric $\varphi_{u_{i\to k}}$, an exponential function is applied and computed as follows:

$$r^2_{u_{ij}} = 1 - e^{-\varphi_{u_{i\to k}}}, \tag{4.24}$$

The third component $r^3_{u_{ij}}$ is the RE of the relay UAVs, which helps to create a balance in energy consumption. A relay UAV with higher RE is more suitable for the next relay. Thus, path $P1$ gives a better RE than $P2$ and $P3$. The $r^3_{u_{ij}}$ is normalized as follows:

$$r^3_{u_{ij}} = \frac{1}{2E_{\max}} (RE_{u_{ij}} + RE_{u_{jk}}), \tag{4.25}$$

The weighted sum of the above three components gives the $r^4_{u_{ij}}$ for selecting the relay UAV and computed as follows:

$$r^4_{u_{ij}} = \frac{1}{3}\left(w_1 r^1_{u_{ij}} + w_2 r^2_{u_{ij}} + w_3 r^3_{u_{ij}}\right). \tag{4.26}$$

Based on the above discussion, path $P1$ receives more reward because it provides better PTS, LD, and RE of UAVs to route data packets toward the BS. If the relay UAV is chosen by using only one-hop information, relay $u_3$ would be the best possible action. However, in the next state transition from $u_3$, the routing trapped in the local minimum as $u_3$ has less LD with $u_8$ and $u_2$ has low RE.

If the selected action is trapped in the local minimum, that is, the chosen relay UAV $u_j$, shows distance progress to the BS but there is no potential neighbor UAV to forward the data packets further, QRIFC allocates the minimum reward $r_{\min}$. If the next state is the BS, it directly allocates the maximum reward $r_{\max}$. Otherwise, when UAV $u_j$ acts as a relay, each $a_{u_{ij}}$ is evaluated by using the $r^4_{u_{ij}}$ given in (4.26). Additionally, if a relay UAV does not convey an ACK to the source UAV, it will consider the failure state and allocate $r_{\min}$ to that relay. Finally, $r_{u_{ij}}$ for updating the Q-value is computed as follows:

$$r_{u_{ij}} = \begin{cases} r_{\max} = 100, & if\ link\ u_{ij}\ lead\ to\ the\ BS \\ r_{\min} = -100, & if\ link\ u_{ij}\ is\ local\ minimum \\ 100 \times r_{u_{ij}}^4, & otherwise \end{cases}, \tag{4.27}$$

#### 4.3.2.3 Adaptive Q-learning Rate and Discount Factor

Based on the discussion in Section 4.2.5, $\alpha_{u_{ij}}$ and $\lambda_{u_{ij}}$ should be adaptively adjusted to produce a stable Q-value considering the dynamic topology. The learning rate $\alpha_{u_{ij}} \in [0\ 1]$ for link $u_{ij}$ is updated according to the exponential of the normalized one-hop $PTT_{u_{ij}}$ as follows:

$$\alpha_{u_{ij}} = \begin{cases} 1 - \exp\left[-\left(\frac{\left\|PTT_{u_{ij}} - m_{u_{ij}}\right\|}{\mu_{u_{ij}}}\right)\right], & \mu_{u_{ij}} \neq 0 \\ 0.3, & \mu_{u_{ij}} = 0 \end{cases}, \tag{4.28}$$

where $m_{u_{ij}}$ and $\mu_{u_{ij}}$ are the mean and variance of the $PTT_{u_{ij}}$ computed in (4), respectively. According to (4.28), if $PTT_{u_{ij}}$ is higher, $\alpha_{u_{ij}}$ increases exponentially to update the Q-value faster. A higher $\lambda_{u_{ij}}$value specifies the stability of the expected future Q-value, and a smaller $\lambda_{u_{ij}}$ provides a vulnerable Q-value expectation. Owing to the need for a reliable link $u_{ij}$, the value of $\lambda_{u_{ij}} \in [0, 1]$ for link $u_{ij}$ is adaptively modified according to the level of mobility defined by the relative distance $d_{u_{ij}}$ as follows:

$$\lambda_{u_{ij}} = \begin{cases} 1 - \frac{\left\|R_r - d_{u_{ij}}\right\|}{R_r}, & if\ \ 0 \leq d_{u_{ij}} \leq R_r \\ 1 - \frac{d_{u_{ij}}}{R_a}, & if\ \ R_r < d_{u_{ij}} \leq R_a \end{cases}. \tag{4.29}$$

#### 4.3.2.4 Exploration-exploitation trade-off for routing decision

Exploration is the search for a new state of data packets via a new action that may provide a better reward. Exploitation refers to the performance of the best action according to the maximum Q-value. However, the action taken during the exploration can be either good or bad, which may produce detours. Thus, to satisfy the trade-off between exploration and exploitation in highly dynamic FANETs, each UAV $u_i$ adaptively determines whether to perform exploration or exploitation according to the value of NALD $\overline{ALD}_{u_i}$. The $\overline{ALD}_{u_i}$ is computed using $LD_{u_{i\to k}}$ given in (4.23) and computed as follows:

$$\overline{ALD}_{u_i} = \frac{\frac{1}{\left|N^1_{u_i} \cup N^2_{u_i}\right|} \sum LD_{u_{i\to k}}}{\max LD_{u_{i\to k}}} < ALD_{th}, \tag{4.30}$$

If $\overline{ALD}_{u_i} < ALD_{th}$, where $ALD_{th}$ is a threshold value set to 0.9, UAV $u_i$ decides to explore because it indicates considerable changes in the neighboring mobility state. During exploration, rather than randomly selecting a link, a neighboring UAV that offers maximum PTS satisfying the constraints $PTS_{u_{i\to k}} > 0$ and $\min(LD_{u_{i\to k}}) > \max(PTT_{u_{i\to k}})$ is included in the $PR_{hop-1}$ set for selection as a relay UAV. It also assists QL to deal with large action space. If $\overline{ALD}_{u_i} > ALD_{th}$, the neighboring mobility state is relatively stable. Thus, the UAV decides to execute exploitation, and the source UAV $u_i$ selects the neighboring UAV, $u_j \in PR_{hop-1}$, which offers the maximum Q-value satisfying the constraint $\min(LD_{u_{i\to k}}) > \max(PTT_{u_{i\to k}})$.

When the source UAV has less neighbor stability and hardly meets the imposed LD constraint, QRIFC can trigger TA to adjust the weight of the cohesion and alignment rules in (4.12) to maintain path stability. To avoid routing loops, for each state transition of data packets, the updated Q-value is continually traced against previously visited UAVs so that none of the intermediate relay UAVs are selected more than once in the end-to-end path. Additionally, the penalty $r_{\min}$ in (4.27) helps to avoid unnecessary detours of data packets. This process is described in Algorithm 4.2. Lines $4-26$ represent the state exploration strategy according to the condition $\overline{ALD}_{u_i}(t_n) < ALD_{th}$ for data packets relayed toward the BS. Additionally, it includes the TA triggering method to improve the neighboring proximity according to the condition $\min\left(LD_{u_{i\to k}}\right) > \max\left(PTT_{u_{i\to k}}\right)$. Lines $27-29$ represent the exploitation strategy according to the maximum Q-value.

### 4.3.3 Topology Update Cost and Time Complexity

Both the AFCA and QRIFC are executed in each UAV in a distributed manner. As a result, the topology update cost for each UAV at each sequential mobility update iteration depends on the degree of each UAV at $HI_{u_i}$ given by (4.21). Thus, the approximate topology update cost for $HI_{u_i}$ is $O(2\Delta)$ messages, including ACKs, where $\Delta$ represents the degree of the UAV in the FANET topology.

The time complexity of the AFCA for each UAV is $O(\Delta^2)$ because each UAV updates its mobility using the mobility information of its one-hop and two-hop neighbors. Because QRIFC is a QL-based algorithm, its time complexity is $O(\Delta^2)$ for both exploring a new state for data packets using two-hop neighbor list as an action and updating the reward or penalty for each action.

**Algorithm 4.2:** QRIFC

**Input:** FANET topology $G(t_n)$ generated by AFCA
**Output:** Optimal relay selection for data packets to reach BS
**Proced** to next timeslot $t_{n+1}$
$Q - value = PTT_{u_{i\to k}} = PTS_{u_{i\to k}} = 0$ // Initialization
**while** data packets need to transmit **do**
  **if** $(d(u_i, BS) \le R_a)$ **then** // UAV near to BS
    Transmit the data to BS and allocate maximum reward
  **else**
    Make routing decisions based on Q-learning
    **if** $\left(\overline{ALD}_{u_i}(t_n) < ALD_{th}\right)$ **then** //exploration
      **for** $u_j \in N_{u_i}^1(t_n)$ of $u_i$ **do**
        Calculate $PTT_{u_{i\to k}}$ using (4.4)
        Calculate $PTS_{u_{i\to k}}$ using (4.21)
      **end for**
      **if** $\left(PTS_{u_{i\to k}} > 0\right)$ **then**
        **if** $\left(\min\left(LD_{u_{i\to k}}\right) > \max\left(PTT_{u_{i\to k}}\right)\right)$ **then**
          Update $PR_{1-hop} \leftarrow$ according to the descending order of $PTS_{u_{i\to k}}$
          Select relay UAV $u_j \in PR_{1-hop}$ that offer maximum $PTS_{u_{i\to k}}$
          Obtain the reward using (4.27) and update Q-value using (4.28), (4.29), and (4.7)
        **else**
          Trigger TA to satisfy $\min LD_{u_{i\to k}} > \max PTT_{u_{i\to k}}$ by adjusting the weight of the flocking rule in (4.11)
          Select relay UAV $u_j \in PR_{1-hop}$ that satisfy $\min\left(LD_{u_{i\to k}}\right) > \max\left(PTT_{u_{i\to k}}\right)$
          Obtain the reward using (4.27) and update Q-value using (4.28), (4.29), and (4.7)
        **end if**
      **else**
        Trigger penalty for bad action
        Give minimum reward and update Q-value
      **end if**
    **else** // exploitation
      Select the relay $u_j \in PR_{1-hop}$ with maximum Q-value
      Obtain the reward using (4.27) and update Q-value using (4.28), (4.29), and (4.7)
    **end if**
  **end if**
**end while**

## 4.4 Performance Evaluation

In this section, extensive computer simulation is conducted to evaluate the performance of the proposed AFCA and QRIFC. MATLAB R2021a and its reinforcement learning toolbox are used to implement both AFCA and QRIFC. Because QRIFC is a position-aware QL-based multi-objective optimization routing protocol, QMR [62] and QTAR [33] are

suitable for comparison with QRIFC. In the reward function, QTAR considers the two-hop delay, PTS, and RE. QMR considers the one-hop delay and UAV RE in the reward function. QTAR executes exploration based on only two-hop PTS information under the generic Gauss–Markov mobility model, whereas QMR performs exploration based on only one-hop PTS under the random waypoint mobility model. However, for comparison in a fair environment, the AFCA is considered for both QMR and QTAR.

### 4.4.1 Simulation Setup

Initially, the UAVs are randomly deployed in a 3D mission area with a topology dimension of $3000 \times 3000 \times [100 - 400]\, m^3$ to perform a collaborative surveillance mission. The maximum transmission (attraction) and repulsion ranges for each UAV are set to $R_a = 250\, m$ and $R_r = 50\, m$, respectively. Additionally, the maximum allowable velocity $v_{\max}$ and acceleration $a_{\max}$ for each UAV are set to 20 m/s and 5 m/$s^2$, respectively. The total surveillance mission duration is $T = 5000$ s, and the length of each time slot is $\tau = 2$ s. The minimum threshold value for calculating the LD is initially set to 2 s and $HI_{u_i} = 0.5$ s. To execute the TA, it is set that $\epsilon_1 = 0.8$, $\epsilon_2 = 0.3$, $\Delta_1 = 55$, and $\Delta_2 = 20$. To produce data traffic, a constant bitrate (CBR)-based video streaming application operating on each UAV is considered. At each timeslot, each UAV periodically transmits the data packet to the destination BS. Other important simulation parameters are listed in Table 4.2.

Table 4.2 Simulation parameters (QRIFC).

| Parameter | Value |
|---|---|
| Topology dimension | $3000 \times 3000 \times [100 - 400]\, m^3$ |
| Number of UAVs ($N$) | A variable number $N$ (50−400) |
| UAV initial energy ($E_{max}$) | $2 \times 10^5$ Joules |
| Path loss exponent ($\zeta$) | 3 |
| SINR threshold ($\gamma_{th}$) | 2 dB |
| MAC protocol | IEEE 802.11 DCF |
| Traffic type | CBR |
| Transport protocol | User datagram protocol |
| Traffic load per video stream | 2 Mbit/s |

| | |
|---|---|
| Packet arrival model | Poisson |

### 4.4.2 Performance Metrics

The performance metrics used to evaluate the stability of the AFCA flocking model are as follows:

- Average maximum and minimum UTU distance: In AFCA, UAVs are initially randomly distributed. Thus, the maximum and minimum UTU distances can be greater than or less than $R_a$ and $R_r$, respectively. Iteratively, each UAV in the swarm communicates with neighboring UAVs and experiences movement to optimally gather as a connected swarm to the BS. Thus, observing the changes in the average maximum and minimum inter-UAV distances with respect to the number of iterations helps to understand the UTU link stability and cohesion of the flocking process.

- Traveling distance fairness (TDF): TDF for each UAV validates the motion fairness among the UAVs during the entire collective motion process generated by the AFCA. It is computed as $\frac{\left(\sum_{i=1}^{N} D_{u_i}\right)^2}{N \times \sum_{i=1}^{N} \left(D_{u_i}\right)^2}$, where $D_{u_i}$ represents the travel distance by a UAV and is computed using (4.18). A TDF value close to 1 indicates that the travel distances of all UAVs are the same, which ensures a balance in energy consumption. The above performance metrics were used to compare the one-hop and two-hop AFCA to validate the effectiveness of extending the local view.

The performance metrics used to evaluate the QRIFC routing performance are as follows:

- Average number of retransmissions (ANR): ANR denotes the average number of packets that must be retransmitted by each UAV owing to link breakages and data congestion.

- Packet delivery ratio (PDR): PDR refers to the ratio between successfully delivered data packets at the BS and the total number of data packets originating from a source UAV.

- Average end-to-end delay (AETED): AETED refers to the total average time required to successfully deliver data packets to the BS from a source UAV. It is computed using (4.5).

- Control overhead: The control overhead includes hello packets that contain mobility information for each UAV and its two-hop neighbors to construct the FANET

topology and make routing decisions. The above network performance metrics were observed with respect to different number of UAVs and different velocities.

- Normalized residual energy (NRE): The NRE for each UAV is computed using $RE_{u_i}$ given by (4.6) and normalized as follows: $\frac{RE_{u_i}}{E_{max}}$. The NRE is observed after completing the simulation, and a lower NRE indicates higher UAV energy consumption.

- Exploration-exploitation trade-off: To examine the exploration-exploitation trade-off and convergence in QL, the average reward with respect to the number of iterations is observed for different types of neighbor state exploration strategies for data packet forwarding with the proposed QRIFC.

### 4.4.3 Simulation Results and Discussion

In this subsection, the simulation results are summarized and comparatively discussed. Figure 4.5 shows a stable UAV swarm flocking in 3D space for 200 UAVs generated by AFCA to perform a collaborative surveillance mission, which is connected to the BS. The green line represents the trajectory of each UAV, which is iteratively generated by the AFCA to achieve swarm cohesion.

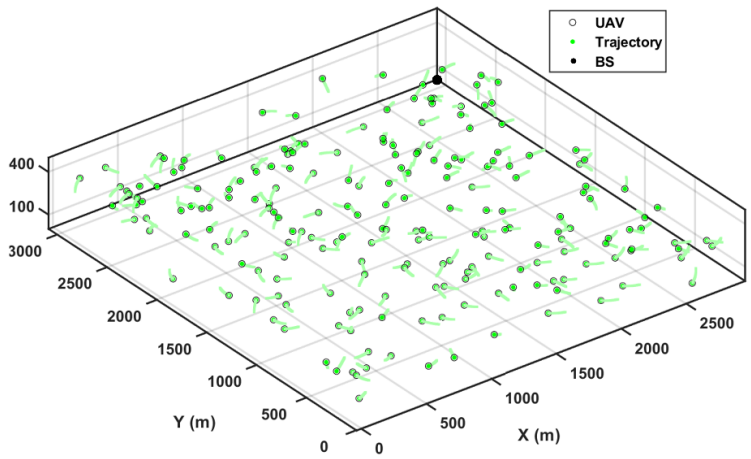
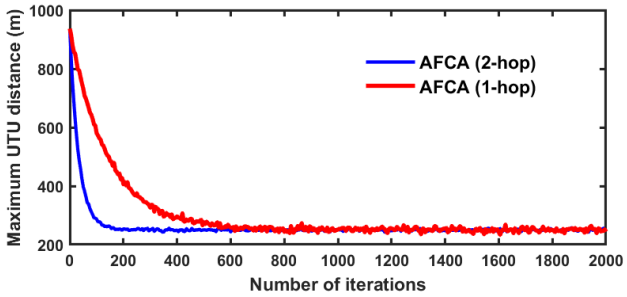



Figure 4.5 An example of flocking generated by AFCA.

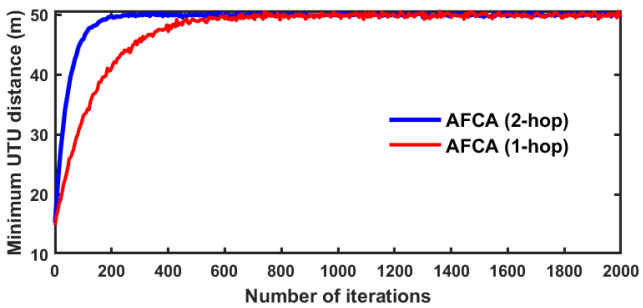
Figure 4.6 shows the changes in the average maximum and minimum UTU distances with respect to the number of iterations for 200 UAVs. Because UAVs are randomly deployed, the maximum and minimum UTU distances are initially greater than or less than $R_a = 300$ m and $R_r = 50$ m, respectively. According to Figure 4.6(a), the maximum UTU distance is initially 930 m. Owing to the weighted cohesion and alignment rules using two-

hop neighbor mobility information in AFCA, the maximum UTU distance begins to decrease rapidly in the first 80 iterations as shown in Figure 4.6(a). Owing to the rapid decrease in the UTU distance, the separation rule gradually starts to operate. Thus, after 80 iterations, the maximum UTU distances slightly decrease. After approximately 240 iterations, an equilibrium state is achieved, and changes in the maximum UTU distances are stabilized at approximately 250 m. In contrast, one-hop AFCA requires more iterations (approximately 750 iterations) to stabilize the changes in the average maximum UTU distances (red line in Figure 4.6(a).

According to Figure 4.6(b), the changes in the average minimum UTU distances initially increase rapidly with the help of the weighted separation rule and stabilize at a threshold of 50 m after approximately 250 iterations using two-hop information. In contrast, the changes in the average minimum distances in the AFCA with one-hop information reach a stable state after approximately 680 iterations. Therefore, generating flocking rules with adaptive adjustment of rule weights using two-hop information provides faster swam cohesion satisfying the imposed safety distance and transmission range constraints. Additionally, balances in the UTU distances control the uniform node distribution within the mission area, which enhances aerial coverage and SINR performance.




(a) Average maximum UTU distance (m) in AFCA.




(b) Average minimum UTU distance (m) in AFCA.

Figure 4.6 Changes in UTU distances in AFCA.

Figure 4.7 shows the TDF for different number of UAVs in the AFCA. In Figure 4.7, the TDF is close to 1 with an increasing number of UAVs, which ensures fairness in the traveling distance while achieving swarm cohesion. Compared to one-hop AFCA, the adaptive flocking rules using two-hop mobility in AFCA produce a smoother and fairer travel distance for each UAV to reach the stability state owing to the advantage of velocity synchronization and extended knowledge of neighboring UAVs positions. Because UAV propulsion energy is proportional to the traveling distance, a better TDF in AFCA creates a balance in the UAV energy consumption.

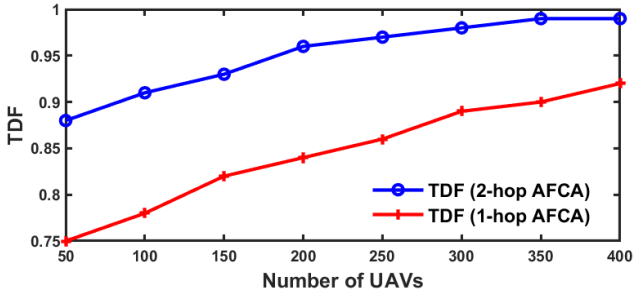


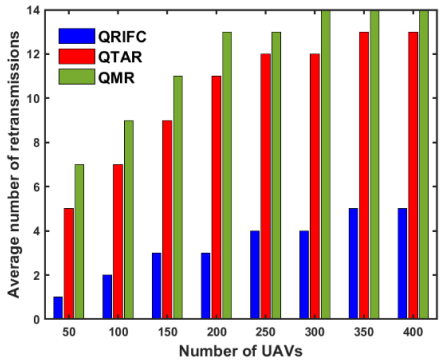

Figure 4.7 TDF versus the number of UAVs.

Figure 4.8 shows the network performance of QRIFC compared to QMR and QTAR with respect to the number of UAVs. According to Figure 4.8(a), QRIFC produces a significantly lower ANR than QTAR and QMR for two reasons. First, owing to the joint consideration of predictive maximum–minimum LD and PTS metrics, up to two-hop neighbors help to select a better stable path because path stability and LD are highly coupled. Second, the AFCA constructs a stable FANET topology with optimal node density, which is connected to the BS by adaptively adjusting the flocking rules and their weights. QTAR and QMR have higher ANR because they do not consider path stability, and they select the next relay based only on the PTS without controlling the relative mobility. Hence, both QMR and QTAR encounter more link breakages. Figure 4.8(b) shows the PDR with respect to the number of UAVs. QRIFC has a higher PDR compared to QTAR and QMR because it requires fewer ANR.

