Multi-Objective MDP-based Routing In UAV Networks For Search-based Operations

Prateek Mahajan, Balamurugan P, Anusha Kumar, G S S Chalapathi, *Senior Member, IEEE*, Vinay Chamola, *Senior Member, IEEE*, and Maurice Khabbaz, *Senior Member, IEEE*

***Abstract*—Unmanned aerial vehicle (UAV) systems have gained widespread recognition due to their versatility and autonomy. Their deployment for disaster mitigation and management operations is seen as one of their most important applications over the past decade. In such UAV networks, routing plays a crucial role in determining network performance parameters such as network lifetime, data transmission latency, and packet delivery ratio. This paper presents a novel routing mechanism - Multi-Objective Markov Decision Based Routing (MOBMDP) for UAV networks carrying out search-based operations. MOBMDP models routing decisions in a Markov Decision Process (MDP) framework and uses Q-learning to take decisions. It compares routing paths using three metrics, viz., Remaining Energy of the Minimum Energy Node (REMEN), Power Distance ratio (PD), and Expected Delay (ED). Amongst these metrics, PD is a novel metric proposed by this work. PD simultaneously optimizes the energy efficiency and energy distribution in the network. Further, MOBMDP uses a novel reinforcement learning inspired method to estimate transmission delay in a given path. Intensive simulation studies compare MOBMDP to four state-of-the-art routing protocols. Results show a significant improvement in network lifetime, packet delivery ratio, energy efficiency, average data transmission delay, and error in delay estimation.**

***Index Terms*—UAV, search and rescue, placement algorithm, routing protocol, network lifetime, network coverage, transmission delay estimation, energy efficiency**

I. Introduction

THE evolution of wireless networks [1] and technology has brought about a fundamental shift in how systems are perceived and designed [2]. With a more flexible system at hand, new protocols for wireless networks have been developed to communicate effectively [3], [4], [5]. One of the most important domains in evolving wireless systems is Unmanned Aerial Vehicles (UAVs) [6]. Initially, UAV systems were deployed in military domains for carrying out border surveillance [7], search-and-destroy missions [8], and reconnaissance operations [9]. However, their versatility and autonomy have led to the deployment of UAV systems in various civilian applications, to name a few: *a*) safety-related missions [10], *b*) smart agriculture [11], *c*) traffic management [12], *d*) environmental monitoring [13], *e*) flying cellular base stations [14], [15], *f*) vehicular networks [16], [17], among others. In particular, Search-based Operations (SO) [18] constitute a key application of UAV networks. These include but are not restricted to: *a*) disaster management [19], *b*) search and rescue [20], *c*) reconnaissance missions [21], and, *d*) environmental surveillance [22]. In such operations, UAV networks are capable of providing a bird's eye view of the area of interest that enables controllers/operators to take appropriate actions. For sensitive missions such as search and rescue, a UAV network can be very efficiently assist in locating and rescuing multiple victims at once.

Usually, UAV systems deployed for SO consist of multiple UAVs working in coordination with one another. In general, multi-UAV systems are preferred over single UAVs due to low hardware complexity, reduced area scanning time, and reduced mission failure probability [23], [24]. Yet, the need for data communication and exchange among these UAVs necessitates the development and employment of efficient routing protocols that need to account for various challenges that are more pronounced in UAV networks compared to other ad-hoc networks. To list of a few of these challenges:

- Difficulty in locating nodes at all times due to high mobility in a three-dimensional environment
- Constrained power supply leading to limited lifetime
- Payload restrictions
- Lower node density compared to other ad-hoc networks

Keeping in mind the sensitivity of SOs, routing protocols must be designed to minimize transmission delay and maximize the network lifetime. This ensures that the entire area is inspected and important data is promptly transmitted to Ground Control Stations (GCSs).

This paper presents a novel routing algorithm - Multi-Objective Markov Decision Process-based routing (MOBMDP) to achieve such efficient transmission. Note that this work exploits UAV clusters such as described in Section III.

Under MOBMDP, the multi-UAV system is modeled as an MDP where the "agent" is a data packet, the "states" are the different Cluster Heads (CHs) that the packet may be transmitted to, the "action" is the transmission of a data packet from one CH to another and the "decision" is the decision of choosing the destination node of transmission. MOBMDP "rewards" MDP states (or, in this case, UAV CHs) based on three metrics, viz., Remaining Energy of the Minimum Energy Node (REMEN), Power Distance ratio (PD), and Expected Delay (ED). These rewards are awarded to the states using the concept of Q-learning. Decisions are then taken by maximizing the cumulative reward (also called "Q-value") of the allowed