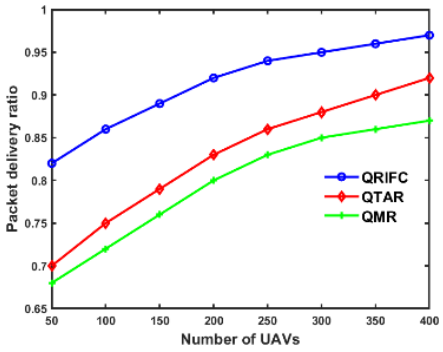
According to Figure 4.8(c), QRIFC exhibits lower AETED than QMR and QTAR for two reasons. First, the relay UAV selection according to the maximum positive PTS is computed based on the ratio between the distance progress toward the BS and PTT. QRIFC precisely computes PTT using the link delay and link packet error rate, which is directly related to the link SINR. Second, according to the imposed safety distance and transmission range constraint of UAVs in AFCA, the adaptive adjustment of UTU distances helps to maintain an optimal node density, which significantly reduces MAC layer contention. Both QMR and QTAR compute the PTS by considering only the link delay (MAC and queuing

delay). However, they do not consider the link packet error rate and SINR, which is a very important metric in a dynamic network.

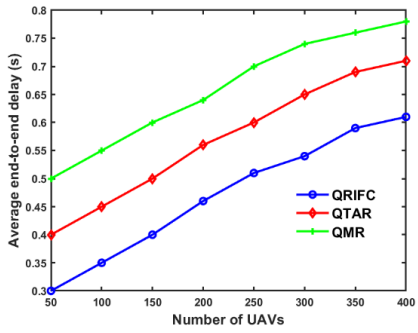
Figure 4.8(d) shows the control overhead with respect to the number of UAVs. QTAR exhibits a very high control overhead owing to the savings of two-hop neighbor list. Although QRIFC saves up to two-hop neighbor information compared to QTAR, it has less control overhead because it adaptively controls the hello interval according to the minimum LD found within the one-hop vicinity rather than broadcasting the hello packet at a fixed interval. Additionally, the adaptive weighted flocking rules in the AFCA maximize the LD with neighboring UAVs by controlling the relative distance, direction, and velocity, which helps to maintain the hello interval at a more optimal level. Because QMR maintains only one-hop neighbor information, it has less control overhead compared to the other routing protocols. However, owing to the broadcast of the hello packets at a fixed interval, the QMR control overhead also raises slowly with the increased number of UAVs although it employs only one-hop neighbor information.



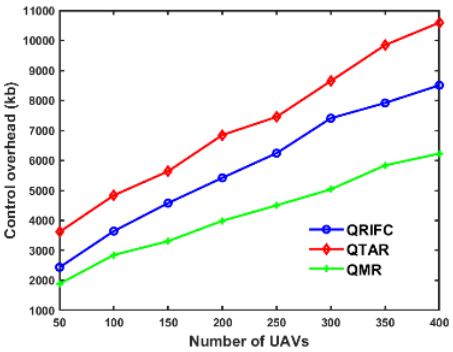

(a) ANR




(b) PDR




(c) AETED

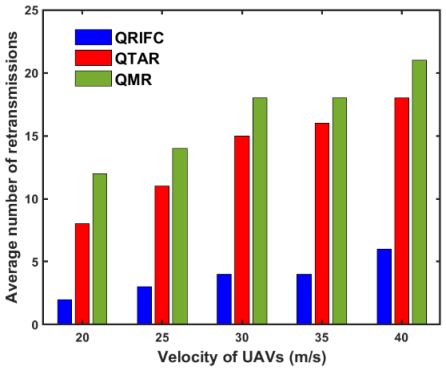



(d) Control overhead

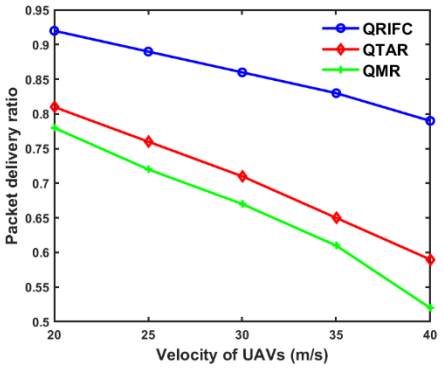
Figure 4.8 Network performance with respect to the different number of UAVs.

Figure 4.9 shows the network performance with respect to different attainable maximum velocities in a swarm of 200 UAVs. In Figure 4.9(a), QRIFC significantly outperforms QMR and QTAR in terms of ANR with increasing maximum attainable velocities for three vital reasons. First, the adaptive weighted flocking rules in AFCA produce optimal mobility for each UAV according to the velocity synchronization with neighboring UAVs, which reduces the possibility of link breakages. Second, the consideration of the maximum–minimum 3D LD metric in the reward function helps in the selection of a more stable relay path. Third, QRIFC can trigger the TA to adjust the weight of the flocking rules to improve the LD proximity with neighboring UAVs if it detects higher link breakages and higher PTT while relaying data packets to the BS. Owing to the above advantages, QRIFC has a better PDR than QMR and QTAR, as shown in Figure 4.9(b). According to Figure 4.9(c), QRIFC exhibits less AETED with increased velocity owing to its adaptive exploration strategy according to NALD. In QRIFC, if the UAV experiences a high degree of change in relative mobility with its neighbors, it explores the next relay that offers the maximum PTS to efficiently forward data packets to the BS.

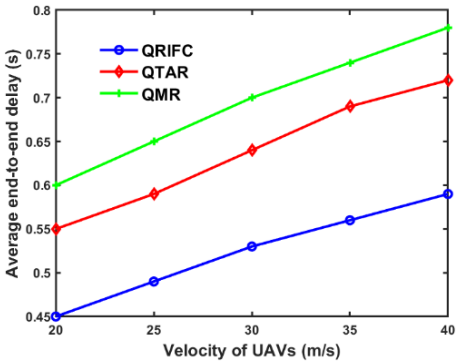
According to Figure 4.9(d), QTAR exhibits higher control overhead compared to the other parameters because of the increment in the velocity states of UAVs, and the hello interval increases proportionally to keep updating the two-hop neighbor list. In contrast, owing to the adaptive weighted flocking rules and the velocity synchronization in the AFCA, the QRIFC efficiently maintains a stable relative distance and relative velocity synchronization with neighboring UAVs, which subsequently maximizes the minimum LD with neighboring UAVs and maintains the control overhead at an optimal level. QMR shows less control overhead because it broadcasts hello packets at a fixed interval, not concerning the relative mobility states. As a result, QMR topology prediction accuracy degrades, subsequently affecting its network performance.



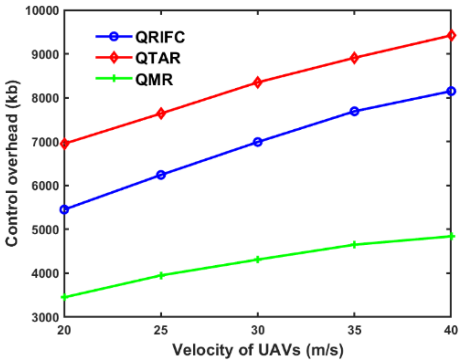

(a) ANR




(b) PDR




(c) AETED




(d) Control overhead

Figure 4.9 Network performance with respect to different UAV velocities.

In Figure 4.10, the QMR exhibits less energy consumption (higher NRE) because it considers one-hop neighbor table and optimizes the energy consumption by selecting a relay UAV with a higher RE.

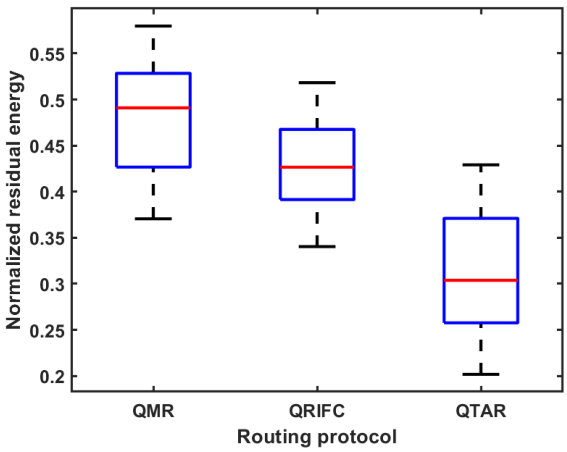



Figure 4.10 NRE of UAVs for the different routing protocols.

Here, the horizontal red line within each box represents the median of NRE for each routing protocol. Although QRIFC utilizes the two-hop neighbor information to generate UAV mobility and makes routing decision to improve the communication performance, it shows significant energy improvement compared to QTAR and is almost close to QMR owing to the advantage of fewer ANR. In addition, QRIFC considers the UAV RE level while selecting relay. Also, the collective motion generated by AFCA ensures higher TDF. Considering the NRE status distribution of UAVs in each box, QRIFC exhibits a greater

balance in energy consumption compared to QMR and QTAR owing to the balance in both UAV propulsion energy and communication energy, which guarantees more topological stability.

Now QL convergence in QRIFC is addressed, and the trade-off between exploration and exploitation is discussed by comparing different exploration strategies adopted in QMR, QTAR, and benchmark techniques such as UCB and ∈-greedy method. Figure 4.11 shows the average reward with respect to the number of iterations for different state exploration strategies for forwarding data packets to the BS. The proposed QRIFC performed exploration based on two-hop NALD and PTS values. In contrast, QTAR performed exploration using only a two-hop PTS, whereas QMR performed exploration using a one-hop PTS. UCB controls the exploration rate by considering the sum of the average cumulative reward and the number of times a particular action is taken within a particular time. In the ∈-greedy strategy, the exploration is selected according to a randomly chosen value of ∈ usually considering a 10% probability.

According to Figure 4.11, the exploration strategy based on two-hop NALD and PTS in QRIFC provided faster convergence and achieved better rewards compared to the other parameters, indicating that it can explore a better relay state for data packets to reach the BS for three major reasons. First, the PTS is the ratio between the progress of the Euclidean distance toward the BS and the PTT. Thus, considering the 3D mobility of UAVs, state exploration based only on UTU distances is not optimal because UAV mobility prediction considering the relative velocity, relative distance, and flying directions provides better stability in state exploration, which is only possible by estimating the predictive LD. Second, considering the dynamic topology in FANETs to precisely update the link Q-value, both QMR and QTAR update the discount factor for each neighbor link based on the degree of change in the neighboring set similarity, which may not deliver accurate link conditions. In contrast, QRIFC updates the discount factor according to the relative distance, as given in (4.29), to produce a precise Q-value by giving a higher discount to the neighboring UAVs, which satisfies the imposed separating distance and transmission range constraints. Third, considering the reward function, QRIFC considers path stability as LD, path delay as PTS, and energy status as the RE of the UAV. By contrast, path stability is not considered in either QTAR or QMR.

Two-hop PTS-based exploration in QTAR provides a better reward compared to the QMR one-hop PTS strategy owing to the advantages of extended knowledge of the time-varying topology. However, both QMR and QTAR exploration strategies converge with lower reward. Then, because the UCB and ∈-greedy perform explorations based on the number of times a specific action is chosen and a random probability without considering the network condition, both give the lower reward. However, their reward slowly increases

with the number of iterations because their learning is time dependent. In Figure 4.11, the QRIFC exploration strategy converges to a maximum reward after approximately 270 iterations whereas, according to Figure 4.11, the AFCA flocking control achieves swarm cohesion after approximately 240 iterations. Thus, the stability of the AFCA mobility controller significantly enhances the routing performance of FANETs because it maximizes LD with neighbors while performing the collaborative mission.

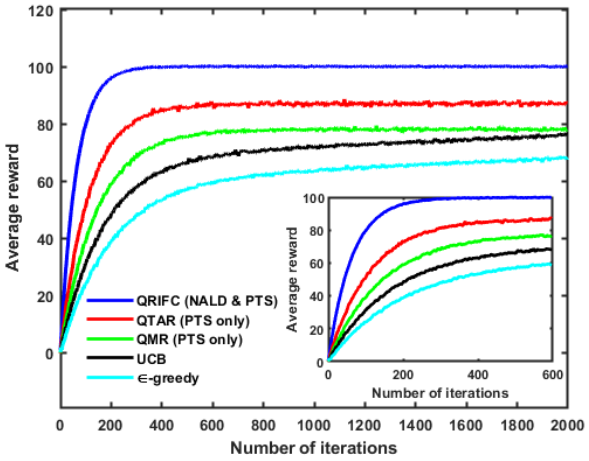



Figure 4.11 Average reward versus the number of iterations.

### 4.4.4 Summary on performance improvement

According to the earlier performance comparison study, the performance improvement over the conventional schemes is summarized and discussed in this subsection. In the scalability test, QRIFC shows 21.28% and 40.16% less AETED compared to QTAR and QMR. QRIFC also provides 9.30% and 12.85% better average PDR compared to QTAR and QMR, respectively. Even though QRIFC exhibits 27.12% higher average control overhead compared to QMR, QRIFC provides 24.55% less average control overhead compared to the QTAR. Similarly, QRIFC exhibits 36.36% better average NRE level (less energy consumption) of UAVs in comparison to QTAR and shows 20.45% less average NRE level (higher energy consumption) in comparison to QMR.

In the velocity increment test, QRIFC provides 21.76% and 32.45% less AETED compared to QTAR and QMR, respectively. Similarly, QRIFC shows 17.95% and 23.08% better average PDR in comparison to QTAR and QMR, respectively. Even though QRIFC offers 19.56% less average control overhead than QTAR, it exhibits 38.58% higher average control overhead in comparison to QMR. Owing to the significant improvement in AETED and PDR, the reasonable cost in control overhead and the energy consumption are acceptable.

Thanks to the AFCA mobility controller, the new multi-objective reward function and TA triggering method in QRIFC contribute to such performance improvement.

## 4.5 Conclusion

In this paper, the relation between swarm mobility control, delay, and routing policy has been addressed to significantly improve the communication performance of FANETs. The AFCA controls the relative mobility with neighboring UAVs and offers a relatively stable state to the QRIFC. Consequently, efficient data routing is performed by using a new reward function in QL, jointly considering the predictive 3D LD, PTS, and UAV RE. This integrated routing strategy with adaptive flocking control provides faster swarm cohesion, high PDR, shorter end-to-end delay, less retransmissions, and more balance in energy consumption while incurring optimal control overhead, compared to the existing routing protocols.

# 5. Joint Trajectory Control, Frequency Allocation, and Routing

## 5.1 Introduction

Due to the flexibility in 3D positioning adjustment, maneuverability, wider coverage and connectivity, and high survivability, collaborative UAVSNs have potential in both military and civilian applications. For instance, an autonomous UAVSN can be deployed to perform real-time tasks, such as sensing and monitoring, over a post-disaster area and wild-fire monitoring [162] by transmitting real-time high-resolution video or 3D images to a ground BS. UAVSNs can also be utilized to function as an aerial base station that provides emergency communication services to ground users and collects data from ground-based IoT devices.

Thus, when executing missions, UAVSNs require collaborative trajectory control to maximize coverage and ensure the QoS (i.e., high data rate and minimal delay) in both U2U and U2BS links [159]. Since UAVs have limited transmission power, data packet transmission from remote UAVs to BS requires a multi-hop path that involves a series of relay UAVs. However, due to the highly dynamic time-varying topology and limited energy, packet routing from UAVs to the BS suffers frequent link breakages, higher delays, routing loops, and energy holes. Although the LoS access in U2U link ensures communication quality, its exposure during simultaneous transmission generates strong mutual interferences. Consequently, the performance of packet routing protocol depends on multiple link quality metrics, such as link SINR, relative trajectory knowledge, queuing delay, and available RE of a relaying UAV.

To achieve high SINR, trajectory control according to physical layer transmission range and frequency resource allocation in the MAC layer are prerequisites. Additionally, the relative mobility prediction metric LD can be utilized to alleviate the effect of the highly mobile time-varying topology [34], [120]. The LD offers a predictable time at which two adjacent UAVs remain within their communication range. In UAVSNs, the UAVs are required to periodically adjust their position, velocity, and flying directions based on the mobility of nearby UAVs to avoid chaotic movements and maintain stable LD. UAVSNs topology should be self-healing to retrieve the connectivity with remaining UAVs in the case of UAV failure or departure due to energy depletion. The trajectory of UAVs should be smooth and preserve fairness in travel distance to balance the propulsion energy consumption.

To overcome the above challenges, researchers have attempted integrating the behavior of SI, such as bird flocks or fish schools, to design self-organized, self-healing, and

distributed collaborative trajectories [40], [81], [101]. The aviation of UAVs in a dynamic environment can preserve a robust topology by generating collective motion according to the behavioral rules of the Reynolds motion model [163]. Trajectory control of UAVs based on their physical layer transmission range inspired by behavior-based motion can obtain the optimal aerial node density. It is achieved by imposing a constraint on the minimal separating distance and maximum inter-UAV distances with neighboring UAVs. Such trajectory control guarantees aerial coverage, safety distance to avoid inter-UAV collisions, and satisfies the communication range constraint. Moreover, it can significantly reduce mutual interference because although a higher node density increases the connectivity, it increases the mutual interference and competition between neighboring UAVs to access the shared medium. Consequently, the optimal allocation of frequency resources in the MAC layer can significantly reduce the mutual interferences. In this study, joint trajectory control, frequency resource allocation, and packet routing (JTFR) is proposed by leveraging the cross-layer design to efficiently design packet routing in UAVSNs.

Nevertheless, behavior-based motion obtains mobility only at the next timeslot based on the mobility at the current timeslot. In a practical scenario, wind disturbances and GPS localization errors can severely affect the trajectory of UAVs. The uncertainties in communication (i.e., congestion, delay, and interference) can make UAVs compute the motion component vector with the outdated mobility of neighbor UAVs. However, each UAV cannot control its trajectory within the boundary of the 3D mission area by solely depending on this behavior-based motion. Thus, an alternative approach is required to precisely estimate the mobility. More specifically, the historical information of the UAV trajectory generated by the motion model can be utilized to precisely predict the mobility of the UAV. Subsequently, allocating the frequency blocks according to the historical frequency state of each UAV and its neighboring UAVs can result in selecting a better frequency state to avoid mutual interference. Consequently, based on the historical information of the neighboring link SINR, delay, and relative trajectory knowledge, the respective source UAVs can select a better next hop to relay a packet toward the BS. Therefore, in the proposed JTFR, the joint consideration of trajectory control in continuous space, frequency resource allocation, and relay UAV selection transforms into a complex collaborative sequential decision-making problem.

Recently, RL has been widely used for designing the trajectory of UAVs [164], allocating resources, and selecting relay UAVs for packet routing [133], [138] owing to the advantage of less computational complexity and less modeling difficulties. Data-driven deep reinforcement learning (DRL), which comprises both deep learning and RL can efficiently solve sequential decision-making problems by adopting the MDP. Here, each learning agent iteratively selects an action based on the current state to maximize its cumulative reward by interacting with the dynamic environment. QL, which can only handle problems with small-

scale discrete state-action space, is the most conventional RL algorithm. Although the DQN can tackle large state space problems by using a Q-value function approximator, it can only deal with low dimensional discrete action spaces [121]. Therefore, actor-critic learning is introduced to obtain the optimal policy in a continuous action space, where the actor maps the input state to a stochastic action policy by leveraging the policy gradient method [165]. The critic network then evaluates the action by generating a Q-value function. An off-policy actor-critic framework based on the DDPG, which comprises the deterministic policy gradient and DQN, efficiently deals with the large action and state space [138], [166]. The utilization of target networks for actor, critic, and replay buffer further improves the training stability in DDPG. However, the single agent DDPG attempts to independently maximize its own reward without considering the influence of neighboring agents' state-action. Thus, the environment appears non-stationary from the perception of any individual agent, and it result in an unstable learning process [167].

Fortunately, the extension of single-agent DDPG to multi-agent DDPG (MA-DDPG) can solve these problems by adopting centralized training and distributed execution [168], [169]. In MA-DDPG centralized training, the critic network utilizes the state-action of each agent to generate a global Q-value function to train the actor-network. However, collecting the global information of large-scale UAVSNs in a centralized server increases the computational complexity, information exchange, and is less scalable. Moreover, the highly dynamic topology can make the information collected by the centralized server easily outdated, which directly affects the training process. Additionally, purely centralized training should exchange the necessary information with a centralized server, which can be vulnerable in terms of security. Conversely, distributed cooperative training can overcome such challenges. However, only considering the observation from the one-hop neighboring agent may trap in local optima. Furthermore, continuous changes in the concurrent learning policy of nearby agents may trigger unstable learning in the multi-agent scenario [167].

The collaborative UAVSNs are similar to a multi-agent system, where each UAV acts as a learning agent. A major challenge in implementing collaborative UAVSNs is that the neighboring set for each agent is not identical and changes with time. Actions taken by the neighboring agent significantly impact the reward of a particular agent. For instance, if two neighboring UAVs select the same frequency band, it generates mutual interference. Similarly, if most neighboring UAVs select the same UAV to relay data packets, it can create network congestion and an increased queuing delay. If the neighboring UAVs randomly choose velocity and flying direction, the LD may reduce significantly, which results in link breakages. Thus, the global Q-value computed by the centralized server is not appropriate for all agents. In JTFR, the decision-making for each UAV to control the trajectory, allocate frequency resources, and select the relay UAV is highly coupled with other UAVs. Notably, each neighboring agent has considerable influence [170]. In JTFR,

multi-agent interaction is required to control the trajectory of UAVs, allocate frequency resources, and select relay UAVs in a large state and action space. Thus, distributed MA-DDPG (DMA-DDPG) is envisioned as the best option to efficiently solve this cooperative sequential decision-making problem.

In MA-DDPG, the actor-network solely depending on the fully connected layer (FCL) cannot deal with time-series data. In UAVSNs, historical information needs to be exploited to make a sequential decision to control the trajectory, allocate frequency resources, and select a relay UAV. Fortunately, recurrent neural networks, such as the LSTM-based actor-network, can store historical sequential information and utilize the information to predict the more precise state in the next timeslot by mining the temporal relationship in the time-series data [171]. The critic network using only the FCL, or even LSTM cannot estimate the value function to adaptively adjust its actor action policy by prioritizing its neighbors in a sequential adaptive weighted manner. Moreover, FCL or LSTM-based critic network has less scalability and slow convergence. Thus, to overcome these challenges, a multi-head attention mechanism is utilized for each agent critic network to adaptively pay attention to its neighbors by generating attention weights using dot product similarity of their state-action features. The adaptive adjustment of each agent policy according to changes in the neighboring agent policy help to avoid environmental non-stationarity, which improves learning stability. Moreover, multi-head attention introduces parallelization in the critic value-function computation in multi-agent interaction, which delivers faster convergence.

The major contributions of this study can be summarized as follows:

- We propose JTFR by formulating and solving a link utility maximization problem for UAVSNs to route packets toward BS by jointly considering UAV trajectory control, frequency resource allocation, and relay UAV selection. The link utility contains a link stability metric defined by predictive 3D maximum-minimum LD, link SINR, queuing delay, and relay UAV RE level under several constraints.

- The adaptive DMA-DDPG-based algorithm coupled with swarming behavior is proposed to obtain the optimal link utility. To adopt the dynamic topology and avoid local optima, the MDP observation state of each UAV comprises both one-hop and two-hop neighbors' dynamic state. Each UAV actor network is represented by three LSTM-based state representation layers (SRLs) and an FCL. The key state parameters of dynamic UAVSNs are embedded into the LSTM-based SRLs to provide better state representation to the actor FCL by extracting the temporal continuity in the historical dynamic states of the time-varying topology. It supports the actor FCL to achieve a better deterministic policy by mapping the SRL output toward the optimal action.

- A multi-head attentional critic network is designed for each UAV to effectively train each agent actor network and adaptively adjust the policy of each agent actor network in a multi-agent dynamic environment. It precisely estimates the value function for the action taken by the actor network by selectively assigning attention weights to its neighboring agents according to their influence, which is used to further minimize critic loss and update the actor network. Moreover, it provides better scalability, learning stability, and accelerates convergence for optimal decision-making.

- Extensive simulative analysis shows that the proposed DMA-DDPG-based JTFR outperforms existing routing protocols in terms of traveling distance fairness, PDR, average end-to-end delay, and energy consumption.

## 5.2 System Model

We consider $\mathcal{U} = \{1, 2, \cdots, u\}$ as a set of quad rotor UAVs having GPS, IMU, camera, and wireless interface. They are deployed to execute surveillance mission over a three-dimensional (3D) post-disaster mission area, as shown in Figure 5.1. The dimension of the 3D mission area is bounded by $(x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}, z_{\min} \leq z \leq z_{\max})$. To track the mission, the overall surveillance time $\mathcal{T}$ is divided into $t$ equal timeslots represented as $\mathcal{T} = \{0,1,2, \cdots, t\}$, where the length of each $t$ is adequately small denoted as $\delta_t$. Hence, UAVSNs topology can be expressed as a time-dependent undirected graph $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$, where $\mathcal{V}(t) \in \{\mathcal{U}(t) \cup BS\}$ represents the vertex set comprising the active UAV set $\mathcal{U}(t)$ and a location-fixed BS. An emergency response vehicle acts as the BS, which can function as a mission control center and edge server. Each UAV $u_i$ can localize its position $\vec{p}_i(t) \in (x_i, y_i, z_i)$ by using GPS and being aware of the position of the BS, $\vec{p}_{BS}$.

The communication range of each UAV is separated into two regions: the repulsion range $R_r$ and attraction range $A_r$. Therefore, to satisfy the safe distance and communication range constraints, the distance between two neighboring UAVs $d_{ij}(t)$ must be retained within $R_r \leq d_{ij}(t) \leq A_r$. If $d_{ij}(t) \leq A_r$, a direct edge $\mathcal{E}(t)$ between two neighboring UAVs is considered. Consequently, the source UAV $u_i$ selects a series of relay UAV represented as $(u_j, u_k \cdots, BS)$ to transmit packets toward BS, as illustrated in Figure 5.1.

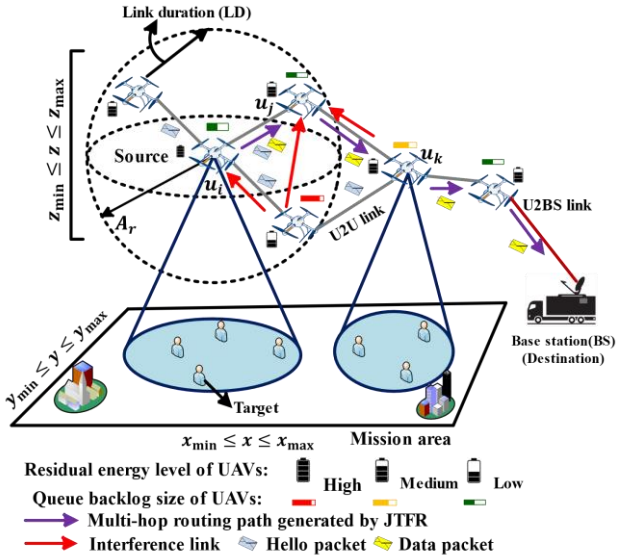



Figure 5.1 An example of UAV swarm networks.

*Notations*: $\|\bullet\|_2$ represents the Euclidean norm, $\|\bullet\|$ represents the absolute value, and $|\bullet|$ represents the cardinality of a set

### 5.2.1 Channel Model

Owing to the high altitude and 3D mobility adjustment, the U2U links and U2BS links are dominated by LoS links. Thus, the channel gain between two UAVs $(u_i, u_j)$ in free-space path can be stated as $g_{ij}(t) = \rho_0 d_{ij}(t)^{-\alpha}$, where $\rho_0$ denotes the LoS channel gain within a reference distance of $1\ m$, and $\alpha$ represents the path-loss exponent. For a given transmit power $P_i^{tx}$ from UAV $u_i$, the received power at UAV $u_j$ can be expressed as $P_{ij}^{rx}(t) = P_i^{tx}(t) g_{ij}(t)$.

The network bandwidth divided into $f_k$ orthogonal frequency bands can be denoted as $f = (f_1, f_2, \cdots, f_K)$, where the frequency band $k$ is between $(1 \le k \le K)$. The bandwidth of each frequency band $f_k$ is equal and denoted by $B$. Each UAV selects a relay UAV and transmits a packet from its queue buffer by following the first in first out approach by choosing a transmission frequency band $f_k$. The index of the frequency band selected by UAV $u_i$ is represented as $f_{k,i}(t)$. If a UAV $u_i$ selects frequency band $f_k$ to transmit a packet to UAV $u_j$ at time $t$, then the corresponding binary channel association $\phi_{f_{k,i}}(t) = 1$; otherwise, $\phi_{f_{k,i}}(t) = 0$. Since UAVs share the frequency band $f_k$ during simultaneous transmission, the interference from the neighboring UAV $u_{\hbar}$ $(\hbar \ne i, j)$ to UAV $u_j$ over the frequency band $f_k$ can be expressed as

$$I_{\hbar j}^{f_k}(t) = \sum_{\hbar \neq i,j} \phi_{f_{k,\hbar}}(t) P_{\hbar j}^{tx}(t)\, \rho_0 d_{\hbar j}(t)^{-\alpha}, \tag{5.1}$$

where $\left[\phi_{f_{k,\hbar}}(t)\right]$ represents an $\mathcal{U}_{\hbar}(t) \times K$ binary frequency band paring matrix. Here, $\mathcal{U}_{\hbar}(t) = [u_1, u_2, \cdots, u_{\hbar}]$ represents the set of active UAVs within the one and two-hop neighborhood of UAV $u_i$ that performs simultaneous transmissions at time $t$. The SINR $\gamma_{ij}(t)$ at UAV $u_j$ can be obtained as

$$\gamma_{ij}(t) = 10 \log \frac{P_{ij}^{rx}(t)}{I_{\hbar j}^{f_k}(t) + \sigma^2(t)}, \tag{5.2}$$

where $\sigma^2(t)$ represents additive white Gaussian noise power. U2U links can be established successfully if $\gamma_{ij}(\mathrm{t}) \geq \gamma_{th}$, where $\gamma_{th}$ represents SINR threshold. Thus, the maximum communication range for UAV $u_i$ to communicate with UAV $u_j$ is $d_{ij}(t) \leq d_{ij}^{th} = \left[\frac{\rho_o P_i^{tx}(t)}{\left(I_{\hbar j}^{f_K}(t) + \sigma^2(t)\right) 10^{\frac{\gamma_{th}}{10}}}\right]^{1/\alpha}$. For each UAV with an omnidirectional antenna, the attainable communication range can be represented as a sphere with radius $A_a = d_{ij}^{th}$. The data transmission rate between two UAVs $C_{ij}(t)$ is estimated as $C_{ij}(t) = B \log_2\left[1 + \gamma_{ij}(t)\right]$.

### 5.2.2 Delay Model

For each UAV $u_i$ the queue backlog size $q_i(t+1)$ is represented as follows:

$$q_i(t+1) = \min\left[\left[q_i(t) - D_i(t)\right] + A_i(t), q_{\max}\right], \tag{5.3}$$

where $D_i(t) = C_{ij}(t)\delta_t$ represents the amount of packets that were successfully transmitted from the queue buffer to the next relay UAV. $A_i(t)$ represents the process of new packet arrival in the queue buffer at timeslot $t$. $q_{\max}$ represents maximum queue buffer size. Each source UAV $u_i$ prefers the next relay UAV $u_j$ having a small queue backlog size $q_j$ given by (5.3) to avoid long queuing delay and network congestion. Considering a sufficiently large queue buffer size, we adopt M/M/1 queuing, where the packet arrival rate $\mathcal{A}_i$ obeys the Poisson process. Thus, the average waiting time of each packet in the queue can be approximated as $t_q = {}^{1}/_{(\mathcal{D}_i - \mathcal{A}_i)}$, where $\mathcal{D}_i$ denotes the average packet service rate. Finally, the total time required to successfully reach the next relay UAV can be approximated as follows:

$$t_d = t_q + \frac{P_{size}}{C_{ij}(t)} \tag{5.4}$$

### 5.2.3 Energy Model

UAV energy consumption has two major portions: propulsion and communication energy consumption. Generally, the propulsion energy consumption is considerably higher than the communication energy. The propulsion power $PP_i$ of UAV $u_i$ generates thrust $T_H$ to fly in the air by overcoming drag forces and gravity. The $T_H$ produced by UAV rotors is a function of the velocity $\vec{v}_i$ and acceleration $\vec{a}_i$. Thus, $PP_i$ is a function of $\vec{v}_i$ and $T_H$. The $T_H$ and $PP_i$ for quadrotor UAVs are obtained according to [138]. $PP_i$ is proportional to the traveled trajectory of each UAV. The communication energy consumption depends on the transmitted packet size $P_{size}$ at each timeslot. The transmitting energy $E_i^{tx}$ can be computed as $E_i^{tx} = \frac{nP_i^{tx}P_{size}}{C_{ij}(t)}$, where $n$ denotes the number of retransmissions. For a given maximum energy level $E_{\max}$, the RE level $E_i(t+1)$ of each UAV at the next timeslot can be tracked as follows:

$$E_i(t+1) = E_{\max} - \sum_t \left[\left\{PP_i(t)\delta_t + E_i^{tx}(t)\right\}\right] \tag{5.5}$$

When the $E_i(t+1)$ is less than the threshold $E_{th}$, UAV can return to BS for battery replacement before rejoining the aerial network.

### 5.2.4 Problem Formulation

Owing to the limited communication range, the source UAV $u_i$ selects a series of relay UAVs to relay the data packet toward the BS. Hence, the end-to-end path becomes $(u_i, u_j, u_k \cdots, BS)$, which comprises $m$ hops. To avoid the detour, each UAV $u_i$ selects relay UAV $u_j \in N_i^1(t)$ in the direction of $\Delta_{ij} = \left[\|\vec{p}_i - \vec{p}_{BS}\|_2 - \left\|\vec{p}_j - \vec{p}_{BS}\right\|_2\right] > 0$. Additionally, during forwarding, the link utility $LU_{ij}$ given in (5.6) is maintained, which jointly considers LD to avoid link breakage, link SINR to achieve highest data rate, small queue backlog size to avoid network congestion or delay, and highest RE energy level of relaying UAV to avoid energy holes. Notably, each term in $LU_{ij}(t)$ is normalized by utilizing the corresponding maximum value found within the neighbor information.

$$LU_{ij}(t) = a\frac{LD_{ij}}{\max LD_{ij}} + b\frac{\gamma_{ij}}{\max \gamma_{ij}} + ce^{-q_j} + d\frac{E_j}{E_{\max}}, \tag{5.6}$$

where $a + b + c + d = 1$, represents the weight of each link quality metric.

The $LU_{ij}$ maximization problem in the end-to-end path is represented as

$$\max \sum_{j=0}^{m-1} LU_{ij}, \tag{5.7}$$

Subject to the following constraints:

$$R_r \le d_{ij}(t) \le A_r, \tag{5.7A}$$

$$\min LD_{ij} > t_d, \tag{5.7B}$$

$$-a_{\max} \le \|\vec{a}_i(t)\| \le a_{\max}, \tag{5.7C}$$

$$\|\vec{v}_i(t)\| \le v_{\max}, \tag{5.7D}$$

$$(x_{\min} \le x \le x_{\max}, y_{\min} \le y \le y_{\max}, z_{\min} \le z \le z_{\max}), \tag{5.7E}$$

$$f = (f_1, f_2, \cdots, f_K), \tag{5.7F}$$

$$\gamma_{ij} \ge \gamma_{th}, \tag{5.7G}$$

$$q_j(t+1) \le q_{\max}, \tag{5.7H}$$

$$E_i(t+1) \le E_{th}, \tag{5.7I}$$

Here, (5.7A) ensures optimal node density by satisfying the minimum separating distance $R_r$ and maximum communication range constraint $A_r$. (5.7B) helps to avoid link breakage during data transmission by ensuring packet traveling time $t_d$ is sufficiently larger than LD. (5.7C) and (5.7D) expresses that the acceleration and velocity should not exceed the maximum threshold. (5.7E) indicates that UAVs should not fly away from the bounded 3D mission area. In particular, the altitude constraint is provided to avoid the ground obstacles. (5.7F) represents available frequency resources. (5.7G) represents the SINR constraint in the U2U links. (5.7H) represents the queue backlog size of the relaying UAV, which should not exceed the maximum buffer size to avoid packet loss due to buffer overflow and network congestion. Finally, (5.7I) represents the RE level constraint for UAVs to stay in the air. According to problem (5.7) and its constraints (5.7A)−(5.7I), $LU_{ij}$ maximization is tightly coupled with trajectory control, frequency resource allocation, and suitable relay UAV selection. Here, sequential decision making is required using historical states of time-varying topology. Thus, we integrated behavior-based motion properties with an adaptive DMA-DDPG algorithm to efficiently solve JTFR problem (5.7) while satisfying all the constraints. The model-free DMA-DDPG does not require convexity to solve the complex optimization problem. By designing a multi-objective reward function with penalty terms, it can satisfy optimization objectives and constraints.

### 5.2.5 Behavior-Based Motion Model

Behavior-based motion obeys three rules: cohesion (attraction), alignment (velocity matching), and separation (repulsion). Each rule generates a motion vector, and the weighted addition of these motion vectors defines the mobility of UAVs. The motion rules solely based on the one-hop neighbor may create partition in UAVSN topology. Hence, two-hop

mobility information is utilized to maintain the connected UAVSN topology. Here, each UAV is treated as a particle with an initial position and velocity.

The cohesion rule $\overrightarrow{CR_i}(t)$ specifies each UAV attracted toward the average centroid position of its neighbor. Each UAV $u_i$ computes $\overrightarrow{CR_i}(t)$ using the relative position with one-hop neighboring UAV $u_j \in N_i^1(t)$ located within $R_r \leq d_{ij}(t) \leq A_r$ and two-hop neighbor UAVs $u_k \in N_i^2(t)$, as given in Figure 5.2. The $\overrightarrow{CR_i}(t)$ is computed as follows:

$$\overrightarrow{CR_i}(t) = w_1 \left[\frac{\sum_{j \in N_i^1(t)} \{\vec{p}_j(t) - \vec{p}_i(t)\}}{|N_i^1(t)|}\right] + w_2 \left[\frac{\sum_{k \in N_i^2(t)} \{\vec{p}_k(t) - \vec{p}_i(t)\}}{|N_i^2(t)|}\right], \tag{5.8}$$

where $w_1 + w_2 = 1$ indicates the weight of the one-hop and two-hop neighbor motion elements. To prioritize one-hop neighbor $w_1 > w_2$ is considered.

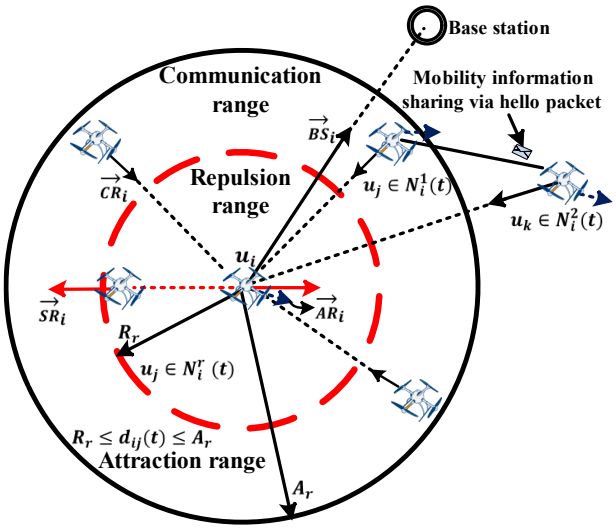



Figure 5.2 Behavior-based motion model of UAVs in UAVSNs.

The alignment rule $\overrightarrow{AR_i}(t)$ guides each UAV to perform velocity matching with neighboring UAVs. It helps UAVs to avoid chaotic movement. According to Figure 5.2, each UAV $u_i$ computes $\overrightarrow{AR_i}(t)$ by using relative velocity with one-hop $u_j \in N_i^1(t)$ and two-hop neighbor $u_k \in N_i^2(t)$ as follows:

$$\overrightarrow{AR_i}(t) = w_3\left[\frac{\sum_{j\in N_i^1(t)}\{\vec{v}_j(t)-\vec{v}_i(t)\}}{|N_i^1(t)|}\right] + w_4\left[\frac{\sum_{k\in N_i^2(t)}\{\vec{v}_k(t)-\vec{v}_i(t)\}}{|N_i^2(t)|}\right], \tag{5.9}$$

where $w_3 + w_4 = 1$ is the weight of the velocity alignment and $w_3 > w_4$ is considered.

The separation rule $\overrightarrow{SR_i}(t)$ guarantees the threshold of separating distance with nearby UAVs to prevent inter-UAV collision, as shown in Figure 5.2. Moreover, it assists UAVs to preserve an optimal routing path length, while forwarding packets toward BS. Each UAV $u_i$ computes $\overrightarrow{SR_i}(t)$ according to the relative distance with one-hop neighboring UAVs $u_j \in N_i^r(t)$ located within $d_{ij}(t) \leq R_r$, which is expressed as

$$\overrightarrow{SR_i}(t) = \frac{\sum_{j\in N_i^r(t)}[\vec{p}_i(t)-\vec{p}_j(t)]}{|N_i^r(t)|} \tag{5.10}$$

The above three rules assist UAVs to satisfy constraint (5.7A). Moreover, with above three motion rules to maintain connectivity with BS, an additional force is applied to detect the motion of each UAV $u_i$ as follows:

$$\overrightarrow{BS_i}(t) = [\vec{p}_{BS} - \vec{p}_i] \tag{5.11}$$

Finally, the resultant force $\vec{F}_i(t)$ also known as control input is obtained by applying the weighted sum of the above motion rules given by (5.8)−(5.11). In JTFR, we feed these motion rules to the actor neural network, which can adaptively adjust the force weights according to the network condition. Additionally, LSTM-based actor neural network can compute each force by using the historical information of relative distance and relative velocity with nearby UAVs, which provides higher accuracy in mobility prediction. According to the Newton's second law of motion, acceleration $\vec{a}_i(t)$ of UAV $u_i$ along with flying direction is computed as follows:

$$\vec{a}_i(t) = \frac{\left[\frac{\vec{F}_i(t)}{\|\vec{F}_i(t\|}\right]\tanh[\|\vec{F}_i(t)\|]a_{\max}}{m_i}, \tag{5.12}$$

where $m_i$ denotes the mass of each UAV. $\tanh(\bullet)$ represents the activation function to satisfy the constraint (7C). Subsequently, the velocity of each UAV in the next timeslot can be obtained as $\vec{v}_i(t+1) = \vec{v}_i(t) + \vec{a}_i(t)\delta_t$. To satisfy constraint (7D), $\vec{v}_i(t+1)$ is further adjusted as follows:

$$\vec{v}_i(t+1) = \begin{cases} \vec{v}_i(t+1), & \|\vec{v}_i(t+1)\| < v_{\max} \\ \left[\frac{\vec{v}_i(t+1)}{\|\vec{v}_i(t+1)\|}\right] \times v_{\max}, & \|\vec{v}_i(t+1)\| \geq v_{\max} \end{cases} \tag{5.13}$$

The position at next timeslots is updated as follows:

$$\vec{p}_i(t+1) = \vec{p}_i(t) + \left[\vec{v}_i(t)\delta_t + \frac{1}{2}\vec{a}_i(t)\delta_t^2\right] \tag{5.14}$$

Notably, the position vector $\vec{p}_i(t+1)$ can be decoupled into three corresponding coordinate axes along with their projection angles $\left(-\pi \le \sigma_i(t) \le \pi, -{}^{\pi}/_{2} \le \varphi_i(t) \le {}^{\pi}/_{2}\right)$ with horizontal $xy$-plane and $z$-axis, which can be further utilized to estimate the $LD_{ij}$. Let two neighboring UAVs $(u_i, u_j)$ have positions $p_i = (x_i, y_i, z_i)$ and $p_j = (x_j, y_j, z_j)$, velocities $v_i$ and $v_j$, and flying directions $(\sigma_i, \varphi_i)$ and $(\sigma_j, \varphi_j)$. Once time $\Delta t$ elapses, $d_{ij}(\Delta t)$ is given as follows:

$$d_{ij}(\Delta t) = \sqrt{(\mathcal{X} + A\Delta t)^2 + (\mathcal{Y} + B\Delta t)^2 + (\mathcal{Z} + C\Delta t)^2}, \tag{5.15}$$

where $\mathcal{X} = (x_i - x_j)$ , $\mathcal{Y} = (y_i - y_j)$ , $\mathcal{Z} = (z_i - z_j)$ , $A = (v_i \sin\sigma_i \cos\varphi_i - v_j \sin\sigma_j \cos\varphi_j)$, $B = (v_i \sin\sigma_i \sin\varphi_i - v_j \sin\sigma_j \sin\varphi_j)$, and $C = (v_i \cos\sigma_i - v_j \cos\theta_j)$. Since $LD_{ij}$ is bounded by the inter-UAV distance $d_{ij} = A_r$, substituting $d_{ij} = A_r$ in (15) yields

$$\Delta t^2(A^2 + B^2 + C^2) + \Delta t(2A\mathcal{X} + 2B\mathcal{Y} + 2C\mathcal{Z}) + \mathcal{X}^2 + \mathcal{Y}^2 + \mathcal{Z}^2 - {A_r}^2 = 0. \tag{5.16}$$

The positive root solution of (5.16) in terms of $\Delta t$ specifies the $LD_{ij}$, which can predict the link lifetime. The hello interval $HI_i$ for each UAV can be adjusted adaptively according to minimum $LD_{ij}$ found within one-hop neighbor to improve the topology prediction accuracy and optimize the control overhead, which can be expressed as follows:

$$HI_i = \psi \times [\min_{j \in N_i^1(t)} LD_{ij}], \tag{5.17}$$

where $\psi$ symbolizes the hello frequency rate, we set 0.5 in this study. At each $HI_i$ given in (5.17), each UAV broadcast hello packet that includes a hello sequence number, unique ID, mobility information (3D position, velocity, LD, RE, frequency state, SINR, and queue backlog size) of it and its neighbors. Based on the received hello packets, each UAV $u_i$ updates its one-hop $u_j \in N_i^1(t)$ and two-hop $u_k \in N_i^2(t)$ neighbor table and motion rules.