Copyright © 2024 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Prateek Mahajan, Balamurugan P, Anusha Kumar, G S S Chalapathi and Vinay Chamola are with the Department of EEE, BITS-Pilani, Pilani, Rajasthan, India, 333031 (e-mail:{f20170317p@alumni, p20220455@pilani,f20170195p@alumni, gssc@pilani, vinay.chamola@pilani}.bits-pilani.ac.in)

Maurice Khabbaz is with the Computer Science Department of the American University of Beirut, Lebanon (e-mail: mk321@aub.edu.lb)

actions in a state. To this end, the fundamental contributions of this paper are elaborated as follows:

1) The design and implementation of a novel reinforcement learning (RL) inspired method to predict transmission delay. When compared to the state-of-the-art work [25] that predicts transmission delay, the proposed method reduces the prediction error by a factor of 10.
2) The proposal of an original and novel routing metric - Power Distance ratio (PD) to simultaneously optimize the energy distribution/efficiency of the decisions taken by the UAV network. This opposes existing routing protocols, which optimize either one of these two metrics.
3) A novel MDP and Q-learning based routing protocol-MOBMDP, designed specifically for multi-UAV systems that support search operations (SO).
4) Intensive simulations conducted to show that MOBMDP notably outperforms four state-of-the-art routing algorithms in terms of network lifetime, packet delivery ratio (PDR) and energy efficiency while maintaining similar performance in terms of data transmission delay.
5) This proposed work uses Network lifetime as one of the performance measures. A higher network lifetime contributes to an increase in the number of successful transmissions. The network lifetime analysis was not presented in competing Q-learning-based algorithms.
6) Extensive scalability analysis conducted showed a stable packet delivery ratio with minimal increase in delay as network size increased.

The remaining of the paper is organized as follows. Section II presents a brief overview of related routing algorithms, followed by a description of the system model considered in Section III. Section IV presents the details underlying the proposed MOBMDP protocol, while Section V presents an extensive comparison of MOBMDP against four state-of-the-art protocols. Finally, Section VI concludes the paper.

II. Related Works

UAV networks require different routing protocols as compared to other networks due to certain constraints imposed on individual nodes. Conventional routing protocols such as Optimized Link State Routing (OLSR) [26] and Ad-hoc On-Demand Distance Vector Routing (AODV) [27] have been modified to suit the requirements of UAV networks. In [28], an enhanced OLSR protocol is used, which takes into account mobility and delay prediction. It uses a self-adjusting Kalman filter model to identify multi-point relay (MPR) nodes. The routing table of OLSR is calculated by the Dijkstra algorithm to minimize the number of hops, and a cross-layer delay prediction model is used to establish low delay routing paths. Another modification of OLSR is Predictive OLSR (P-OLSR) [29] which takes advantage of the GPS information available on-board. Both simulation and real-world experiments show that P-OLSR shows better performance than OLSR. Another example of predictive routing is [25], which considers anticipated locations of UAVs for path selection. The path is selected using Dijkstra's shortest path algorithm. Results show that this algorithm shows superior delay performance when compared with conventional routing algorithms.

An example of AODV modification is seen in [30], which utilizes a combination of AODV, Langrangian interpolation, and artificial bee colony algorithm. The protocol is divided into three algorithms - the first algorithm computes the distance between each node, the second uses artificial bee colony algorithm to discover the position of all the nodes, and the third algorithm uses Langrangian interpolation to verify nodes in the network to pass on the information. Another example of modified AODV is [31], where the authors use a combination of AODV and greedy peripheral stateless routing protocol. The routing process is divided into two stages - the greedy routing stage and the flooding path-finding stage. Further, particle swarm optimization (PSO) is used to solve the sub-optimal choice problem of greedy forwarding.

Apart from modifications, several different protocols have been developed which perform much better than the traditional routing protocols. Specifically, for post-disaster operations, Arafat *et al.* [32] propose a location-aided delay-tolerant protocol. The protocol first establishes contact between search and ferry UAVs using the GPS location shared via messages. This is followed by location-aware single-copy data packet forwarding, and finally, data is forwarded from the search UAV to the base station by the shortest link available. In [33], Gankhuyag *et al.* propose a robust and reliable predictive routing strategy. Their hybrid scheme uses unicasting and geocasting routing using location and trajectory information. Robustness is ensured by predicting the intermediate node toward the predicted location, enabling a longer transmission range and keeping track of the changing topology. Additionally, reliability is achieved by reducing path re-establishment and service disruption time along with successful packet transmission. In GeoUAVs [34], Bousbaa *et al.* propose a protocol in which information is only transmitted to a specific group of UAVs. A source UAV sends geocast packets to all UAVs in its transmission region, and then these packets are further transmitted in the direction determined by the algorithm. Mukherjee *et al.* [35] propose a multi-armed bandit (MAB) based routing protocol, where path selection is based on the residual node energy for a better distribution of tasks among the nodes. The protocol outperforms the shortest path algorithm in terms of network lifetime.