## 5.3 DMA-DDPG-Based JTFR Algorithm

In this section, a DMA-DDPG-based JTFR algorithm is proposed to obtain the optimal solution for the problem given in Section 5.2.4.

### 5.3.1 Necessary Preliminaries of DRL

In DRL, the agent learns to obtain an optimal policy to maximize a long-term cumulative reward by interacting with the dynamic environment without any prior knowledge. JTFR problem is treated as a multi-agent Markov game having a MDP tuple $(\mathcal{U}, O, A, R, O')$. Here, $\mathcal{U} \in u_i$ represents the set of UAVs acting as learning agent, $O \in$

$o_i(t)$ represents the observation state space (historical observation), $A \in a_i(t)$ represents the action space, $R \in r_i(t)$ represents the immediate reward after UAV $u_i$executes action $a_i(t)$, and $O' \in o_i{}'$ represents the next observation state at time $(t+1)$. In this game, each UAV $u_i$ obtains an optimal policy $\pi_i: o_i(t) \times a_i(t)$ to maximize an expected discounted cumulative reward $G_i(t) = \sum_{t=0}^{\infty} \lambda^t r_i(t)$, where $\lambda \in [0\ 1]$ represents the discount factor. The values of actions for sequential historical observations are measured by utilizing the state-action value function known as the Q-value. The Q-value is formulated as $Q\big(o_i(t), a_i(t)\big) = \mathbb{E}(G_i(t)) = \mathbb{E}[r_i(t) + \lambda G_i(t+1)] = \mathbb{E}[r_i(t) + \lambda Q(o_i', a_i')]$. In JTFR, the environmental state transition, specially the behavior-based motion, updates the mobility of each UAV in the next timeslot based on the mobility in the current timeslot. This property satisfies the Markov property and can be easily integrated with MDP formulation, including most recent historical observation states, to efficiently solve the JTFR.

### 5.3.2 MDP Formulation for JTFR

• Observation state space: Each UAV $u_i$ observation state $o_i(t)$ comprises three components. The first component $o_i^1(t) = \{\overrightarrow{CR}_i, \overrightarrow{AR}_i, \overrightarrow{SR}_i, \overrightarrow{BS}_i\}$ encompasses cohesion, alignment, separation, and connectivity with BS rule given by (5.8)−(5.11). The second component $o_i^2(t) = \{\phi_{f_{k,i}}, \phi_{f_{k,k}}\}$ contains the frequency state $\phi_{f_{k,i}}$ of UAV $u_i$ and binary frequency band paring matrix up to two-hop neighbor $\phi_{f_{k,k}}$. Finally, the third component $o_i^3(t) = \{(LD_{ij}, LD_{jk}), \gamma_{ij}, E_j, q_j\}$ contains two-hop LD $(LD_{ij}, LD_{jk})$, SINR $\gamma_{ij}$, RE level $E_j$, and queue backlog size $q_j$ of one-hop neighboring UAVs $u_j$. Thus, the $o_i(t)$ is expressed as $o_i(t) = [o_i^1(t), o_i^2(t), o_i^3(t)]$.

• Action space: The action space $a_i(t)$ comprises three components. The first component is the control input $\vec{F}_i$. Then, $\vec{a}_i$, $\vec{v}_i$, and $\vec{p}_i$ of UAV $u_i$ is updated according to the motion model given by (5.12), (5.13), and (5.14), respectively. The second component is UAV $u_i$ selecting the frequency band $\phi_{f_{k,i}}$ to transmit data packet while avoiding mutual interference given by (5.1). The third component is relay UAV selection $u_j \in N_i^1(t)$ in the exploration direction $\Delta_{ij} > 0$, while maximizing $LU_{ij}$. Thus, $a_i(t) = [\vec{F}_i, \phi_{f_{k,i}}, u_j \in N_i^1]$.

• Reward: The reward function $r_i(t)$ is designed according to $LU_{ij}$ given by (5.6) and its constraints are considered as penalties. Thus, the first component in $r_i(t)$ is a reward for the maximum-minimum LD $r_{LD}(t)$ given by (5.16). In a multi-hop path, the minimum LD between two adjacent UAVs specifies the link lifetime. Thus, if there are several links to reach the destination BS from a particular UAV, the maximum of the minimum LD along with these multi-hop links returns the best stable link. Thus, $r_{LD}(t)$ for up to two neighbors $(LD_{ij}, LD_{jk})$ is computed as follows:

$$r_{LD}(t) = \frac{\max\limits_{j \in N_i^1(t), k \in N_i^2(t)} [\min\{LD_{ij}, LD_{jk}\}]}{\max[\min\{LD_{ij}, LD_{jk}\}]} \tag{5.18}$$

The second component is the reward for link SINR $r_{SINR}(t)$ given by (5.2). Each UAV selects the link with highest $\gamma_{ij}(t) \geq \gamma_{th}$ to achieve the highest data rate. If $\gamma_{ij}(t) < \gamma_{th}$, $r_{SINR}(t)$ is zero and computed as follows:

$$r_{SINR}(t) = \begin{cases} \frac{\gamma_{ij}(t)}{\max\limits_{j \in N_i^1(t)} \gamma_{ij}(t)}, & \gamma_{ij}(t) \geq \gamma_{th} \\ 0, & otherwise \end{cases} \tag{5.19}$$

The third component of the reward is $r_q(t)$ to ensure minimum queuing delay given by (5.3). Accordingly, each UAV selects the relay UAV that has smaller queue backlog size. The $r_q(t)$ is computed as follows:

$$r_q(t) = e^{-q_j(t)} \tag{5.20}$$

The fourth component of the reward is $r_E(t)$ to avoid energy holes. Each UAV selects the relay UAV that has highest RE level given by (5.5). If the relay UAV RE level does not satisfy constraint (5.7I), $r_E(t)$ is set to zero. Otherwise, $r_E(t)$ is computed as follows:

$$r_E(t) = \frac{E_j}{E_{\max}} \tag{5.21}$$

Finally, the total reward $r_i(t)$ is computed as follows:

$$r_i(t) = ar_{LD} + br_{SINR} + cr_q + dr_E - \mu_{lm} r_{lm} - \mu_{pl} r_{pl} - \mu_{mo} r_{mo}, \tag{5.22}$$

where $r_{lm}$, $r_{pl}$, and $r_{mo}$ represent positive constants as penalties for trapping in the local minimum, violating constraint (5.7H), and violating constraints (5.7A) and (5.7E), respectively. Accordingly, $\mu_{lm}$, $\mu_{pl}$, and $\mu_{mo}$ represent the binary coefficient for respective penalty terms, whose value turns into one, when associated constraints are violated, otherwise set to zero. The local minimum penalty term $r_{lm}$ considers three different cases. First, since JTFR selects the relay UAV in the direction of $\Delta_{ij} > 0$ with maximum $LU_{ij}$, it will detect it as local minimum if the relay UAV has no further relaying UAV within its communication range to forward the packet toward BS. Second, if the routing loop is detected by tracing the previously visited hops in the end-to-end path. Third, if link breakage occurs for violating constraint (5.7B) and UAV failure.

### 5.3.3 Adaptive DMA-DDPG for JTFR

As shown in Figure 5.3, each agent utilized an adaptive DMA-DDPG algorithm with cooperative training and distributed execution to solve JTFR. Each agent DMA-DDPG model consists of actor-critic neural network frameworks. To stabilize the learning process and make it convergent, the actor network and critic network of each agent consists of an online network and a target network, as shown in Figure 5.3. The target networks for both actor and critic have a similar neural network structure along with soft-updated parameters. The actor network is responsible for the approximate action policy and produce actions by mapping its own historical observation. The actor network contains three LSTM-based SRLs and an FCL, as shown in Figure 5.4. The details of the neural network structures of the actor network are discussed in Section 5.3.3.1.

The critic network evaluates the performance of the action by generating a Q-value function. In the proposed DMA-DDPG, we designed an adaptive multi-head attentional critic network that generates a Q-value of the actions taken by the actor network. It is achieved by considering the influence of the neighboring agents' state-action according to the generated attention weight, as depicted in Figure 5.5. The Q-value estimation in the critic network via cooperative training is briefly discussed in Section 5.3.3.2.

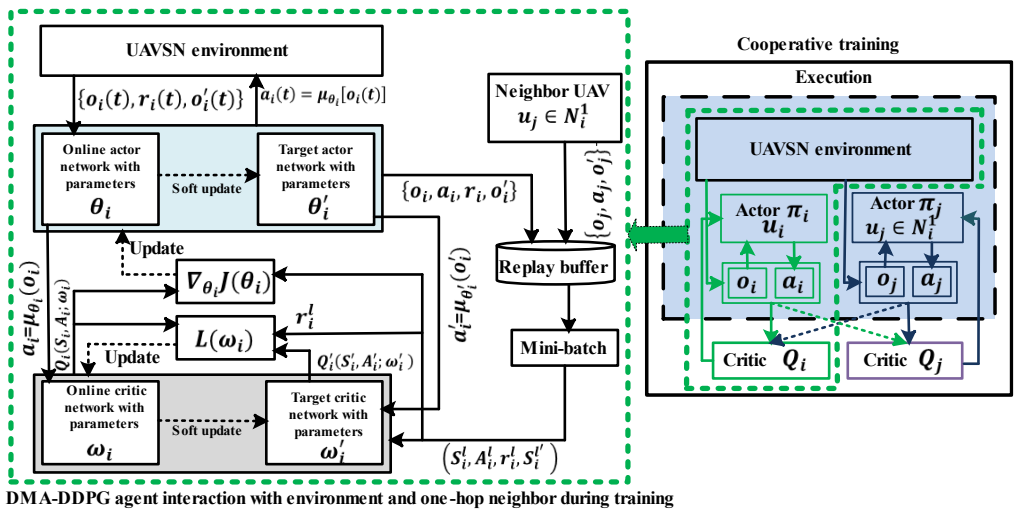



Figure 5.3 Adaptive DMA-DDPG training process and neural network architecture of an agent UAV.

For each UAV$u_i$, $\theta_i$ and $\omega_i$ represent the learnable parameters of the online actor and critic network. Similarly, $\theta_i'$ and $\omega_i'$ represent the learnable parameters of the target actor and citric networks, respectively. The actor action policy function is defined as $a_i(t) = \mu_{\theta_i}[o_i(t)]$ for the observation state $o_i(t)$ and parameter $\theta_i$. Since each agent UAV intends to maximize the long-term cumulative reward by obtaining an optimal action policy, the objective function for the actor policy can be expressed as $J(\theta_i) = \mathbb{E}_{\theta_i}[G_i(t)]$. Accordingly, the optimal action policy $\pi_i \approx \mu_{\theta_i}^*$ can be obtained by maximizing $J(\theta_i)$ with respect to $\theta_i$ as follows: $\mu_{\theta_i}^* = \arg\max_{\theta_i} J(\theta_i)$.

As discussed in Section 5.3.1, a state-action value function Q-value is utilized to evaluate the expected discounted cumulative reward $\mathbb{E}(G_i(t))$. In DMA-DDPG, the Q-value of each UAV $u_i$ is not only related to its own observation state and action $\big(o_i(t), a_i(t)\big)$; it is also related to the observation and action of one-hop neighbor UAVs $u_j \in N_i^1(t)$ represented as $\Big(o_j(t), a_j(t)\Big)$, as shown in Figure 5.3. Thus, in distributed cooperative training, Q-value given by the online-critic network of UAV $u_i$ with parameter $\omega_i$ is represented as $Q_i(o_i(t), o_j(t), a_i(t), a_j(t); \omega_i)$. For simplicity, we consider $Q_i(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i)$, where $\boldsymbol{S_i} = \Big(o_i(t), o_j(t)\Big)$, and $\boldsymbol{A_i} = \Big(a_i(t), a_j(t)\Big)$. Generally, to obtain the optimal action policy in actor network gradient ascent is applied. According to the estimated $Q_i(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i)$, the gradient of $J(\theta_i)$ is obtained with respect to $\theta_i$ as follows:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{\theta_i}[\nabla G_i(t)] = \mathbb{E}_{\theta_i}\left[\nabla_{\theta_i}\mu_{\theta_i}(o_i(t))\nabla_{a_i} Q_i(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i)\big|_{a_i=\mu_{\theta_i}(o_i)}\right] \tag{5.23}$$

Then, $\nabla_{\theta_i} J(\theta_i)$ given by (23) are backpropagated to the online actor network with learning rate$\xi \in [0\ 1]$ to update $\theta_i$ as follows:

$$\theta_i \leftarrow \theta_i + \xi \nabla_{\theta_i} J(\theta_i) \tag{5.24}$$

The online critic network is updated by using the temporal difference error given by the critic loss function as follows:

$$L(\omega_i) = \mathbb{E}_{\omega_i}\left[\Big(y_i^t - Q_i(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i)\Big)^2\right], \tag{5.25}$$

where $y_i^t = r_i(t) + \lambda Q_i'(\boldsymbol{S_i}', \boldsymbol{A_i}'; \omega_i')|_{a_i'=\mu_{\theta_i'}(o_i')}$ represents the target value given by the target critic network. The online critic network is updated by minimizing $L(\omega_i)$ given by (5.25) according to gradient descent with respect to $\omega_i$ as follows:

$$\nabla_{\omega_i} L(\omega_i) = -2\mathbb{E}_{\omega_i}[r_i + \lambda Q_i'(\boldsymbol{S_i}', \boldsymbol{A_i}'; \omega_i') - Q(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i)]\nabla_{\omega_i} Q_i(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i) \tag{5.26}$$

According to $\nabla_{\omega_i} L(\omega_i)$ and critic network learning rate $\varsigma$, $\omega_i$ is updated as follows:

$$\omega_i \leftarrow \omega_i - \varsigma \nabla_{\omega_i} L(\omega_i) \tag{5.27}$$

The target actor and critic network parameters are then updated by slowly tracking the learned online network parameters by updating rate $\tau$ as follows:

$$\begin{cases} \theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i' \\ \omega_i' \leftarrow \tau\omega_i + (1-\tau)\omega_i', \end{cases} \tag{5.28}$$

Finally, to stabilize the training process, a replay buffer $\mathcal{R}_i$ is employed to save the state transition samples and it is utilized to efficiently update the network parameters, as given in Figure 5.3. In each training epoch, we randomly pick a mini batch $M$ containing $\boldsymbol{l}$ samples experience dataset denoted as $(\boldsymbol{S_i^l}, \boldsymbol{A_i^l}, \boldsymbol{r_i^l}, \boldsymbol{S_i^{l'}})$. According to (5.23) and (5.26), $\nabla_{\theta_i} J(\theta_i)$ and $\nabla_{\omega_i} L(\omega_i)$ is approximated as follows:

$$\nabla_{\theta_i} J(\theta_i) \approx \frac{1}{M} \sum_{l=1}^{M} \left[ \nabla_{\theta_i} \mu_{\theta_i}(o_i(t)) \nabla_{a_i} Q_i(\boldsymbol{S_i^l}, \boldsymbol{A_i^l}; \omega_i)\big|_{a_i=\mu_{\theta_i}(o_i)} \right], \tag{5.29}$$

$$\nabla_{\omega_i} L(\omega_i) \approx -\frac{2}{M} \sum_{l=1}^{M} \left[ \left[ r_i^l + \lambda Q_i'\left(\boldsymbol{S_i^{l'}}, \boldsymbol{A_i}^{l'}; \omega_i'\right) - Q_i(\boldsymbol{S_i^l}, \boldsymbol{A_i^l}; \omega_i) \right] \nabla_{\omega_i} Q_i(\boldsymbol{S_i^l}, \boldsymbol{A_i^l}; \omega_i) \right], \tag{5.30}$$

#### 5.3.3.1 LSTM-Based Actor Network

The actor network considers $o_i(t)$ as input, then it forwards its three components $o_i^1(t) = \{\overrightarrow{CR_i}, \overrightarrow{AR_i}, \overrightarrow{SR_i}, \overrightarrow{BS_i}\}$, $o_i^2(t) = \{\phi_{f_{k,i}}, \phi_{f_{k,k}}\}$, and $o_i^3(t) = \{(LD_{ij}, LD_{jk}), \gamma_{ij}, E_j, q_j\}$ to three different LSTM-based SRLs, as illustrated in Figure 5.4.

LSTM utilizes cell memory to store summary of the previous inputs sequence, and gating mechanisms to control the information flow between forget gate, input gate, output gate, and cell memory. Accordingly, LSTM can adaptively learn the long-term dependency relationships between time-series data of UAVSN topology. Due to space limitations, we will not provide detailed explanations of the internal cell structure of LSTM. More details on the structure of LSTM can be found in [172]. The decoupling of observation state $o_i(t)$ through three different LSTM-based SRL forwards better environmental state representation to the actor FCL, which is conducive to achieving a better deterministic policy. If all the observation states are mixed and forwarded as input to one FCL or LSTM-based SRL in actor network, it may hardly distinguish them, which leads to learning an undesirable policy. At each time, the LSTM-based SRL-1 takes the input $o_i^1(t)$ and based on the previous hidden state $o_i^{h,1}(t-1)$, it returns the next hidden state $o_i^{h,1}(t) = LSTM[o_i^1(t), o_i^{h,1}(t-$

1)] as output. Similar procedures are applied to obtain $o_i^{h,2}(t)$, and $o_i^{h,3}(t)$ for $o_i^2(t)$ and $o_i^3(t)$, respectively. Finally, the outputs given by three LSTM-based SRLs are fed into FCL to produce the action $a_i(t) = [\vec{F}_i, \phi_{f_{k,i}}, u_j \in N_i^1]$.

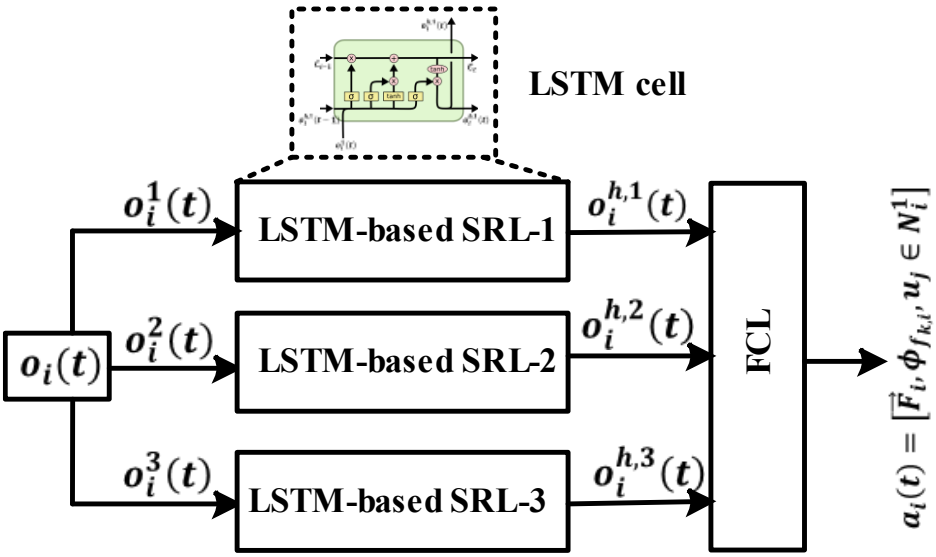



Figure 5.4 Structure of an actor network.