The existing routing algorithms for other networks may not be suitable for FANETs due to the dynamic nature of the network. Therefore, it is crucial to choose a routing algorithm that can learn to act optimally. Q-learning is a model-free reinforcement learning algorithm that enables agents to learn and act optimally in a controlled Markovian environment [42]. Many works in literature have employed Q-learning to make routing decisions. J. Liu *et al.* [39] proposed a new routing algorithm for FANETs that utilizes Q-learning. The algorithm dynamically adjusts the Q-learning parameters, such as the learning rate and discount rate, to account for the unstable nature of the network. Each link is assigned a different learning rate and discount rate. However, this work does not analyze their protocol for network lifetime nor optimize it. Network lifetime is an important parameter to be optimized for SO application. L. Antonio *et al.* [40] devised a routing algorithm that factors in the channel condition to calculate the Q-learning

TABLE I: Related Works

Protocol	Type	Simulation Parameters	Simulation Tool	Compared With	Results
OLSR_PMD [28]	Topological	• Number of nodes • Packet Delivery Ratio (PDR)	NS-3, MATLAB	DSDV, OLSR	• Lower end-to-end delay • Higher PDR
P-OLSR [29]	Topological	• Link-quality aging • Speed-weighted ETX Metric • Throughput • Average Outage Time	Linux Containers, iperf, EMANE	OLSR	• Average outage time improved by 85% • Higher PDR due to lower channel fluctuations
Rovira-Sugranes *et al.* [25]	Topological	• End-to-end delay • Hop distance	-	Dijkstra's Shortest Path	• Lower delay, especially with larger network size • Increased network lifetime
Bhardwaj *et al.* [30]	Hybrid	• Jitter • Throughput • PDR	-	AODV	• Improved accuracy and throughput • Jitters reduced, improving effectiveness by 31%
PSO-GLFR [31]	Hybrid	• Network Bandwidth • Hop Count • Energy Consumption per packet	OMNET++	GFR, AODV	• Lower latency, packet loss and delay • Increased energy efficiency
LADTR [32]	Geographical with Store-and-Carry-Forward	• UAV Speed • Traffic • Network Bandwidth • PDR	NS-3	AODV, GPSR, Spray and Wait, Epidemic	• Higher PDR • Lower delay • Lower routing overhead
RARP [33]	Hybrid	• Data Transmission Rate • Node energy • Network Size • Hop Count	C++	AODV	• Higher data transmission success • Increased path lifetime and route setup time • Hop counts increase over 60 nodes in network
GeoUAVs [34]	Geographical	• PDR • End-to-end delay • Throughput	NS-3	AntHocNet, BeeAdHoc	• Lower end-to-end delay • Increased throughput
MAB [35]	-	• Node processing power • Residual energy • Task List	-	Shortest Path Selection	• Increased network lifetime
Bhardwaj *et al.* [36]	Hybrid	• Node mobility • Node energy consumption • Network lifetime • Transmission delay • Signal reception strength	NS-2	AODV, GPSR, DTN	• Higher PDR • Lower end-to-end delay • Lower routing overhead
BICSF [37]	Swarm Intelligence Based	• Number of UAVs • Cluster binding time • Network energy consumption • Cluster lifetime	MATLAB	Cluster oriented protocols like ACO and GWO	• Increased energy efficiency • Higher PDR
E-AntHocNEt [38]	Swarm Intelligence Based	• Quality of service • Speed • Network energy consumption • Cluster lifetime	NS-2	AntHocNet, DSR, M-DART and TORA	• Increased energy efficiency • Higher PDR
QMR [39]	Q-learning Based	• Learning Rate • Discount Factor	WSNet	QGeo	• Low delay • Low energy consumption • Higher packet arrival ratio
Q-FANET [40]	Q-learning Based	• Learning Rate • Discount Factor • Q-Value	WSNet	QGeo, Q-Noise+, and QMR	• Lower delay • Lower jitter • Minor increase in packet arrival ratio
QTAR [41]	Q-learning Based	• Path Loss • SINR Threshold • CBR rate	MATLAB	GPSR and QGeo	• Better packet delivery ratio • Lower delay • Less energy consumption

parameters. Moreover, the proposed algorithm considers a few episodes to update the parameters instead of relying solely on recent ones. Another work proposed by M. Y. Arafat *et al.* [41] uses an adaptive Q-learning algorithm in which the learning rate and reward factor are adjusted dynamically based on the network topology. There are various Q-Learning based routing protocols are proposed for FANETs. [43] [44]

Another category of routing algorithms incorporates bio-inspired algorithms. Bhardwaj *et al.* [36] use two bio-inspired algorithms for cluster-based UAV networks. The Chaotic Algae algorithm is used for cluster formation, which conserves energy levels at each node. For inter-cluster routing, the Dragonfly algorithm is used for the election of the CH. This algorithm is also useful in supporting the transmission between the clusters in terms of routing and selecting the next optimal node. Bio-Inspired Clustering Scheme for FANETs [37] is a hybrid mechanism of Glowwarm Swarm Optimization and Krill Herd. It consists of three phases - energy-aware cluster formation and cluster head election, UAV motion aware cluster management, and cluster maintenance. H. Wu *et al.* [45] proposed a cooperative clustering scheme for UAVs that offloads and exploits diversity gain to improve coverage. Khan *et al.* [38] present a novel routing protocol that uses a modified ant colony optimization algorithm. This algorithm introduced an energy stabilizing parameter, which leads to improved energy

efficiency and overall network performance. The protocols and their details are summarized in Table 1.

III. Network Model

This work considers a cluster-based UAV system, consisting of a Ground Control Station (GCS), Cluster Head/Member (CH/CM) nodes, each having a role as defined below.

A. Nodes

1) Ground Control Station (GCS): This is the base station, which receives data packets from all other nodes. It does not have any power or computational constraints with no mobility. Based on the information received about the location of survivors or damage incurred, appropriate action is taken here.

2) Cluster Head (CH): These are the core nodes of the network, i.e., they receive the packets from the cluster members and forward them towards the GCS. These UAVs have low mobility (quasi-static), and enhanced computational power and energy characteristics. This work uses EFTA [46] to place these nodes.

3) Cluster Member (CM): These are highly mobile nodes and are allotted a certain area for scanning. On capturing important information (such as the location of survivors), CMs pass this information to the CH. In case multiple CHs are available to a CM, the best one is selected as explained in Section IV. In this paper, CMs are placed randomly in the area of scanning.

B. Placement of Nodes: EFTA

An efficient method for node placement is essential to maximize the coverage and minimize the power requirement [47]. This work uses an Energy-efficient, Fault Tolerant and Area-optimized placement scheme (EFTA) [46] to place the CHs. EFTA uses the Multi-Objective Cuckoo Search Algorithm (MO-CSA) [48] to determine the placement of nodes while maximizing area coverage and fault tolerance and simultaneously minimizing power consumption. The reader is specifically refered to Eq. (2) on Page-3 of [46] for the area maximization problem definition's details, Eq. (4) on Page-3 of [46] for the Nodal Power Consumption (NPC) optimization problem and Eq. (6) on Page-6 of [46] for the Fault Tolerance Index (FTI) optimization problem. The placement algorithm can be found on Page-4, Section IV-D (Problem Formulation) of [46].

The optimization problem solved in CH placement is:

$$\begin{aligned} &\text{Maximize } \{\textit{Area, FTI}\} \text{ and Minimize } \{\textit{NPC}\} \\ &\textit{such that} \\ &\textit{Every CH has at least one transmission path} \\ &\textit{available to send information to the GCS} \end{aligned} \tag{1}$$

where FTI refers to the Fault Tolerance Index, NPC refers to the total Nodal Power Consumption, and Area is the area covered by the CHs. FTI is calculated using the average number of connections per CH. A general representation of the network layout is given in Fig. 1. As shown, the GCS is located at one corner of the rectangular area. The UAVs with a purple background are CHs. The CMs are spread throughout the rectangular area. The red dotted circle represents the communication range of each CH. Note that the network setup phase for this system model has been summarized in Algorithm 1.

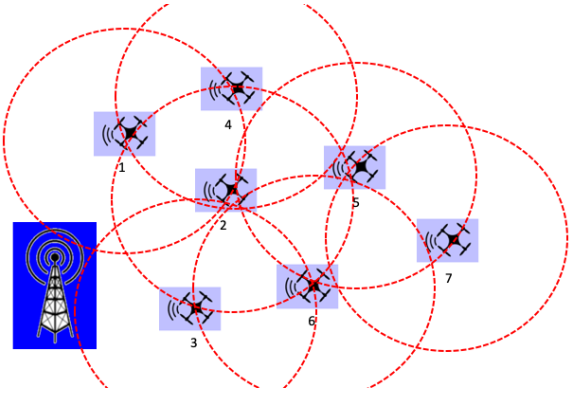


Fig. 1: Sample Network Layout

C. Flow of Information

CMs scan their respective areas and pass on important information to the CHs. CHs then process the information received and forward it to the GCS on their selected optimal routing path.