In the offline training process to explore the optimal action under current historical observation, we applied a Gaussian noise $W_n$ with zero mean and limited variance as follows $a_i(t) = [\vec{F}_i + W_n, \phi_{f_{k,i}}, u_j \in N_i^1]$. Combining the Gaussian noise with action $\vec{F}_i$ enhances the adaptability of JTFR to the realistic UAVSN environment, including sensor noise, positional disturbance caused by the wind, and communication delays. Notably, the parameters of actor LSTM-based SRLs and FCL are updated according to the (5.23), (5.24), and (5.29).

### 5.3.3.2 Multi-Head Attentional Critic Network

As discussed in Section 5.3.3, each UAV actor network selects an action according to its current historical observation $o_i(t)$. Its attentional critic network then produces the Q-value to evaluate the actor network performance not only by considering the observation-action of the current UAV but also by selectively paying attention to the neighboring UAVs observation-action as decision making of each UAV is coupled with other UAVs. Notably, in large-scale UAVSNs, it is impractical for each UAV to pay attention to all the remaining UAVs, particularly as some UAVs may stay very far away (i.e., outside communication

range), and their local observation-actions have an extremely low impact on the current UAV. Thus, to reduce computational complexity and increase scalability, we applied distributed cooperative training by only considering one-hop neighbor UAV's $u_j \in N_i^1(t)$ observation-action. The attention mechanism generates normalized attention weight by checking the similarity between query and key vector, which was originally proposed in [173].

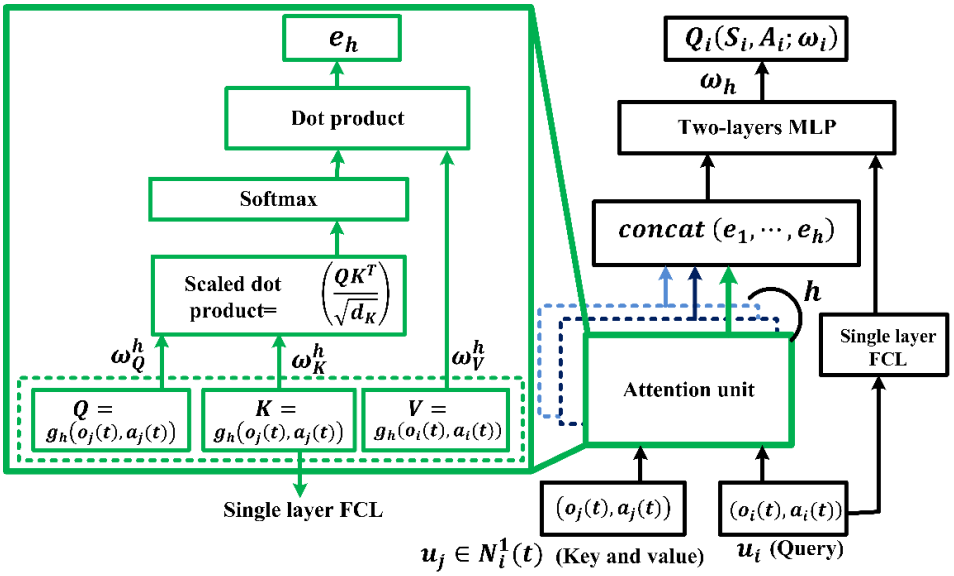



Figure 5.5 Structure of a multi-head attentional critic network.

In our multi-head attentional critic network, the query $Q = g_h(o_i(t), a_i(t))$ contains the features of the observation state-action of a particular UAV $u_i$, where key $K = g_h(o_j(t), a_j(t))$ and $V = g_h(o_j(t), a_j(t))$ are the features of state-action of its one-hop neighbors $u_j \in N_i^1(t)$. Here, $g_h(\bullet)$ represents a single-layer FCL with learnable weights $\omega_Q^h$, $\omega_K^h$, and $\omega_V^h$, as shown in Figure 5.5. Subsequently, based on the scaled dot product similarity between query $Q$ and key $K$, the critic network generates weights to adaptively pay attention to the different one-hop neighbor UAVs. A SoftMax operation was performed to normalize the attention weight before multiplying with the $V$ value to compute the context Q-value $e_h$ for each head $h$ given by (5.31).

Finally, the output of each attention head is concatenated and passed through another two layers of multi-layer perceptron (MLP) to generate a final Q-value $Q_i(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i)$ given by (5.32) to update the critic loss and actor network parameters. We use three attention heads

to focus on the features related to trajectory control, frequency band selection, and relay selection. Thus,

$$e_h = ATT(Q, K, V) = \left[ softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)\right] \times V, \tag{5.31}$$

$$Q_i(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i) = f_i\big(g_i(o_i, a_i), \text{concat}(e_1, \cdots, e_h)\big), \tag{5.32}$$

where $d_K$ represents the dimension of the key $K$, which is used as a scaling factor to prevent gradient disappearance. $f_i(\bullet)$ denotes two-layers of MLP with $\omega_h$ learnable weight and $g_i(\bullet)$ is a single FCL. The two-layers of MLP helps to extract the features and reduce the dimension of the concatenated matrix. Notably, the critic network parameters $\omega_i \cong \{\omega_Q^h, \omega_K^h, \omega_V^h, \omega_h\}$ are updated according to procedures (5.26), (5.27), and (5.30). The above-mentioned training process is systematically outlined in Algorithm 5.1.

### 5.3.4 Computational Complexity

The computational complexity of the proposed DMA-DDPG can be divided into the complexity of the actor network and that of critic network. The complexity of the LSTM-based actor SRL layer is $\mathcal{O}(N_L I_D S_L)$, where $N_L$ denotes the number of LSTM units, $I_D$ indicates the dimension of the input observation, and $S_L$ represents the sequence length remembered by the LSTM. Here, $I_D$ is directly related to the number of UAVs in the swarm. The actor FCL will then have a complexity of $\mathcal{O}(LNIO)$, where $L$, $N$, $I$, and $O$ represent the number of layers, number of neurons per layer, input features, and output features, respectively. For the multi-head attentional critic networks, complexity is $\mathcal{O}(h\mathcal{U}^2 I_A)$, where $h$ denotes the number of heads (we set $h = 3$), $\mathcal{U}^2$ is for performing the dot product between query, key, and value of $\mathcal{U}$ UAVs, and $I_A$ denotes the dimension of the observation-action spaces of each UAV. Since the actor network only has the observation by considering one-hop and two-hop neighboring UAVs, and the attentional critic network pays attention to only one-hop neighboring UAVs, with the increasing number of UAVs, the observation-action dimensionality should remain less compared to the fully centralized MA-DDPG. Accordingly, DMA-DDPG provides higher scalability and lower computational complexity compared to the fully centralized MA-DDPG.

---

**Algorithm 5.1:** DMA-DDPG-based JTFR algorithm

---

**Input:** UAV number $\mathcal{U}$; frequency band $f$; and BS location $\vec{p}_{BS}$.
**Output:** Optimal mobility, frequency band, and relay UAV selection
**// Initialization**
1: Initialize each agent online actor and critic, and target actor and critic with parameters $\theta_i$, $\omega_i$, $\theta_i'$, and $\omega_i'$, respectively;
2: Initialize each agent replay buffer $\mathcal{R}_i$;
3: **for** each episode $= 0:$ max_episode **do**
4: Randomly initialize the position and velocity of each UAV;
5: **for** each timeslot $\mathcal{T} = 0: t$ **do**
6: **for** each UAV $u_i \in \mathcal{U}$ **do**
7: Obtain the motion rules using (5.8)−(5.11) and initial $o_i(t)$;
8: Decouple $o_i(t)$ into $o_i^1(t)$,$o_i^2(t)$, and $o_i^3(t)$;
9: Input $o_i^1(t)$,$o_i^2(t)$, and $o_i^3(t)$ to actor LSTM-based SRLs to obtain output $o_i^{h,1}(t)$, $o_i^{h,2}(t)$, and $o_i^{h,3}(t)$, respectively;
10: Forward $o_i^{h,1}(t)$, $o_i^{h,2}(t)$, and $o_i^{h,3}(t)$ to actor FCL to obtain output action $a_i(t) = \left[\vec{F}_i, \phi_{f_{K,i}}, u_j \in N_i^1\right]$;
11: Execute action $a_i(t) = \left[\vec{F}_i + W_n, \phi_{f_{k,i}}, u_j \in N_i^1\right]$;
12: Update $\vec{a}_i$, $\vec{v}_i$, and $\vec{p}_i$ using motion model (5.12)−(5.14);
13: Update $LD_{ij}$ using (5.15)−(5.16) and adjust $HI_i$ using (5.17);
14: Update SINR $\gamma_{ij}$ using (5.2);
15: Update queue backlog size using (5.3) and delay using (5.4);
16: Update residual energy (RE) level $E_i$ using (5.5);
17: Get reward $r_i(t)$ by using (5.18)−(5.22) and obtain $o_i'$;
18: Obtain $(o_j, a_j, o_j')$ from $u_j \in N_i^1$ and construct $(\boldsymbol{S_i}, \boldsymbol{A_i}, r_i, \boldsymbol{S_i'})$;
19: Store state transition data $(\boldsymbol{S_i}, \boldsymbol{A_i}, r_i, \boldsymbol{S_i'})$ in replay buffer $\mathcal{R}_i$;
20: Overwrites oldest transition data if replay buffer $\mathcal{R}_i$ is full;
21: Select a random mini batch $M$ with $\boldsymbol{l}$ samples $(\boldsymbol{S_i^l}, \boldsymbol{A_i^l}, \boldsymbol{r_i^l}, \boldsymbol{S_i^{l'}})$;
22: Compute $Q_i(\boldsymbol{S_i}, \boldsymbol{A_i}; \omega_i)$ according to (5.31)−(5.32);
23: Update online critic network using (5.27), and (5.30);
24: Update online actor network using (5.24), and (5.29);
25: Update both target actor and critic network using (5.28);
26: **end for**
27: **end for**
28: **end for**

---

## 5.4 Performance Evaluation

In this section, we present the details of extensive computer simulations that were performed to evaluate the performance of the proposed JTFR. Subsequently, the results are compared with that of existing schemes. We used MATLAB R2022a to develop the UAVSN environment. Notably, JTFR is a DMA-DDPG-based algorithm, where the actor network

has observation from a two-hop neighbor and the critic network pays attention to the one-hop neighbor. For comparison, the following existing algorithms are considered:

- We consider JTFR variation DMA-DDPG-1, in which both the actor and critic network has only one-hop neighbor information. DMA-DDPG-1 has a similar MDP formulation and neural network architecture as discussed in Section 5.3.2.

- We consider MA-DDPG-LSTM [135], in which both actor, critic, and their target network are developed by only the LSTM cell. MDP is then formulated using the one-hop neighbor information according to the procedure given in [135]. Here, the state space comprises one hop neighbor list and SINR, action is neighbor link selection, and the reward consists of one-hop LD, SINR, and queue buffer length.

- Finally, we consider the MCA-OLSR [108], which is a recently published novel topology-based proactive cross-layer routing protocol, to validate the effectiveness of the adaptive learning-based algorithm in packet routing. MCA-OLSR is used according to the test environment to obtain the optimal multi-hop routing path between a remote UAV and BS using a table-driven method. Notably, MA-DDPG-LSTM [135] and MCA-OLSR [108] utilize the Gaussian Markov and smooth turn mobility models in their simulation, respectively. To compare in a fair environment and obtain more realistic simulation results for UAVSN, we consider the behavior-based mobility model proposed in [120].

In JTFR, each LSTM-based SRL in actor network contains 64 LSTM units. Then, the actor FCL has one input layer, two hidden layers with 256 and 128 neurons, and one output layer with 5 neurons. In the hidden layer, the rectified linear unit is used as activation function to avoid the vanishing gradient problem during training. In the output layers of actor FCL, we used $tanh$ activation function to predict $\vec{F}_i$, and the SoftMax activation function to select the frequency band and relay UAV. The value for the constant penalty term $r_{lm}$, $r_{pl}$, and $r_{mo}$ in (5.22) are set to 2, 4, and 5, respectively. Additionally, the summary of hyper-parameter values in off-line training of JTFR are listed in Table 5.1.

Table 5.1 Hyper-Parameters in DMA-DDPG of JTFR.

| Parameter | Value |
|---|---|
| Discount factor ($\lambda$) | 0.95 |
| max_episode | 2000 |
| Maximum timeslot per episode ($\mathcal{T}$) | 1000 |
| Replay buffer memory size ($\mathcal{R}$) | 50000 |
| Mini batch size ($M$) | 512 |
| Target network soft update rate ($\tau$) | 0.05 |
| Online actor learning rate ($\xi$) | 0.0001 |
| Online critic learning rate ($\varsigma$) | 0.0002 |
| Optimizer | ADAM |

Initially, UAVs were randomly positioned within a 3D mission area with dimensions $2500 \times 2500 \times [100 - 400]\ m^3$. For each UAV, the value of $A_r$ and $R_r$ were set to $300\ m$ and $50\ m$, respectively. The entire surveillance duration is $\mathcal{T} = 1000$ s, and $\delta_t = 2$ s. The threshold value to calculate the LD was set to 2 s. Initially, $HI_i$ was set to 0.5 s and later adaptively adjusted according to (5.17). Additionally, the values of $v_{\max}$ and $a_{\max}$ for each UAV are set to 20 m/s and 5 m/$s^2$, respectively. For producing data traffic, we assumed a constant bitrate (CBR)-based video streaming application operating on each UAV. At each timeslot, each UAV periodically sends the data packet toward BS. Other important simulation parameters to set the UAVSN environment are listed in Table 5.2.

Table 5.2 Environment Parameters of UAVSNs (JTFR).

| Parameter | Value |
|---|---|
| Dimension of 3D mission area | $2500 \times 2500 \times [100 - 400]\ m^3$ |
| Number of UAVs ($\mathcal{U}$) | $(30 - 100)$ |
| Channel bandwidth | 20 MHz |
| Bandwidth per sub-carrier ($B$) | 1 MHz |
| UAV maximum energy ($E_{\max}$) | $2 \times 10^5$ Joules |
| Path loss exponent ($\alpha$) | 3 |
| SINR threshold ($\gamma_{th}$) | 2 dB |
| CBR rate | 2 Mbps |
| Transport protocol | User datagram protocol |
| Packet arrival model | Poisson |
| Maximum queue buffer size ($q_{\max}$) | 1000 |

### 5.4.1 Performance Metrics

The performance metrics to verify algorithm convergence are as follows:

- Average reward versus number of episodes: It visualizes the learning process of the UAVs and the algorithm convergence over time. As the number of episodes increase, the average reward given by (5.22) should increase as the UAVs interact with the UAVSN environment and gradually learn to improve their action to obtain an optimal policy.

    - Traveling distance fairness (TDF): The TDF for each UAV justifies the motion fairness between UAVs. It is calculated as $\frac{\left(\sum_{i=1}^{\mathcal{U}} D_i\right)^2}{\mathcal{U}\times\sum_{i=1}^{\mathcal{U}}(D_i)^2}$, where $D_i$ represents the distance traveled by each UAV over the collaborative mission. Notably, $D_i$ is calculated by using (5.14). A TDF value close to one implies that the travel distance of each UAV is similar. Since the propulsion energy consumption of a UAV is proportional to the travel distance, a balance in the travel distance ensures equal energy consumption for each UAV within the swarm.

The performance metrics to evaluate the routing protocol performance are as follows:

- Packet delivery ratio (PDR): PDR indicates the ratio between successfully transmitted data packets at the BS and the total number of data packets generated by all UAVs.

- Average end-to-end delay (AE2ED): AE2ED refers to the average time required to successfully transmit data packets to the BS from a particular UAV given by (5.4).

- Normalized control overhead (NCO): NCO corresponds to the ratio between the total size of hello packets required by UAVSN and the total traffic load in the UAVSN transmitted throughout the simulation.

- Normalized residual energy (NRE): NRE for each UAV is computed using $E_i$ given by (5.5) and normalized as follows: $\frac{E_i}{E_{\max}}$. The NRE is examined once the simulation is complete, and a lower NRE specifies the higher energy consumption of UAVs. The above performance metrics are utilized to experiment with two different categories: scalability test and velocity increment test.

### 5.4.2 Simulations Results and Discussion

In this section, to show the effectiveness of our proposed JTFR, we will perform a rigorous comparative analysis according to the above-mentioned performance metrics.

#### 5.4.2.1 Convergence Analysis

Figure 5.6 demonstrates the average reward versus the number of episodes during training of 100 UAVs. Both JTFR and its variation DMA-DDPG-1 obtain better average reward and stable learning curves compared to the MA-DDPG-LSTM. It can be attributed to the fact that each UAV's multi-head attentional critic network pays attention to one-hop neighbor UAVs based on the normalized attention weight and adaptively adjusts its policy according to the neighboring UAVs policy changes, which helps to overcome the environmental non-stationarity. Through the multi-head attention in the critic network, each UAV can learn to obtain only the important features from the neighboring UAVs related to trajectory control, frequency band, and relay UAV selection, which is conducive to making better collaborative decisions. Moreover, owing to the parallelization in feature extraction and adaptive attention weight assignment to neighbor UAVs according to their degree of influence, UAVs can precisely estimate the value-function in muti-agent collaborative UAVSN. Accordingly, it enables UAVs to obtain better action policies and accelerate the convergence for making the optimal decision.

In particular, JTFR obtains the highest average reward and reaches convergence state after approximately 220 episodes because of three crucial reasons. First, owing to the benefit of expanded knowledge about dynamic topology (i.e. up to two-hop neighbor), JTFR obtains a better observation state and can avoid local optima. Second, in JTFR, each observation state that is related to controlling the trajectory, selecting the frequency band, and relay UAV are embedded to the three different LSTM-based SRLs in actor network to mine the temporal relationship in time-series data. Consequently, LSTM-based actor SRLs output features forward better observation state to the actor FCL, which assists to obtain an optimal action policy. Finally, due to the relay UAV selection considering maximum-minimum 3D LD up to two-hop neighbors help UAVs to avoid unexpected link breakages. Since DMA-DDPG-1 utilizes only one-hop neighbor information, it provides less average reward compared to the JTFR. In contrast, MA-DDPG-LSTM did not consider the trajectory control, frequency band allocation, and two crucial constraints (5.7B) and (5.7I) in the action selection process when dealing with UAVSN dynamic environment; thus, it obtains a smaller reward. Additionally, MA-DDPG-LSTM's critic network is solely based on LSTM units. As a result, its learning encounters more oscillations in dynamic multi-agent scenarios. However, MA-DDPG-LSTM average reward is slowly increasing with the number of episodes as LSTM units in both actor and critic are gradually updating their parameters by using policy gradient and minimizing a critic loss function.

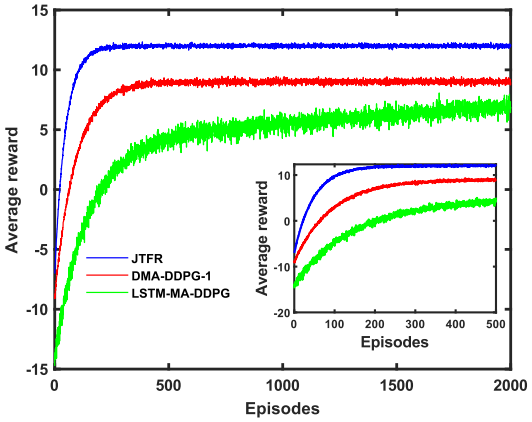



Figure 5.6 Average reward versus the number of episodes.

Figure 5.7 illustrates TDF for different number of UAVs. Since MA-DDPG-LSTM [135] did not consider the trajectory control, we have excluded it from the TDF comparison. Instead, a behavior-based adaptive flocking control algorithm (AFCA) [120] is considered, in which, the control input is generated by performing simple vector addition of motion rules without using any LSTM/DRL method to predict the mobility of UAVs. Moreover, in AFCA, the weight of each motion rule is adaptively adjusted by computing the changes in inter-UAV distances within the current and previous timeslot. In Figure 5.7, with the increasing number of UAVs JTFR provides a better TDF value (close to one), which indicates better motion fairness and swarm cohesion in the collaborative motion task.

Both JTFR and its variation DMA-DDPG-1 provide better TDF compared to the AFCA because of two key reasons. First, in JTFR, the motion rules for each UAV at a particular timeslot are treated as the observation state and forwarded to the LSTM-based actor SRL-1. The LSTM-based actor SRL-1 utilized the most recent historical state of relative distance and relative velocity to precisely predict each motion rules, which is conducive to obtaining a smoother trajectory for each UAV. Additionally, the LSTM-based actor SRL-1 and actor FCL adaptively adjust their weights according to the learning process to produce the optimal control input satisfying the constraints (5.7A)−(5.7D). Second, attention weights in the attentional critic network help to improve trajectory control policy in actor network of each UAV through decision-making according to neighboring UAV's state-action similarity. Since the mobility of each UAV at next timeslot in AFCA is estimated only based on the current timeslot mobility, AFCA provides less accuracy in mobility prediction and less TDF value compared to JTFR and DMA-DDPG-1, even though AFCA generates the motion rules using two-hop information.

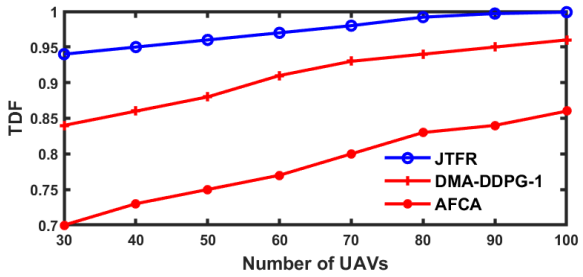



Figure 5.7 TDF versus the number of UAVs.