IV. MOBMDP : Routing Mechanism

The routing mechanism proposed by this work consists of two parts - CM to CH routing, and Inter-CH routing. They are explained below:

1) **CM to CH Routing**: This deals with the selection of the optimum CH for a CM at a given location. The following equation is used for optimal CH selection:

$$\text{Maximize } \frac{\log_{10}(E_i \times 10^6)}{D_i} \tag{2}$$

where E_i and D_i are the residual energy of CH_i and distance of the CM from CH_i respectively. Clearly, the value in Eq. (2) is directly proportional to a given CH's residual energy and inversely proportional to its distance from a given CM. Hence, a given CM chooses the CH with the best balance of residual energy and distance to maximize both the lifetime and efficiency of the network to avoid overwhelming and draining farther CHs with higher residual energy. If the CH is excessively loaded

Algorithm 1. Network Setup Phase

1: Place GCS at $(1, 1)$
2: Place CH according to EFTA solution
3: Initialise CH nodes: $Q_{S_i}^{S_j} = \tau_{S_i}^{S_j} = 0$, $\chi_i^j = \frac{d_i^j}{c}$, $\eta_i = CH_{int,i}$ and $\psi_i^j = \frac{\eta_j}{d_i^j}$
4: Assign CM search cells in grid

TABLE II: Notation Summary: MOBMDP

Symbol	Parameter Represented
CH_i	CH in network with index i
η	Remaining energy of minimum energy node (REMEN)
η_i^j	CH_i's Path REMEN for path through CH_j
η_i	Nodal REMEN of CH_i
E_i	Residual Energy of CH_i
D_i	Distance of CM from CH_i
$\zeta_{i,j}$	Expected Delay of Connection between CH_i and CH_j
$t_{i,j}$	Delay observed in most recent transmission between CH_i and CH_j
χ_i^j	CH_i's Expected Path Delay for path through CH_j
χ_i	CH_i's Expected Path Delay for optimal path to GCS
ψ_i^j	CH_i's Power Distance (PD) Ratio for path through CH_j
S_t, S_{t+1}	Current and next states of agent
$P[S_j\|Si]$	State transition probability from state Si to Sj
$a(i)$	Set of allowed actions in state i
$\tau_{S_i}^{S_j}$	Total Expected Return (TER) of changing state S_i to S_j
γ	Discount factor
π	Policy
$Q_{S_i}^{S_j}$	Q-value of changing state S_i to S_j
l	Learning Rate
R_0	UAV CH communication range
$CH_{int,i}$	Initial Power of CH_i
d_i^j	Distance travelled by packet from CH_i to GCS using CH_j as forwarding node
α	Delay weighted constant
c	Speed of light

by forwarding requests from many CMs then the residual energy of the CH will reduce; hence, reducing its suitability for data forwarding in subsequent data transmissions. Thus, CH selection for a CM is about balancing the trade-off between residual energy and distance.

2) **Inter-CH Routing**: This deals with the flow of data to the GCS through the CHs in the network. For this phase of routing, a novel Multi-Objective Markov Decision Process (MOBMDP) based proactive routing algorithm is proposed. MOBMDP uses Q-learning to evaluate paths on the basis of three parameters - Expected Delay of the path (ED), Remaining Energy of the Minimum Energy Node (REMEN) in the path, and a Power Distance ratio (PD). Section IV-A discusses these parameters in detail. The notations used in this work are summarized in Table II.

A. Parameters Used by Routing Mechanism

This section describes the parameters used in this work and their calculation. They are explained below:

1) Remaining Energy of the Minimum Energy Node in Optimum Path (REMEN): Due to energy constraints in UAV networks, energy consumption is an essential factor while deciding the routing path. For a network to work efficiently and maximize its lifetime, it is crucial to ensure that it consumes energy uniformly over all its nodes. Therefore, as a measure of energy distribution, this work proposes the usage of two parameters :

1) **Path REMEN** : This is the remaining energy of the minimum energy node in a specific path starting at a CH and ending at the GCS. For a path starting at CH_i and using CH_j as its forwarding node, it is denoted by η_i^j.
2) **Nodal REMEN** : This is the remaining energy of the minimum energy node in the *current optimal routing path of a CH to the GCS* in the network. For CH_i, this parameter is denoted by η_i.

The method of calculating *Nodal* and *Path REMEN* has been illustrated with an example below. Note that this example uses Fig. 1 as a reference. As the GCS is considered to have infinite energy, the REMEN value of nodes forwarding packets directly to the GCS is their residual energy. Hence, the *nodal REMEN* of CH1, CH2 and CH3 are their residual energies. Now, consider CH4, which is in the range of both CH2 and CH1 but not the GCS. Therefore, CH4 has two paths to the GCS available, namely, through CH1 and CH2. Before assigning a nodal REMEN value to CH4, its optimal forwarding node must be determined. However, determining the optimal forwarding node of a CH requires the REMEN value of the paths available to it. In the case of CH4, there are two paths available, and so, two *Path REMEN* values need to be calculated for the paths through CH1 (denoted by η_4^1) and CH2 (denoted by η_4^2). Consider the routing path through CH1. Its corresponding Path REMEN is calculated by comparing the residual energy of CH4 with the REMEN value of CH1. η_4^1 is assigned the lower of the two above values, i.e., $\eta_4^1 = min(E_4, \eta_1)$. η_4^2 is calculated in a similar manner. As explained in section IV-C, these values are used to determine the optimal forwarding node, and CH4's *Nodal REMEN* value is assigned accordingly. In other words, if CH1 is chosen as the optimal forwarding node for CH4, $\eta_4 = \eta_4^1$. Similarly, if CH2 is chosen as the optimal forwarding node for CH4, $\eta_4 = \eta_4^2$. The remaining CHs calculate their *Path* and *Nodal REMEN* values in a similar fashion. For inter-CH communication, every CH transmits information to its neighboring CH. Based on that information, every CH calculates its *Path REMEN* and *Nodal REMEN*. The details about the periodic exchange are described in section IV-B.

As REMEN values are maximized, paths with low energy nodes are avoided, and there is uniform energy consumption, allowing a greater network lifetime.

2) Expected Delay of Path (ED): ED is an estimate of the real-time delay expected in a transmission and is periodically updated along a given path. For time-sensitive applications such as SO, minimizing end-to-end delay is extremely important. This is the motivation behind using ED as one of the routing parameters.

Consider two CHs in a given network - CH_i and CH_j, where CH_i and CH_j are neighbours, such that CH_j is closer to the GCS than CH_i. The ED of a path starting at CH_i and ending at the GCS using CH_j as the forwarding node is denoted by χ_i^j. It is calculated using two parameters:

1) ED of the path from CH_j to the GCS (denoted by χ_j)
2) Predicted delay of a transmission between CH_i and CH_j (denoted by $\zeta_{i,j}$)

χ_j is available to CH_i as it is periodically transmitted by CH_j to all of its neighbours. Further, $\zeta_{i,j}$ is updated periodically using three parameters - a weighted constant α, the current value of $\zeta_{i,j}$ and the actual delay observed in the most recent transmission between CH_i and CH_j.

Mathematically, the following equation is used to update $\zeta_{i,j}$:

$$\zeta_{i,j} \leftarrow \alpha \times \zeta_{i,j} + (1-\alpha) \times t_{i,j} \tag{3}$$

where $\alpha \in (0,1)$ is a weighted constant and $t_{i,j}$ is the actual delay observed in the most recent transmission between CH_i and CH_j.

Further, the ED of a path from CH_i to the GCS using CH_j as the forwarding node is calculated using the following equation:

$$\chi_i^j = \zeta_{i,j} + \chi_j \tag{4}$$

Note that χ_j is the expected path delay from CH_j to the GCS along CH_j's last calculated optimal path.

Refer to Fig. 2 for an example of the proposed method to calculate expected path delay. Since CH1 and CH2 are neighbours of the GCS, they will send any information directed to them to the GCS directly. Therefore, $\chi_1 = \zeta_{1,0}$ and $\chi_2 = \zeta_{2,0}$. CH3, however, has two available forwarding nodes : CH1 and CH2. Hence, using Eq. (4), the following path delays are calculated through the available paths :

1) Through CH1 : $\chi_3^1 = \chi_1 + \zeta_{3,1}$
2) Through CH2 : $\chi_3^2 = \chi_2 + \zeta_{3,2}$

χ_3^1 and χ_3^2 are then used to determine the optimal forwarding node for CH3 as explained in section IV-C. Finally, χ_3 is assigned the value χ_3^1 or χ_3^2 depending on whether CH1 or CH2 is determined to be the optimal forwarding node for CH3 at a given point of time. The expected path delays of CH4 and CH5 are calculated in a similar fashion.

3) Power Distance Ratio (PD): While REMEN acts as a good measure for energy distribution, its value does not reflect the energy efficiency of a routing path. While maximizing network lifetime is important, doing so can lead to the network taking less efficient routes and thus, consuming more energy per transmission. However, in SO, it is important to balance maximizing network lifetime and minimizing energy consumption per transmission. This work proposes a novel Power Distance ratio (PD) metric to maintain this balance. PD is defined as the ratio of REMEN to the total distance a packet has to travel to reach the GCS from a given CH. For CH_i, it can be mathematically represented as:

$$\psi_i^j = \frac{\eta_j}{d_i^j} \tag{5}$$

where ψ_i^j and d_i^j are the PD and distance associated with CH_i while using CH_j as the forwarding node, and η_j is the *Nodal REMEN* of the forwarding node CH_j.

Since the PD of a path is directly proportional to its REMEN and inversely proportional to the distance travelled by the packet (and therefore, energy consumed in data transmission), maximizing PD enables nodes to find the right balance between energy consumption and energy distribution.