### 5.4.2.2 Routing Protocol Performance Analysis

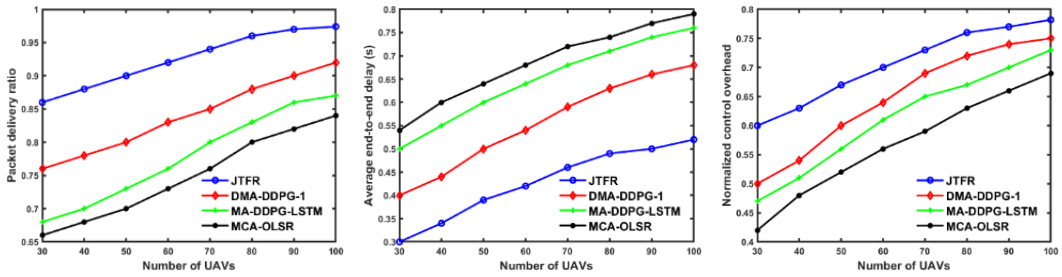
In this section, the network performance of JTFR is discussed by comparing it with that of existing routing protocols for scalability test and velocity increment test.

#### 5.4.2.2.1 Scalability Test

Figure 5.8 demonstrates the network performance (PDR, AE2ED, and NCO) for different number of UAVs. According to Figure 5.8(a), JTFR exhibits better PDR performance compared to other baseline routing protocols due to two major reasons. First, owing to the trajectory control according to the physical layer transmission range UAV swarm maintains optimal node density and connectivity with BS. Here, both trajectory control and optimal frequency band allocation are conducive to achieving higher SINR. Since JTFR produces the motion rules and frequency band using two-hop neighbor information, it makes better decisions compared to its variation DMA-DDPG-1. Additionally, the attentional critic network improves the decision-making process to generate trajectory control input and select frequency band in actor network by paying adaptive attention to the one-hop neighbor of each UAV. In contrast, MA-DDPG-LSTM and MCA-OLSR exhibit less PDR performance compared to JTFR and DMA-DDPG-1, primarily because they did not consider trajectory control and frequency resource allocation in the physical and MAC layer. Second, unlike other routing protocols, JTFR selects the relay UAV according to the maximum-minimum 3D LD while satisfying the constraint (5.7B), which is conducive to obtaining a more stable path and a smaller number of retransmissions to relay data packets toward BS.

According to Figure 5.8(b), JTFR provides less AE2ED compared to others because of three vital reasons. First, JTFR selects the relay UAV that has a small queue backlog length while satisfying constraint (5.7H) along with high SINR, which helps to reduce the queuing delay. Moreover, JTFR only explores the relay UAV in the direction given by $\Delta_{ij} > 0$ to avoid excessive detours of data traffic while ensuring a smaller number of hops in the end-to-end path. Second, the attentional critic network pays attention to other UAVs relay selections, which helps each UAV actor-network to avoid similar actions in relay selection.

It is because if most of the UAVs choose the same relay it may congest the network, creating higher queuing delay. Third, owing to the advantage of trajectory control using two-hop knowledge according to the imposed communication range constraint (5.7A), JTFR maintains an optimal aerial node density during the entire collective motion task. In association with the optimal node density and frequency band allocation, each UAV achieves a higher SINR and less contention during simultaneous transmission to relay data packets toward BS. Such features are not considered by both MA-DDPG-LSTM and MCA-OLSR, thus, they encounter higher AE2ED compared to JTFR and its variation DMA-DDPG-1. Moreover, MCA-OLSR utilizes carrier sense multiple access with collision avoidance, which encounters more contentions and retransmissions even though they employed queue management in the MAC layer.



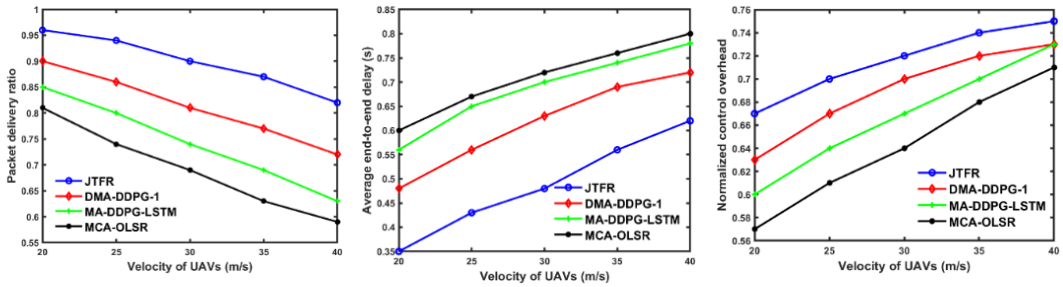

(a) PDR (b) AE2ED (c) NCO

Figure 5.8 Network performance in scalability test.

Figure 5.8(c) illustrates the NCO performance for different number of UAVs. JTFR and its variation DMA-DDPG-1 exhibits higher NCO compared to others. In particular, JTFR utilizes two-hop neighbor information related to the mobility information and frequency block state, thus, it requires a little higher control overhead compared to DMA-DDPG-1. Additionally, during the training process, both JTFR and DMA-DDPG-1 require obtaining the observation-action from the one-hop neighbor UAVs, thus, they encounter control overhead compared to MA-DDPG-LSTM and MCA-OLSR. However, both MA-DDPG-LSTM and MCA-OLSR broadcast hello packets in the fixed hello interval without sensing the mobility changes in their local state. As a result, they have less adaptivity to time-varying dynamic topology. In contrast, JTFR adaptively tweaks the hello interval given by (5.17), to address the trade-off between topology prediction accuracy and control overhead. Consequently, the NCO for both MA-DDPG-LSTM and MCA-OLSR increases almost linearly with increasing the number of UAVs. In contrast, JTFR and its variation DMA-DDPG-1, NCO exhibits a lesser increment in the slope. Therefore, it can be stated that JTFR has better adaptivity and scalability performance compared to others with a reasonable cost in control overhead for large-scale UAVSN.

#### 5.4.2.2.2 Velocity Increment Test

Figure 5.9 illustrates the network performance for the different maximum achievable velocities for 100 UAVs. Figure 5.9(a) and 5.9(b) show that JTFR offers significantly higher PDR and less AE2ED compared to others owing to three vital reasons. First, in JTFR, to control the trajectory, each UAV computes its motion rules utilizing two-hop neighbor mobility information, and each motion rules are fed into the LSTM-based actor SRL-1 as state observation at each timeslot. The LSTM-based actor SRL-1 utilizes previous historical information of relative distance and relative velocity with neighboring UAVs to represent a better state of dynamic UAVSNs to the actor FCL to predict the control input for each UAV for updating the acceleration, velocity, and position. Moreover, the attentional critic network generates a more precise Q-value to adaptively update the learning parameters in LSTM-based actor SRLs and actor FCL according to network condition defined by the link utility maximization problem (5.7) and its constraints (5.7A) – (5.7I). Second, in the reward function given by (5.18), JTFR gives more reward for selecting the relay UAV that has stable mobility intimacy with neighboring UAVs defined by the maximum-minimum 3D LD up to two-hop neighbors. In highly dynamic topology, link stability is highly coupled with residual LD, and JTFR obtains a better stable path in high mobility UAVSNs. Third, in JTFR, since each UAV selects a frequency band by paying attention to the neighboring UAVs participating in the simultaneous transmissions, each UAV can significantly minimize the mutual interference, which helps to achieve higher SINR and data rate for forwarding packets toward BS.




(a) PDR (b) AE2ED (c) NCO

Figure 5.9 Network performance in velocity increment test.

Furthermore, the relay UAV selection considering less queuing backlog size with imposed constraint (5.7H) and packet travel time constraint (5.7B) helps to exhibit less network congestion, delay, and avoid unexpected link breakages in during data transmission. Thus, it helps to reduce the unnecessary retransmissions of data packets in UAVSNs. Consequently, less retransmission of data packets reduces the traffic overload and delay in highly dynamic UAVSN. Both MA-DDPG-LSTM and MCA-OLSR do not support the

above-mentioned features, thus, they exhibit less PDR and higher AE2ED compared to JTFR and its variation DMA-DDPG-1.

According to Figure 5.9(c), JTFR and its variation DMA-DDPG-1 exhibit higher NCO compared to other baseline protocols. It is because with changing increasing velocity in UANSN, each UAV encounters a higher degree of changes in mobility, thus, residual LD changes. Consequently, in JTFR, the hello interval frequency given by (5.17) becomes smaller and triggers higher control overhead compared to others to achieve the dynamic topology. Additionally, JTFR requires collecting mobility information from one-hop and two-hop neighbors, thus, it encounters higher NCO compared to others. Since, both MA-DDPG-LSTM and MCA-OLSR use a fixed hello interval, they have less sensitivity to dynamic topology changes and less NCO compared to JTFR and its variation DMA-DDPG-1. In particular, owing to the advantages of multi-point relay selection in MCA-OLSR reduces redundancy in hello packet broadcasting, MCA-OLSR exhibits less NCO than others. However, we observe for both MA-DDPG-LSTM and MCA-OLSR, the NCO increases almost linearly with increasing the maximum attainable velocity within the swarm. In contrast, JTFR and its variation DMA-DDPG-1 have less degree of increment in control overhead with increasing velocity, owing to its adaptive learning in both LSTM-based actor and attention network-based critic network of each UAV. Therefore, considering this reasonable cost in NCO, it can be stated that JTFR offers a significant improvement in network performance in high-mobility UANSN.

Figure 5.10 presents the NRE for different routing protocols for 100 UAVs. In Figure 5.10, the horizontal red line within each NRE distribution box for each routing protocol represents the median of NRE. According to Figure 5.10, the proposed JTFR provides better NRE status (less energy consumption) compared to other routing protocols owing to two vital reasons. First, in JTFR, we notice better TDF meaning that each UAV travels almost a similar distance to execute the collective motion task and achieve swarm cohesion while obeying the behavior-based motion rules. Since propulsion energy consumption is directly proportional to the flying distance, and it is significantly higher than communication energy consumption, balancing the flying distance between UAVs is equivalent to obtaining equal and minimal UAV propulsion energy consumption. Second, in the link utility function, JTFR jointly considers the relay UAV RE and mobility prediction metric LD, which facilitates obtaining stable end-to-end paths with fewer retransmissions. Thus, it significantly reduces the packet transmission energy consumption given by (5.5). JTFR provides better NRE status and balance in energy consumption as the width of NRE distribution box of JTFR is small compared to others. Additionally, owing to the consideration of relay UAV RE level in the link utility or reward function given by (5.21), JTFR successfully avoids the energy holes and obtains a more stable link in the end-to-end path. Since both MA-DDPG-LSTM and MCA-OLSR do not consider the trajectory control and UAV RE level in the routing

metric, they produce less NRE status (higher energy consumption) compared to the proposed JTFR and its variation DMA-DDPG-1.

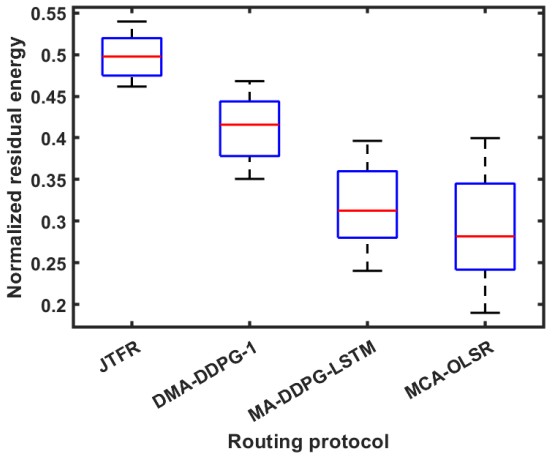



Figure 5.10 Normalized residual energy.

Notably, the computational energy cost to train the JTFR model is not considered, because the entire training process will be performed in offline mode. Following the training, the trained model will be uploaded to each UAV to execute the mission online. During the training process, we introduced the Gaussian noise with a control input to achieve adaptivity to the dynamic UAVSNs environment in real-life scenarios. Moreover, during online execution, the trained model of each UAV collects the observation from one-hop and two-hop neighbors using hello packets and constructs its MDP tuple to make optimal real-time decisions. Subsequently, JTFR can also utilize the attentional critic network to improve its policy since it can act only use one-hop neighbor information. Additionally, joint consideration of controlling trajectory using a realistic-behavior-based motion model, selecting frequency band, and relay UAV using multi-objective link quality metrics (3D maximum-minimum LD, SINR, delay including constraint queue backlog size, and RE), facilitates to obtain more realistic results in the simulation environment of UAVSNs and high fidelity for behaving optimally during online execution.

#### 5.4.2.2.3 Summary on Performance Improvement

In this subsection, according to the performance analysis in Section 5.4.2, a comparative summary on the performance improvement over the baseline routing protocols is presented. In the scalability test, JTFR exhibits 19.36%, and 10.03% better average TDF compared to AFCA and DMA-DDPG-1, respectively. JTFR then gives 15.20%, 25.03%, and 32.42% better PDR averages compared to DMA-DDPG-1, MA-DDPG-LSTM, and MCA-OLSR, respectively. JTFR provides 30.82%, 51.46%, and 60.23% less AE2ED

compared to DMA-DDPG-1, MA-DDPG-LSTM, and MCA-OLSR, respectively. Nevertheless, JTFR exhibits 8.18%, 13.15%, and 19.35% higher average NCO compared to DMA-DDPG-1, MA-DDPG-LSTM, and MCA-OLSR, respectively.

In the velocity increment test, JTFR exhibits 14.51%, 22.37%, and 24.02% better average PDR compared to DMA-DDPG-1, MA-DDPG-LSTM, and MCA-OLSR, respectively. Moreover, JTFR provides 30.04%, 51.46%, and 57.25% better AE2ED compared to DMA-DDPG-1, MA-DDPG-LSTM, and MCA-OLSR, respectively. However, JTFR exhibits 5.63%, 6.80%, and 10.37% higher average NCO compared to DMA-DDPG-1, MA-DDPG-LSTM, and MCA-OLSR, respectively. Additionally, JTFR exhibits 20%, 36%, and 46% less average energy consumption (propulsion energy and transmission energy) compared to DMA-DDPG-1, MA-DDPG-LSTM, and MCA-OLSR, respectively. Owing to the remarkable performance enhancement in PDR, AE2ED, and energy consumption, such reasonable cost in control overhead is acceptable.

## 5.5 Conclusion

In this study, we formulated a link utility maximization problem by jointly considering the 3D LD, link SINR, delay, and UAV RE level under several practical constraints to route data packets from UAVSNs to BS. To solve this problem, the adaptive DMA-DDPG-based JTFR algorithm coupled with swarming behavior is proposed, in which each UAV actor network obtains the adaptivity with dynamic time-varying topology by using its LSTM-based SRLs. Subsequently, critic networks obtain the precise Q-value function to train each UAV actor policy and minimize the critic loss by adaptively paying attention to the neighboring UAVs, according to the collaborative decision-making of trajectory control, frequency band allocation, and relay UAV selection. Joint consideration of trajectory control and frequency band selection maximizes both link SINR and link stability in UAVSNs as they are highly coupled. Additionally, owing to the consideration of 3D maximum-minimum LD, queue backlog size, and RE level of relaying UAV in JTFR is conducive to achieving significant improvements in packet routing in terms of PDR, AE2ED, and energy consumption.

# 6. Conclusions and Future Works

## 6.1 Conclusions

Owing to high mobility, limited transmission range, communication uncertainties (i.e., delay and mutual interference), and wind disturbance, maintaining both mission and communication performance of UAVSN is very challenging. To alleviate the effects of dynamic topology in UAVSN this thesis jointly investigated the relation between collaborative mobility control, and RL/DRL-based packet routing based on multiple link quality metrics.

In the first work, we proposed joint topology control and routing to efficiently execute crowd surveillance utilizing UAVSN. The two-phase topology control of UAVSN meets the trade-off between coverage to the ground terminal and aerial connectivity. Thus, it provides better tracking coverage ratio in terms of mission performance and better PDR, fewer retransmissions, and less end-to-end delay in terms of communication performance. Additionally, adaptive exploration-exploitation strategy in inter-cluster routing along with multi-objective reward function helps to avoid unexpected link breakages, routing loops, and network congestion.

In the second work, the proposed QRIFC jointly investigated the relations between mobility control, delay, and routing policy using two-hop neighbor information. The proposed adaptive flocking model based on swarming behavior defines the optimal mobility for each UAV to maintain optimal node density, traveling distance fairness, connectivity, and coverage. Moreover, the mobility alignment according to the relative distance and velocity with neighboring UAVs in the two-hop neighborhood gives faster swarm cohesion and stable LD, while incurring optimal control overhead. The adaptive exploration-exploitation strategy, topology triggering, and new multi-objective reward function in QRIFC based on two-hop 3D maximum-minimum LD, link PTS, and relaying UAVs RE provides high PDR, shorter end-to-end delay, less retransmissions, and more balance in energy consumption. Additionally, it helps to avoid local optima and gives better average reward compared to the existing baseline routing protocols.

Finally, the proposed JTFR jointly considers collaborative trajectory control, frequency resource allocation, and relay UAV selection to route data packets from UAVSNs to BS by maximizing a link utility function considering the cross-layer design. The link utility comprises link stability defined by 3D maximum-minimum LD, link SINR, queuing delay, and RE of relaying UAV under the constraint of minimum separating distance, communication range, flight constraint of UAVs, threshold SINR, queue buffer size, and energy level of UAVs. To efficiently solve and deal with large state-action space in this joint optimization problem, we utilized adaptive DMA-DDPG-based algorithm coupled with swarming behavior. In DMA-DDPG, to deal with the dynamic topology and avoid local

optima, LSTM-based actor policy network is designed by leveraging the historical observation from two-hop neighborhood. Then, a multi-head attentional critic network is utilized to achieve better learning stability along with faster convergence in multi-agent interaction by adaptively paying attention to nearby UAVs according to their degree of influence. Joint consideration of controlling trajectory, selecting frequency band, and relay UAV according to multi-metric link utility function while satisfying imposed constraint helps to achieve better link utility. Additionally, it significantly improves packet routing performance along with less UAV energy consumption compared to the baseline routing protocols.

## 6.2 Future Works

In the future work, we will consider heterogenous UAVSN flocking control and priority-based packet routing in LAP and HAP environment to conduct collaborative missions in an emergency. Another interesting research idea is designing flocking-based neighbor discovery for millimeter (mm)-Wave assisted UAVSN, in which each UAV utilizes separate low-frequency omnidirectional control channel to discover neighbor, control the relative mobility, and MAC layer transmission scheduling while avoiding deafness, beam-alignment error, and hidden terminal problem in directional communication [174]. Then, high-frequency mm-Wave data channel is utilized for only directional data transmission in U2U link to achieve higher data rate, low latency, and spatial multiplexing. We will also consider joint collaborative trajectory control and resource allocation in mobile edge server enabled UAVSN using multi-agent deep reinforcement learning to provide computing services to the low power IoT devices in remote areas to minimize task execution delay and energy consumption of IoT devices.

# Bibliography

[1] H. Shakhatreh *et al.*, "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019, doi: 10.1109/ACCESS.2019.2909530.

[2] M. Mozaffari, W. Saad, M. Bennis, Y. H. Nam, and M. Debbah, "A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019, doi: 10.1109/COMST.2019.2902862.

[3] Y. Tang *et al.*, "Vision-Aided Multi-UAV Autonomous Flocking in GPS-Denied Environment," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 616–626, 2019, doi: 10.1109/TIE.2018.2824766.

[4] A. Feriani and E. Hossain, "Single and Multi-Agent Deep Reinforcement Learning for AI-Enabled Wireless Networks: A Tutorial," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 2, pp. 1226–1252, 2021, doi: 10.1109/COMST.2021.3063822.

[5] P. S. Bithas, E. T. Michailidis, N. Nomikos, D. Vouyioukas, and A. G. Kanatas, "A survey on machine-learning techniques for UAV-based communications," *Sensors (Switzerland)*, vol. 19, no. 23, pp. 1–39, 2019, doi: 10.3390/s19235170.

[6] J. Tang, S. Lao, and Y. Wan, "Systematic Review of Collision Avoidance Approaches for Unmanned Aerial Vehicles," *IEEE Syst. J.*, pp. 1–12, 2021.

[7] J. Lansky, A. M. Rahmani, and M. Hosseinzadeh, "Reinforcement Learning-Based Routing Protocols in Vehicular Ad Hoc Networks for Intelligent Transport System (ITS): A Survey," *Mathematics*, vol. 10, no. 24, p. 4673, Dec. 2022, doi: 10.3390/math10244673.

[8] A. Rovira-Sugranes, A. Razi, F. Afghah, and J. Chakareski, "A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook," *Ad Hoc Networks*, vol. 130, no. 2008784, p. 102790, May 2022, doi: 10.1016/j.adhoc.2022.102790.

[9] D. Shumeye Lakew, U. Sa'ad, N.-N. Dao, W. Na, and S. Cho, "Routing in Flying Ad Hoc Networks: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1071–1120, 2020, doi: 10.1109/COMST.2020.2982452.

[10] A. Bujari, C. E. Palazzi, and D. Ronzani, "A Comparison of Stateless Position-based Packet Routing Algorithms for FANETs," *IEEE Trans. Mob. Comput.*, vol. 17, no. 11, pp. 2468–2482, Nov. 2018, doi: 10.1109/TMC.2018.2811490.

[11] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, "Routing in Flying Ad Hoc Networks: Survey, Constraints, and Future Challenge Perspectives," *IEEE Access*, vol. 7, pp. 81057–81105, 2019, doi: 10.1109/ACCESS.2019.2923840.

[12] M. Y. Arafat and S. Moh, "A Survey on Cluster-Based Routing Protocols for

Unmanned Aerial Vehicle Networks," *IEEE Access*, vol. 7, pp. 498–516, 2019, doi: 10.1109/ACCESS.2018.2885539.

[13] M. Y. Arafat and S. Moh, "Routing protocols for unmanned aerial vehicle networks: A survey," *IEEE Access*, vol. 7, pp. 99694–99720, 2019, doi: 10.1109/ACCESS.2019.2930813.