B. Periodic Inter CH Data Exchange

The following information is periodically transmitted by a CH to its neighbours for inter-CH communication. Note that the periodic Inter-CH data exchange process has been summarized in Algorithm 2.

1) Nodal REMEN of CH: As mentioned in Section IV-A1, the *Nodal REMEN* of a CH is used by its neighbours to determine their *Path* and *Nodal REMENs*. Once the optimal real-time path for transmitting information from a CH to the GCS is identified, the CH compares its residual energy to the *Nodal REMEN* of its immediate forwarding node along its optimal real-time path. The lower of the two is set as the *Nodal REMEN* value of the CH itself. This updated REMEN value is periodically transmitted from CHs to their neighbours.

2) Latest Receiving Time Stamp Values of CH: Consider two nodes in the network, CH_i and CH_j. In an event-based transmission, let CH_i send a data packet to its forwarding node CH_j. Let the sending timestamp of the transmission be saved by CH_i in a variable $T1$. Once CH_j receives the data packet, it saves the receiving timestamp as $T2$. Hence, in its periodic transmissions, CH_j includes the latest receiving timestamp values associated with each of its neighbours. When CH_i receives this periodic packet, it subtracts $T1$ from the corresponding $T2$ value. The resulting value ($T2-T1$) is used to calculate the expected delay of the connection between CH_i and CH_j. This $T2-T1$ value represents the observed delay of the most recent transmission of this connection ($t_{i,j}$ in Eq. (3)).

3) ED of current optimal path: The expected delay in information transfer to the GCS, χ_j, is stored by CH_j and is included in its periodic transmissions to its neighbours. χ_j is used by CH_i to calculate χ_i^j using Eq. (4).

4) Current Energy of CH: In addition to the above quantities, CH_j includes its current energy, E_j, in its periodic transmissions. E_j is used by CMs in the vicinity of CH_j to choose their optimal forwarding CH using Eq. (2).

Complexity calculation of Periodic Inter-CH Communication: The main calculations performed in the Periodic Inter-CH Communication (Algorithm-2), which form an overhead for this algorithm, are steps 10 to 15 of Algorithm-2. These operations are performed for each of the neighbours of CH_i. The number of operations in these steps is constant; let that be k. Let there be m neighbours for CH_i. Thus, the number of operations performed for CH_i is $m \times k$, where k is a constant. Thus, the number of calculations for each CH in the network is $O(m)$. If there are n nodes in the whole network, the number of calculations in Algorithm-2 will be $O(mn)$.

C. Markov Decision Process (MDP)

As mentioned in [49], MDP is a formulation based on decision theory and discrete time Markov Process Theory. The decisions are taken on the basis of rewards and all the states are "Markov". For a state to be "Markov", the following condition needs to be satisfied:

$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, S_2, S_3....S_t] \tag{6}$$

where S_t denotes the current state of the agent and S_{t+1} denotes the next state of the agent. In other words, the future state is solely dependent on the current state. As routing decisions are primarily based on the current state of the agent, MDP is a good choice for routing protocols.