[14] L. Cao, Y. Cai, and Y. Yue, "Swarm Intelligence-Based Performance Optimization for Mobile Wireless Sensor Networks: Survey, Challenges, and Future Directions," *IEEE Access*, vol. 7, pp. 161524–161553, 2019, doi: 10.1109/ACCESS.2019.2951370.

[15] G. Chen, C. Cheng, X. Xu, and Y. Zeng, "Minimizing the Age of Information for Data Collection by Cellular-Connected UAV," *IEEE Trans. Veh. Technol.*, pp. 1–5, 2023, doi: 10.1109/TVT.2023.3249747.

[16] S. H. Alsamhi, O. Ma, M. S. Ansari, and F. A. Almalki, "Survey on collaborative smart drones and internet of things for improving smartness of smart cities," *IEEE Access*, vol. 7, pp. 128125–128152, 2019, doi: 10.1109/ACCESS.2019.2934998.

[17] T. Shafique, H. Tabassum, and E. Hossain, "End-to-end energy-efficiency and reliability of UAV-assisted wireless data ferrying," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1822–1837, 2020, doi: 10.1109/TCOMM.2019.2958805.

[18] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wirel. Commun.*, vol. 27, no. 1, pp. 140–146, 2020, doi: 10.1109/MWC.001.1800594.

[19] R. A. Nazib and S. Moh, "Routing protocols for unmanned aerial vehicle-aided vehicular Ad Hoc Networks: A survey," *IEEE Access*, vol. 8, pp. 77535–77560, 2020, doi: 10.1109/ACCESS.2020.2989790.

[20] Z. Yang, W. Xu, and M. Shikh-Bahaei, "Energy Efficient UAV Communication with Energy Harvesting," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1913–1927, 2020, doi: 10.1109/TVT.2019.2961993.

[21] L. Xie, X. Cao, J. Xu, and R. Zhang, "UAV-Enabled Wireless Power Transfer: A Tutorial Overview," *IEEE Trans. Green Commun. Netw.*, vol. 2400, no. c, pp. 1–23, 2021, doi: 10.1109/TGCN.2021.3093718.

[22] M. Kishk, A. Bader, and M. S. Alouini, "Aerial Base Station Deployment in 6G Cellular Networks Using Tethered Drones: The Mobility and Endurance Tradeoff," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 103–111, 2020, doi: 10.1109/MVT.2020.3017885.

[23] H. Wang, H. Zhao, W. Wu, J. Xiong, D. Ma, and J. Wei, "Deployment Algorithms of Flying Base Stations: 5G and Beyond With UAVs," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10009–10027, 2019, doi: 10.1109/JIOT.2019.2935105.

[24] A. Trotta, U. Muncuk, M. Di Felice, and K. R. Chowdhury, "Tracking Using Unmanned Aerial," *IEEE Veh. Technol. Mag.*, vol. 15, no. April, pp. 96–103, 2020.

[25] Z. Mou, Y. Zhang, F. Gao, H. Wang, T. Zhang, and Z. Han, "Three-Dimensional Area Coverage with UAV Swarm based on Deep Reinforcement Learning," vol. 39, no. 10,

pp. 1–6, 2021, doi: 10.1109/icc42927.2021.9500895.

[26] X. Liu, Y. Liu, and Y. Chen, “Reinforcement learning in multiple-UAV networks: Deployment and movement design,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8036–8049, 2019, doi: 10.1109/TVT.2019.2922849.

[27] X. Deng, J. Li, P. Guan, and L. Zhang, “Energy-Efficient UAV-Aided Target Tracking Systems Based on Edge Computing,” *IEEE Internet Things J.*, vol. 4662, no. c, pp. 1–8, 2021, doi: 10.1109/JIOT.2021.3091216.

[28] L. Zhou, S. Leng, Q. Liu, and Q. Wang, “Intelligent UAV Swarm Cooperation for Multiple Targets Tracking,” *IEEE Internet Things J.*, vol. 4662, no. c, pp. 1–12, 2021, doi: 10.1109/JIOT.2021.3085673.

[29] D. Chen, Q. Qi, Z. Zhuang, J. Wang, J. Liao, and Z. Han, “Mean Field Deep Reinforcement Learning for Fair and Efficient UAV Control,” *IEEE Internet Things J.*, vol. 8, no. 2, pp. 813–828, Jan. 2021, doi: 10.1109/JIOT.2020.3008299.

[30] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, “Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, 2018, doi: 10.1109/JSAC.2018.2864373.

[31] X. Jian, P. Leng, Y. Wang, M. Alrashoud, and M. S. Hossain, “Blockchain-Empowered Trusted Networking for Unmanned Aerial Vehicles in the B5G Era,” *IEEE Netw.*, vol. 35, no. 1, pp. 72–77, 2021, doi: 10.1109/MNET.011.2000177.

[32] Yueh-Ting Wu, Wanjiun Liao, Cheng-Lin Tsao, and Tsung-Nan Lin, “Impact of Node Mobility on Link Duration in Multihop Mobile Networks,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2435–2442, 2009, doi: 10.1109/TVT.2008.2008190.

[33] M. Y. Arafat and S. Moh, “A Q -Learning-Based Topology-Aware Routing Protocol for Flying Ad Hoc Networks,” *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1985–2000, Feb. 2022, doi: 10.1109/JIOT.2021.3089759.

[34] L. Hong, H. Guo, J. Liu, and Y. Zhang, “Toward Swarm Coordination: Topology-Aware Inter-UAV Routing Optimization,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10177–10187, Sep. 2020, doi: 10.1109/TVT.2020.3003356.

[35] B. Wang, Y. Sun, T. Do-Duy, E. Garcia-Palacios, and T. Q. Duong, “Adaptive $ D$-Hop Connected Dominating Set in Highly Dynamic Flying Ad-Hoc Networks,” *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2651–2664, Jul. 2021, doi: 10.1109/TNSE.2021.3103873.

[36] K. Namuduri and R. Pendse, “Analytical estimation of path duration in mobile ad hoc networks,” *IEEE Sens. J.*, vol. 12, no. 6, pp. 1828–1835, 2012, doi: 10.1109/JSEN.2011.2176927.

[37] W. Liu, B. Li, W. Xie, Y. Dai, and Z. Fei, “Energy Efficient Computation Offloading in Aerial Edge Networks With Multi-Agent Cooperation,” *IEEE Trans. Wirel. Commun.*, pp. 1–1, 2023, doi: 10.1109/TWC.2023.3235997.

[38] A. Trotta, M. Di Felice, F. Montori, K. R. Chowdhury, and L. Bononi, "Joint Coverage, Connectivity, and Charging Strategies for Distributed UAV Networks," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 883–900, 2018, doi: 10.1109/TRO.2018.2839087.

[39] N. Kumar, M. Ghosh, and C. Singhal, "UAV network for surveillance of inaccessible regions with zero blind spots," *IEEE INFOCOM 2020 - IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPS 2020*, pp. 1213–1218, 2020, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162686.

[40] M. M. Alam, M. Y. Arafat, S. Moh, and J. Shen, "Topology control algorithms in multi-unmanned aerial vehicle networks: An extensive survey," *J. Netw. Comput. Appl.*, vol. 207, no. August, p. 103495, Nov. 2022, doi: 10.1016/j.jnca.2022.103495.

[41] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, "UAV-Enabled Secure Communications by Multi-Agent Deep Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11599–11611, 2020, doi: 10.1109/TVT.2020.3014788.

[42] J. Wang, C. Jiang, Z. Han, and Y. Ren, "Taking Drones to the Next Level," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, 2017.

[43] S. Poudel and S. Moh, "Task assignment algorithms for unmanned aerial vehicle networks: A comprehensive survey," *Veh. Commun.*, vol. 35, p. 100469, Jun. 2022, doi: 10.1016/j.vehcom.2022.100469.

[44] A. A. Aziz, Y. A. Şekercioğlu, P. Fitzpatrick, and M. Ivanovich, "A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 1, pp. 121–144, 2013, doi: 10.1109/SURV.2012.031612.00124.

[45] Z. Zhao, C. Liu, X. Guang, and K. Li, "A Transmission-Reliable Topology Control Framework Based on Deep Reinforcement Learning for UWSNs," *IEEE Internet Things J.*, pp. 1–1, 2023, doi: 10.1109/JIOT.2023.3262690.

[46] M. Lúis, R. Oliveira, L. Bernardo, A. Garrido, and P. Pinto, "Joint topology control and routing in ad hoc vehicular networks," *2010 Eur. Wirel. Conf. EW 2010*, pp. 528–535, 2010, doi: 10.1109/EW.2010.5483482.

[47] A. Steinbusch and M. Reyhanoglu, "Robust Nonlinear Output Feedback Control of a 6-DOF Quadrotor UAV," *2019 12th Asian Control Conf. ASCC 2019*, pp. 1655–1660, 2019.

[48] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying Ad-Hoc Networks (FANETs): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013, doi: 10.1016/j.adhoc.2012.12.004.

[49] H. Wang, H. Zhao, J. Zhang, D. Ma, J. Li, and J. Wei, "Survey on Unmanned Aerial Vehicle Networks: A Cyber Physical System Perspective," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1027–1070, 2020, doi: 10.1109/COMST.2019.2962207.

[50] H. Zhao, H. Wang, W. Wu, and J. Wei, "Deployment algorithms for UAV airborne networks toward on-demand coverage," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9,

pp. 2015–2031, 2018, doi: 10.1109/JSAC.2018.2864376.

[51] R. Chen, X. Li, Y. Sun, S. Li, and Z. Sun, "Multi-UAV Coverage Scheme for Average Capacity Maximization," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 653–657, 2020, doi: 10.1109/LCOMM.2019.2962774.

[52] S. C. Noh, H. B. Jeon, and C. B. Chae, "Energy-Efficient Deployment of Multiple UAVs Using Ellipse Clustering to Establish Base Stations," *IEEE Wirel. Commun. Lett.*, vol. 9, no. 8, pp. 1155–1159, 2020, doi: 10.1109/LWC.2020.2982889.

[53] K. Guo, C. Wang, Z. Li, D. W. K. Ng, and K. K. Wong, "Multiple UAV-Borne IRS-Aided Millimeter Wave Multicast Communications: A Joint Optimization Framework," *IEEE Commun. Lett.*, vol. 7798, no. c, pp. 1–5, 2021, doi: 10.1109/LCOMM.2021.3111602.

[54] Y. Q. Chen and Z. Wang, "Formation control: A review and a new consideration," *2005 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, no. 435, pp. 3181–3186, 2005, doi: 10.1109/IROS.2005.1545539.

[55] S. R. Yeduri, N. S. Chilamkurthy, O. J. Pandey, and L. R. Cenkeramaddi, "Energy and Throughput Management in Delay-Constrained Small-World UAV-IoT Network," *IEEE Internet Things J.*, vol. 14, no. 8, pp. 1–1, 2022, doi: 10.1109/JIOT.2022.3231644.

[56] M. Y. Arafat and S. Moh, "Localization and Clustering Based on Swarm Intelligence in UAV Networks for Emergency Communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8958–8976, Oct. 2019, doi: 10.1109/JIOT.2019.2925567.

[57] H. Zhao, H. Liu, Y. W. Leung, and X. Chu, "Self-Adaptive Collective Motion of Swarm Robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1533–1545, 2018, doi: 10.1109/TASE.2018.2840828.

[58] H. Kang, W. Wang, C. Yang, and Z. Li, "Leader-Following Formation Control and Collision Avoidance of Second-Order Multi-Agent Systems with Time Delay," *IEEE Access*, vol. 8, pp. 142571–142580, 2020, doi: 10.1109/ACCESS.2020.3012992.

[59] M. Y. Arafat and S. Moh, "Bio-inspired approaches for energy-efficient localization and clustering in uav networks for monitoring wildfires in remote areas," *IEEE Access*, vol. 9, pp. 18649–18669, 2021, doi: 10.1109/ACCESS.2021.3053605.

[60]Y. Yang, Y. Xiao, and T. Li, "A Survey of Autonomous Underwater Vehicle Formation: Performance, Formation Control, and Communication Capability," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 2, pp. 815–841, 2021, doi: 10.1109/COMST.2021.3059998.

[61] Y. Ding, X. Wang, Y. Cong, and H. Li, "Scalability analysis of algebraic graph-based multi-UAVs formation control," *IEEE Access*, vol. 7, pp. 129719–129733, 2019, doi: 10.1109/ACCESS.2019.2938991.

[62] J. Liu *et al.*, "QMR:Q-learning based Multi-objective optimization Routing protocol for Flying Ad Hoc Networks," *Comput. Commun.*, vol. 150, pp. 304–316, Jan. 2020, doi: 10.1016/j.comcom.2019.11.011.

[63] X. Gu, G. Zhang, M. Wang, W. Duan, M. Wen, and P.-H. Ho, "UAV-aided Energy

Efficient Edge Computing Networks: Security Offloading Optimization," *IEEE Internet Things J.*, vol. 4662, no. c, pp. 1–1, 2021, doi: 10.1109/jiot.2021.3103391.

[64] W. Xu, L. Xiang, T. Zhang, M. Pan, and Z. Han, "Cooperative Control of Physical Collision and Transmission Power for UAV Swarm: A Dual-Fields Enabled Approach," *IEEE Internet Things J.*, vol. 4662, no. c, pp. 1–15, 2021, doi: 10.1109/JIOT.2021.3096955.

[65] M. Y. Arafat, S. Poudel, and S. Moh, "Medium Access Control Protocols for Flying Ad Hoc Networks: A Review," *IEEE Sens. J.*, vol. 21, no. 4, pp. 4097–4121, 2021, doi: 10.1109/JSEN.2020.3034600.

[66]S. Park, H. T. Kim, and H. Kim, "Energy-efficient topology control for UAV networks," *Energies*, vol. 12, no. 23, pp. 1–19, 2019, doi: 10.3390/en12234523.

[67] X. Huang, A. Liu, H. Zhou, K. Yu, W. Wang, and X. Shen, "FMAC: A Self-Adaptive MAC Protocol for Flocking of Flying Ad Hoc Network," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 610–625, 2021, doi: 10.1109/JIOT.2020.3007071.

[68] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automat. Contr.*, vol. 51, no. 3, pp. 401–420, 2006, doi: 10.1109/TAC.2005.864190.

[69] Y. Wan, J. Tang, and S. Lao, "Distributed Conflict-Detection and Resolution Algorithm for UAV Swarms Based on Consensus Algorithm and Strategy Coordination," *IEEE Access*, vol. 7, pp. 100552–100566, 2019, doi: 10.1109/ACCESS.2019.2928034.

[70] J. Lwowski, S. Member, A. Majumdar, and S. Member, "Bird Flocking Inspired Formation Control for," vol. 13, no. 3, pp. 3580–3589, 2019.

[71] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Proc. 14th Annu. Conf. Comput. Graph. Interact. Tech. SIGGRAPH 1987*, vol. 21, no. 4, pp. 25–34, 1987, doi: 10.1145/37401.37406.

[72] K. K. Oh, M. C. Park, and H. S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015, doi: 10.1016/j.automatica.2014.10.022.

[73] T. J. Choi and C. W. Ahn, "Artificial life based on boids model and evolutionary chaotic neural networks for creating artworks," *Swarm Evol. Comput.*, vol. 47, no. July, pp. 80–88, 2019, doi: 10.1016/j.swevo.2017.09.003.

[74] T. Dapper e Silva, C. F. Emygdio de Melo, P. Cumino, D. Rosario, E. Cerqueira, and E. Pignaton de Freitas, "STFANET: SDN-Based Topology Management for Flying Ad Hoc Network," *IEEE Access*, vol. 7, pp. 173499–173514, 2019, doi: 10.1109/ACCESS.2019.2956724.

[75] K. A. M, F. Hu, and S. Kumar, "Deep Q-Learning Based Node Positioning for Throughput-Optimal Communications in Dynamic UAV Swarm Network," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 554–566, 2019, doi: 10.1109/TCCN.2019.2907520.

[76] H. Zhao, J. Wei, S. Huang, L. Zhou, and Q. Tang, "Regular Topology Formation Based

on Artificial Forces for Distributed Mobile Robotic Networks," *IEEE Trans. Mob. Comput.*, vol. 18, no. 10, pp. 2415–2429, Oct. 2019, doi: 10.1109/TMC.2018.2873015.

[77] L. Wang, H. Zhang, S. Guo, and D. Yuan, "Communication-, Computation-, and Control-Enabled UAV Mobile Communication Networks," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20393–20407, Oct. 2022, doi: 10.1109/JIOT.2022.3172358.

[78] K. Derr and M. Manic, "Extended Virtual Spring Mesh (EVSM): The Distributed Self-Organizing Mobile Ad Hoc Network for Area Exploration," *IEEE Trans. Ind. Electron.*, vol. 58, no. 12, pp. 5424–5437, Dec. 2011, doi: 10.1109/TIE.2011.2130492.

[79] Z. Pan, Z. Sun, H. Deng, and D. Li, "A Multilayer Graph for Multiagent Formation and Trajectory Tracking Control Based on MPC Algorithm," *IEEE Trans. Cybern.*, vol. PP, pp. 1–12, 2021, doi: 10.1109/tcyb.2021.3119330.

[80] J. ZHAO, J. SUN, Z. CAI, Y. WANG, and K. WU, "Distributed coordinated control scheme of UAV swarm based on heterogeneous roles," *Chinese J. Aeronaut.*, 2021, doi: 10.1016/j.cja.2021.01.014.

[81] G. Shen *et al.*, "Deep Reinforcement Learning for Flocking Motion of Multi-UAV Systems: Learn From a Digital Twin," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11141–11153, Jul. 2022, doi: 10.1109/JIOT.2021.3127873.

[82] H. Shiri, J. Park, and M. Bennis, "Communication-Efficient Massive UAV Online Path Control: Federated Learning Meets Mean-Field Game Theory," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6840–6857, Nov. 2020, doi: 10.1109/TCOMM.2020.3017281.

[83] D. Y. Kim and J. W. Lee, "Joint Mission Assignment and Topology Management in the Mission-Critical FANET," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2368–2385, 2020, doi: 10.1109/JIOT.2019.2958130.

[84] D. Y. Kim and J. W. Lee, "Integrated Topology Management in Flying Ad Hoc Networks: Topology Construction and Adjustment," *IEEE Access*, vol. 6, pp. 61196–61211, 2018, doi: 10.1109/ACCESS.2018.2875679.

[85] X. Qi, X. Gu, Q. Zhang, and Z. Yang, "A Link-Prediction Based Multi-CDSs Scheduling Mechanism for FANET Topology Maintenance," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 280, Springer International Publishing, 2019, pp. 587–601.

[86] S. Bhandari, X. Wang, and R. Lee, "Mobility and Location-Aware Stable Clustering Scheme for UAV Networks," *IEEE Access*, vol. 8, pp. 106364–106372, 2020, doi: 10.1109/ACCESS.2020.3000222.

[87] X. Qi, P. Yuan, Q. Zhang, and Z. Yang, "CDS-Based Topology Control in FANETs via Power and Position Optimization," *IEEE Wirel. Commun. Lett.*, vol. 9, no. 12, pp. 2015–2019, Dec. 2020, doi: 10.1109/LWC.2020.3009666.

[88] F. Xiong *et al.*, "Energy-Saving Data Aggregation for Multi-UAV System," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9002–9016, Aug. 2020, doi: 10.1109/TVT.2020.2999374.

[89] B. Wang, Y. Sun, T. Do-Duy, E. Garcia-Palacios, and T. Q. Duong, "Adaptive d-Hop Connected Dominating Set in Highly Dynamic Flying Ad-hoc Networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 4697, no. c, pp. 1–1, 2021, doi: 10.1109/tnse.2021.3103873.

[90] L. Ruan *et al.*, "Cooperative Relative Localization for UAV Swarm in GNSS-Denied Environment : A Coalition," vol. XX, no. XX, 2021, doi: 10.1109/JIOT.2021.3130000.

[91] J. Chen *et al.*, "A Multi-Leader Multi-Follower Stackelberg Game for Coalition-Based UAV MEC Networks," *IEEE Wirel. Commun. Lett.*, vol. 10, no. 11, pp. 2350–2354, Nov. 2021, doi: 10.1109/LWC.2021.3100113.

[92] N. Xing, Q. Zong, L. Dou, B. Tian, and Q. Wang, "A Game Theoretic Approach for Mobility Prediction Clustering in Unmanned Aerial Vehicle Networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 9963–9973, 2019, doi: 10.1109/TVT.2019.2936894.

[93] Q. Wu *et al.*, "Joint Computation Offloading, Role, and Location Selection in Hierarchical Multicoalition UAV MEC Networks: A Stackelberg Game Learning Approach," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18293–18304, Oct. 2022, doi: 10.1109/JIOT.2022.3158489.

[94] A. Khan, F. Aftab, and Z. Zhang, "Self-organization based clustering scheme for FANETs using Glowworm Swarm Optimization," *Phys. Commun.*, vol. 36, p. 100769, Oct. 2019, doi: 10.1016/j.phycom.2019.100769.

[95] A. Trotta, L. Montecchiari, M. Di Felice, and L. Bononi, "A GPS-Free Flocking Model for Aerial Mesh Deployments in Disaster-Recovery Scenarios," *IEEE Access*, vol. 8, pp. 91558–91573, 2020, doi: 10.1109/ACCESS.2020.2994466.

[96] M. Chen, F. Dai, H. Wang, and L. Lei, "DFM: A Distributed Flocking Model for UAV Swarm Networks," *IEEE Access*, vol. 6, pp. 69141–69150, 2018, doi: 10.1109/ACCESS.2018.2880485.

[97] S. M. Hung, S. N. Givigi, and A. Noureldin, "A Dyna-Q (Lambda) Approach to Flocking with Fixed-Wing UAVs in a Stochastic Environment," *Proc. - 2015 IEEE Int. Conf. Syst. Man, Cybern. SMC 2015*, pp. 1918–1923, 2016, doi: 10.1109/SMC.2015.335.

[98] F. Dai, M. Chen, X. Wei, and H. Wang, "Swarm Intelligence-Inspired Autonomous Flocking Control in UAV Networks," *IEEE Access*, vol. 7, pp. 61786–61796, 2019, doi: 10.1109/ACCESS.2019.2916004.

[99] W. You, C. Dong, X. Cheng, X. Zhu, Q. Wu, and G. Chen, "Joint Optimization of Area Coverage and Mobile-Edge Computing with Clustering for FANETs," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 695–707, 2021, doi: 10.1109/JIOT.2020.3006891.

[100] X. Cheng, C. Dong, H. Dai, and G. Chen, "MOOC: A Mobility Control Based Clustering Scheme for Area Coverage in FANETs," *19th IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks, WoWMoM 2018*, 2018, doi: 10.1109/WoWMoM.2018.8449771.

[101] J. Wu *et al.*, "Autonomous Cooperative Flocking for Heterogeneous Unmanned

Aerial Vehicle Group," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 12477–12490, Dec. 2021, doi: 10.1109/TVT.2021.3124898.

[102] A. Trotta, M. Di Felice, K. R. Chowdhury, and L. Bononi, "Fly and recharge: Achieving persistent coverage using Small Unmanned Aerial Vehicles (SUAVs)," *IEEE Int. Conf. Commun.*, 2017, doi: 10.1109/ICC.2017.7996482.

[103] M. Di Felice, A. Trotta, L. Bedogni, K. R. Chowdhury, and L. Bononi, "Self-organizing aerial mesh networks for emergency communication," *IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. PIMRC*, vol. 2014-June, pp. 1631–1636, 2014, doi: 10.1109/PIMRC.2014.7136429.

[104] Z. Chen, X. Fu, and X. Gao, "Formation and Conical Obstacle Avoidance Control of UAS Based on Two-hop Network," *Eur. Control Conf. 2020, ECC 2020*, pp. 1967–1972, 2020, doi: 10.23919/ecc51009.2020.9143626.

[105] A. V. Leonov, G. A. Litvinov, and D. A. Korneev, "Simulation and Analysis of Transmission Range Effect on AODV and OLSR Routing Protocols in Flying Ad Hoc Networks (FANETs) formed by Mini-UAVs with Different Node Density," *2018 Syst. Signal Synchronization, Gener. Process. Telecommun. SYNCHROINFO 2018*, 2018, doi: 10.1109/SYNCHROINFO.2018.8457014.

[106] P. Xie, "An enhanced OLSR routing protocol based on node link expiration time and residual energy in ocean FANETS," *2018 24th Asia-Pacific Conf. Commun. APCC 2018*, pp. 598–603, 2019, doi: 10.1109/APCC.2018.8633484.

[107] A. Garcia-Santiago, J. Castaneda-Camacho, J. F. Guerrero-Castellanos, and G. Mino-Aguilar, "Evaluation of AODV and DSDV routing protocols for a FANET: Further results towards robotic vehicle networks," *9th IEEE Lat. Am. Symp. Circuits Syst. LASCAS 2018 - Proc.*, pp. 1–4, 2018, doi: 10.1109/LASCAS.2018.8399972.

[108] S. Garg, A. Ihler, E. S. Bentley, and S. Kumar, "A Cross-Layer, Mobility and Congestion-Aware Routing Protocol for UAV Networks," *IEEE Trans. Aerosp. Electron. Syst.*, pp. 1–18, 2022, doi: 10.1109/TAES.2022.3232322.

[109] T. Li *et al.*, "A mean field game-theoretic cross-layer optimization for multi-hop swarm UAV communications," *J. Commun. Networks*, vol. 24, no. 1, pp. 68–82, Feb. 2022, doi: 10.23919/JCN.2021.000035.

[110] J. Guo *et al.*, "ICRA: An Intelligent Clustering Routing Approach for UAV Ad Hoc Networks," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–14, 2022, doi: 10.1109/TITS.2022.3145857.

[111] X. Tan, Z. Zuo, S. Su, X. Guo, X. Sun, and D. Jiang, "Performance Analysis of Routing Protocols for UAV Communication Networks," *IEEE Access*, vol. 8, pp. 92212–92224, 2020, doi: 10.1109/ACCESS.2020.2995040.

[112] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, and M. B. Yagoubi, "A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs)," *Veh. Commun.*, vol. 10, no. October, pp. 29–56, 2017, doi: 10.1016/j.vehcom.2017.10.003.

[113] W. Jung, J. Yim, and Y. Ko, "QGeo: Q-Learning-Based Geographic Ad Hoc Routing Protocol for Unmanned Robotic Networks," *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2258–2261, Oct. 2017, doi: 10.1109/LCOMM.2017.2656879.

[114] C. Yan *et al.*, "Collision-Avoiding Flocking With Multiple Fixed-Wing UAVs in Obstacle-Cluttered Environments: A Task-Specific Curriculum-Based MADRL Approach," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–15, 2023, doi: 10.1109/TNNLS.2023.3245124.

[115] S. Guo and X. Zhao, "Multi-Agent Deep Reinforcement Learning Based Transmission Latency Minimization for Delay-Sensitive Cognitive Satellite-UAV Networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 131–144, Jan. 2023, doi: 10.1109/TCOMM.2022.3222460.

[116] H. Jiang, M. Cui, D. W. K. Ng, and L. Dai, "Accurate Channel Prediction Based on Transformer: Making Mobility Negligible," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2717–2732, Sep. 2022, doi: 10.1109/JSAC.2022.3191334.

[117] Z. Wang *et al.*, "Learning to Routing in UAV Swarm Network: A Multi-agent Reinforcement Learning Approach," *IEEE Trans. Veh. Technol.*, vol. 14, no. 8, pp. 1–14, 2023, doi: 10.1109/TVT.2022.3232815.

[118] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," in *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, Jun. 2020, pp. 98–101, doi: 10.1109/IWECAI50956.2020.00027.

[119] M. M. Alam and S. Moh, "Joint Trajectory Control, Frequency Allocation, and Routing for UAV Swarm Networks: A Multi-Agent Deep Reinforcement Learning Approach", submitted to *IEEE Internet of Things Journal*, Feb. 2023.

[120] M. M. Alam and S. Moh, "Q-learning-based routing inspired by adaptive flocking control for collaborative unmanned aerial vehicle swarms," *Veh. Commun.*, vol. 40, p. 100572, Apr. 2023, doi: 10.1016/j.vehcom.2023.100572.

[121] X. Chu and H. Ye, "Parameter Sharing Deep Deterministic Policy Gradient for Cooperative Multi-agent Reinforcement Learning," Oct. 2017, [Online]. Available: http://arxiv.org/abs/1710.00336.

[122] X. Qiu, Y. Yang, L. Xu, J. Yin, and Z. Liao, "Maintaining Links in the Highly Dynamic FANET Using Deep Reinforcement Learning," *IEEE Trans. Veh. Technol.*, pp. 1–15, 2022, doi: 10.1109/TVT.2022.3217888.

[123] C. Yan *et al.*, "PASCAL: PopulAtion-Specific Curriculum-based MADRL for collision-free flocking with large-scale fixed-wing UAV swarms," *Aerosp. Sci. Technol.*, vol. 133, p. 108091, 2023, doi: 10.1016/j.ast.2022.108091.

[124] L. Zhang, X. Ma, Z. Zhuang, H. Xu, V. Sharma, and Z. Han, "$Q$-Learning Aided Intelligent Routing With Maximum Utility in Cognitive UAV Swarm for Emergency Communications," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3707–3723, Mar.

2023, doi: 10.1109/TVT.2022.3221538.

[125] W. Jin, R. Gu, and Y. Ji, "Reward Function Learning for Q-learning-Based Geographic Routing Protocol," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1236–1239, Jul. 2019, doi: 10.1109/LCOMM.2019.2913360.

[126] B. Sliwa, C. Schuler, M. Patchou, and C. Wietfeld, "PARRoT: Predictive Ad-hoc Routing Fueled by Reinforcement Learning and Trajectory Knowledge," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, Apr. 2021, vol. 2021-April, pp. 1–7, doi: 10.1109/VTC2021-Spring51267.2021.9448959.

[127] L. A. L. F. da Costa, R. Kunst, and E. Pignaton de Freitas, "Q-FANET: Improved Q-learning based routing protocol for FANETs," *Comput. Networks*, vol. 198, p. 108379, Oct. 2021, doi: 10.1016/j.comnet.2021.108379.

[128] Y. Cui, Q. Zhang, Z. Feng, Z. Wei, C. Shi, and H. Yang, "Topology-Aware Resilient Routing Protocol for FANETs: An Adaptive Q -Learning Approach," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18632–18649, Oct. 2022, doi: 10.1109/JIOT.2022.3162849.

[129] Q. Wu *et al.*, "Routing protocol for heterogeneous FANETs with mobility prediction," *China Commun.*, vol. 19, no. 1, pp. 186–201, Jan. 2022, doi: 10.23919/JCC.2022.01.014.

[130] A. Rovira-Sugranes, F. Afghah, J. Qu, and A. Razi, "Fully-Echoed Q-Routing With Simulated Annealing Inference for Flying Adhoc Networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2223–2234, Jul. 2021, doi: 10.1109/TNSE.2021.3085514.

[131] M. Zhang, C. Dong, P. Yang, T. Tao, Q. Wu, and T. Q. S. Quek, "Adaptive Routing Design for Flying Ad Hoc Networks," *IEEE Commun. Lett.*, vol. 26, no. 6, pp. 1438–1442, Jun. 2022, doi: 10.1109/LCOMM.2022.3152832.

[132] J. Liu, Q. Wang, and Y. Xu, "AR-GAIL: Adaptive routing protocol for FANETs using generative adversarial imitation learning," *Comput. Networks*, vol. 218, no. September, p. 109382, Dec. 2022, doi: 10.1016/j.comnet.2022.109382.

[133] R. Ding, J. Chen, W. Wu, J. Liu, F. Gao, and X. Shen, "Packet Routing in Dynamic Multi-Hop UAV Relay Network: A Multi-Agent Learning Approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 10059–10072, Sep. 2022, doi: 10.1109/TVT.2022.3182335.

[134] J. Wang, X. Zhang, X. He, and Y. Sun, "Bandwidth Allocation and Trajectory Control in UAV-Assisted IoV Edge Computing Using Multiagent Reinforcement Learning," *IEEE Trans. Reliab.*, pp. 1–10, 2022, doi: 10.1109/TR.2022.3192020.

[135] X. Qiu, L. Xu, P. Wang, Y. Yang, and Z. Liao, "A Data-Driven Packet Routing Algorithm for an Unmanned Aerial Vehicle Swarm: A Multi-Agent Reinforcement Learning Approach," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 10, pp. 2160–2164, Oct. 2022, doi: 10.1109/LWC.2022.3195963.

[136] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory Design and Power Control for Multi-UAV Assisted Wireless Networks: A Machine Learning Approach," *IEEE Trans.*

*Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019, doi: 10.1109/TVT.2019.2920284.

[137] C. He, Q. Wang, Y. Xu, J. Liu, and Y. Xu, "A q-learning based cross-layer transmission protocol for MANETs," *Proc. - 2019 IEEE Int. Conf. Ubiquitous Comput. Commun. Data Sci. Comput. Intell. Smart Comput. Netw. Serv. IUCC/DSCI/SmartCNS 2019*, pp. 580–585, 2019, doi: 10.1109/IUCC/DSCI/SmartCNS.2019.00122.

[138] R. Ding, F. Gao, and X. S. Shen, "3D UAV Trajectory Design and Frequency Band Allocation for Energy-Efficient and Fair Communication: A Deep Reinforcement Learning Approach," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 12, pp. 7796–7809, Dec. 2020, doi: 10.1109/TWC.2020.3016024.

[139] X. Jiang, M. Sheng, N. Zhao, C. Xing, W. Lu, and X. Wang, "Green UAV communications for 6G: A survey," *Chinese J. Aeronaut.*, no. June, 2021, doi: 10.1016/j.cja.2021.04.025.

[140] S. Hwang, H. Lee, J. Park, and I. Lee, "Decentralized Computation Offloading with Cooperative UAVs: Multi-Agent Deep Reinforcement Learning Perspective," *IEEE Wirel. Commun.*, vol. 29, no. 4, pp. 24–31, 2022, doi: 10.1109/MWC.003.2100690.

[141] H. Song, W. Choi, and H. Kim, "Robust Vision-Based Relative-Localization Approach Using an RGB-Depth Camera and LiDAR Sensor Fusion," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3725–3736, 2016, doi: 10.1109/TIE.2016.2521346.

[142] P. Mittal, R. Singh, and A. Sharma, "Deep learning-based object detection in low-altitude UAV datasets: A survey," *Image Vis. Comput.*, vol. 104, p. 104046, 2020, doi: 10.1016/j.imavis.2020.104046.

[143] M. Y. Arafat, M. M. Alam, and S. Moh, "Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges," *Drones*, vol. 7, no. 2, p. 89, Jan. 2023, doi: 10.3390/drones7020089.

[144] S. Guler, M. Abdelkader, and J. S. Shamma, "Peer-to-Peer Relative Localization of Aerial Robots With Ultrawideband Sensors," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 5, pp. 1981–1996, Sep. 2021, doi: 10.1109/TCST.2020.3027627.

[145] S. M. Hung and S. N. Givigi, "A Q-Learning Approach to Flocking with UAVs in a Stochastic Environment," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 186–197, 2017, doi: 10.1109/TCYB.2015.2509646.

[146] A. Kumar, K. Sharma, H. Singh, S. G. Naugriya, S. S. Gill, and R. Buyya, "A drone-based networked system and methods for combating coronavirus disease (COVID-19) pandemic," *Futur. Gener. Comput. Syst.*, vol. 115, pp. 1–19, 2021, doi: 10.1016/j.future.2020.08.046.

[147] Y. Miao, Y. Tang, B. A. Alzahrani, A. Barnawi, T. Alafif, and L. Hu, "Airborne LiDAR Assisted Obstacle Recognition and Intrusion Detection towards Unmanned Aerial Vehicle: Architecture, Modeling and Evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4531–4540, 2021, doi: 10.1109/TITS.2020.3023189.

[148] O. S. Oubbati, A. Lakas, P. Lorenz, M. Atiquzzaman, and A. Jamalipour, "Leveraging communicating UAVs for emergency vehicle guidance in Urban Areas," *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 2, pp. 1070–1082, 2021, doi: 10.1109/TETC.2019.2930124.

[149] S. Minaeian, J. Liu, and Y. J. Son, "Vision-Based Target Detection and Localization via a Team of Cooperative UAV and UGVs," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 7, pp. 1005–1016, 2016, doi: 10.1109/TSMC.2015.2491878.

[150] M. M. Alam and S. Moh, "Survey on Q-Learning-Based Position-Aware Routing Protocols in Flying Ad Hoc Networks," *Electron.*, vol. 11, no. September, p. 2021, 2022, doi: 10.3390/electronics11071099.

[151] C. Dixon and E. W. Frew, "Optimizing cascaded chains of unmanned aircraft acting as communication relays," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 5, pp. 883–898, 2012, doi: 10.1109/JSAC.2012.120605.

[152] Y. Wu, B. Zhang, S. Yang, X. Yi, and X. Yang, "Energy-efficient joint communication-motion planning for relay-assisted wireless robot surveillance," *Proc. - IEEE INFOCOM*, 2017, doi: 10.1109/INFOCOM.2017.8057072.

[153] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of adaptive modulation and Coding with truncated ARQ over wireless links," *IEEE Trans. Wirel. Commun.*, vol. 3, no. 5, pp. 1746–1755, 2004, doi: 10.1109/TWC.2004.833474.

[154] A. Tomar, L. Muduli, and P. K. Jana, "A Fuzzy Logic-based On-demand Charging Algorithm for Wireless Rechargeable Sensor Networks with Multiple Chargers," *IEEE Trans. Mob. Comput.*, vol. 1233, no. c, pp. 1–1, 2020, doi: 10.1109/tmc.2020.2990419.

[155] A. Serhani, N. Naja, and A. Jamali, "QLAR: A Q-learning based adaptive routing for MANETs," *2016 IEEE/ACS 13th Int. Conf. Comput. Syst. Appl.*, 2016.

[156] N. U. Prabhu, D. Gross, and C. M. Harris, *Fundamentals of Queueing Theory.*, vol. 82, no. 399. 1987.

[157] X. Hong, M. Gerla, G. Pei, and C. C. Chiang, "A group mobility model for ad hoc wireless networks," *Proc. 2nd ACM Int. Work. Model. Anal. Simul. Wirel. Mob. Syst. MSWiM 1999*, pp. 53–60, 1999, doi: 10.1145/313237.313248.

[158] L. Ruan *et al.*, "Cooperative Relative Localization for UAV Swarm in GNSS-Denied Environment: A Coalition Formation Game Approach," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11560–11577, Jul. 2022, doi: 10.1109/JIOT.2021.3130000.

[159] M. M. Alam and S. Moh, "Joint topology control and routing in a UAV swarm for crowd surveillance," *J. Netw. Comput. Appl.*, vol. 204, no. January, p. 103427, Aug. 2022, doi: 10.1016/j.jnca.2022.103427.

[160] A. H. Arani, M. M. Azari, P. Hu, Y. Zhu, H. Yanikomeroglu, and S. Safavi-Naeini, "Reinforcement Learning for Energy-Efficient Trajectory Design of UAVs," *IEEE Internet Things J.*, vol. 4662, no. c, pp. 1–11, 2021, doi: 10.1109/JIOT.2021.3118322.

[161] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of queuing with

adaptive modulation and coding over wireless links," *Proc. - IEEE Mil. Commun. Conf. MILCOM*, vol. 1, no. 5, pp. 717–722, 2003, doi: 10.1109/milcom.2003.1290192.

[162] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT Enabled UAV: Network Architecture and Routing Algorithm," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3727–3742, Apr. 2019, doi: 10.1109/JIOT.2018.2890428.

[163] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*, 1987, vol. 21, no. 4, pp. 25–34, doi: 10.1145/37401.37406.

[164] J. Xiao, G. Yuan, J. He, K. Fang, and Z. Wang, "Graph attention mechanism based reinforcement learning for multi-agent flocking control in communication-restricted environment," *Inf. Sci. (Ny).*, vol. 620, pp. 142–157, Jan. 2023, doi: 10.1016/j.ins.2022.11.059.

[165] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 5261–5270, 2019.

[166] B. Chen, D. Liu, and L. Hanzo, "Decentralized Trajectory and Power Control Based on Multi-Agent Deep Reinforcement Learning in UAV Networks," *IEEE Int. Conf. Commun.*, vol. 2022-May, pp. 3983–3988, 2022, doi: 10.1109/ICC45855.2022.9838637.

[167] H. Mao, Z. Zhang, Z. Xiao, and Z. Gong, "Modelling the dynamic joint policy of teammates with attention multi-agent ddpg," *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS*, vol. 2, pp. 1108–1116, 2019.

[168] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, pp. 6380–6391, 2017.

[169] J. Tian, Q. Liu, H. Zhang, and D. Wu, "Multiagent Deep-Reinforcement-Learning-Based Resource Allocation for Heterogeneous QoS Guarantees for Vehicular Networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1683–1695, Feb. 2022, doi: 10.1109/JIOT.2021.3089823.

[170] T. Li *et al.*, "Joint Power Control and Scheduling for High-Dynamic Multi-Hop UAV Communication: A Robust Mean Field Game," *IEEE Access*, vol. 9, pp. 130649–130664, 2021, doi: 10.1109/ACCESS.2021.3113909.

[171] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.

[172] R. W. Liu, M. Liang, J. Nie, W. Y. B. Lim, Y. Zhang, and M. Guizani, "Deep Learning-Powered Vessel Trajectory Prediction for Improving Smart Traffic Services in Maritime Internet of Things," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3080–3094, Sep. 2022, doi: 10.1109/TNSE.2022.3140529.

[173] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[174] M. M. Alam and S. Moh, “Survey on Neighbor Discovery and Beam Alignment in mmWave-Enabled UAV Swarm Networks,” *Proc. of 11th Int. Conf. on Smart Media and Applications (SMA 2022)*, pp. 3-8, Saipan, USA, Oct. 19-22, 2022.

# Acknowledgements

I would like to express my sincere gratitude and appreciation to my advisor, Prof. Sangman Moh, for his unwavering support, guidance, and valuable suggestions throughout my Ph.D. studies. His mentorship has been invaluable in laying the foundation for the completion of my thesis, and I am truly thankful to him.

I am also immensely grateful to the members of my thesis committee, Prof. Seokjoo Shin, Prof. Moonsoo Kang, Prof. Wooyeol Choi, and Prof. Myeong-Hoon Oh, for their constructive feedback and insightful suggestions that have helped me to enhance and extend my research from various perspectives.

Furthermore, I would like to express my heartfelt thanks to all the members of the Mobile Computing Lab and my friends at Chosun University for their support and encouragement. I would also like to extend my gratitude to my uncle, Mr. Fakrul Alam, for his invaluable guidance and continuous support throughout my journey.

Last but not the least, I am extremely grateful to my parents, wife, teachers, and siblings for their motivation and encouragement during difficult times. Especially, I would like to express heartfelt thanks and admiration to my wife and parents for taking care of my beloved daughter in my absence. I want to convey my sincere apology to my three years old daughter Mashiyat Alam Ifza since as a father I could not provide her enough time due to my current situation of studying abroad.

Finally, I extend my special thanks to the Global Korea Scholarship (GKS) program for their financial support, encouragement, and assistance throughout my doctoral studies. Without the help of the National Institute for International Education (NIIED), South Korea, it would have been impossible for me to pursue my studies, and I am truly grateful for their invaluable guidance and support.