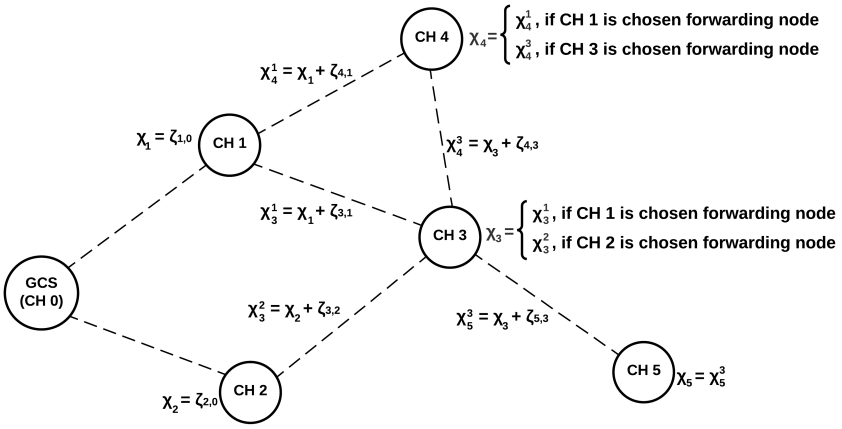


Fig. 2: Path Delay Calculation in the Network

Algorithm 2. Periodic Inter-CH Communication

Let the time period of periodic data transmission be T. Let the transmitting CH node be CH_j.
CH_j creates payload consisting of the its η_j, neighbouring CH receiving timestamp values, χ_j and E_j.
Transmission:
T seconds from the previous periodic transmission, the above payload is transmitted by CH_j to all CMs and CHs in its vicinity.
When a CM receives:
CM saves E_j. It uses this value to choose its optimal routing node using Eq. (2).
When a CH receives:
Let the receiving CH be CH_i.
if $j \in a(i)$ **then**
    1. Update CH_i's χ_j with χ_j obtained from the periodic packet
    2. Use receiving timestamp value corresponding to CH_j in the packet to calculate $t_{i,j}$. Use $t_{i,j}$ to update the value of $\zeta_{i,j}$ using Eq. (3)
    3. Update the value of Nodal REMEN of CH$_j$ (η_j)
    4. Update χ_i^j using Eq. (4)
    5. Update the value of $\tau_{S_i}^{S_j}$ using Eq. (10)
    6. Update the value of $Q_{S_i}^{S_j}$ using Eq. (12)
end if

Algorithm 3. Episodic Data Transfer from CH

Let CH i be the transmitting CH
if $((x_D^t - 1)^2 + (y_D^t - 1)^2 < R_0^2)$ **then**
    Send packet to GCS
else
    Send packet to jth CH, where j is chosen such that: Maximise $Q_{S_i}^{S_j}\ \forall\ j \in a(i)$
end if

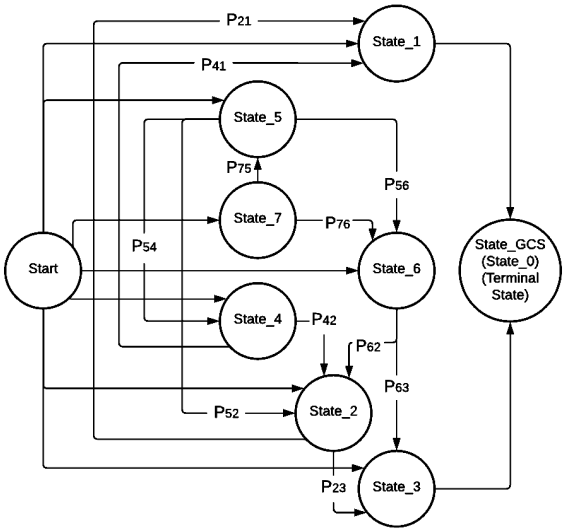


Fig. 3: Markov State Diagram

For the MDP considered in this work, the agent is a data packet, states are the different CHs that the packet may be transmitted to, and action is the transmission of a data packet from one CH to another. The CM to CH selection is not a part of this MDP, and this selection is carried out using Eq. (2). Further details of the MDP are presented below:

1) State Space (SS): This is the set of possible states in an MDP. For this work, the SS consists of all the CHs and the GCS. It is mathematically represented as below:

$$S_t \ \epsilon \ SS \ such \ that \ SS = \{S_0, S_1, S_2 .... S_k\} \tag{7}$$

where k is the number of CHs, S_t is the current state, and SS is the state space where S_0 represents the GCS, and the remaining values in SS represent the CHs in the network. Figure 3 shows the Markov State Diagram for the network layout given in Fig. 1. Since there are seven CHs in the network, eight states are present in the ensuing MDP, where S_0 corresponds to the GCS and the remaining states correspond to the CHs as labelled in Fig. 1. Note that the GCS is the

terminal state of this MDP.

2) Action Space (A): This is the set of actions that the MDP environment allows in a given state. Here, there are two types of possible actions - sending the data packet from one CH to another and from a CH to the GCS. The set of allowed actions are defined as follows:

$$\begin{aligned} &j \in a(i)\ \forall \textit{ possible } j \textit{ iff} \\ &1.\ (x_D^i - x_D^j)^2 + (y_D^i - y_D^j)^2 < R_0^2 \\ &2.\ (x_D^i - 1)^2 + (y_D^i - 1)^2 > (x_D^j - 1)^2 + (y_D^j - 1)^2 \\ &\textit{for } i \ \in \{S_1, S_2, S_3 .... S_k\} \\ &\textit{and } j \ \in \{S_0, S_1, S_2 .... S_k\} \end{aligned} \tag{8}$$

where *a* represents the action of sending a packet from CH_i to the *j*th state (CH or GCS), *k* is the number of CHs, x_D^i and y_D^i are the x and y coordinates of CH_i, R_0 represents the communication range of the CHs and (1,1) is the location of the GCS. Note that condition 1 checks whether CH_i is in range of CH_j and condition 2 checks whether CH_j (receiving CH) is closer to the GCS than CH_i (transmitting CH). Figure 3 represents the set of allowed actions. The MDP starts once the CM selects a CH. All other allowed actions are obtained using Eq. (8).

3) State Transition Probability (P): This matrix stores the probabilities of transitioning from one state to another. The state transition probability is defined using PD, as defined earlier. It is mathematically represented as follows:

$$P[S_j|S_i] = \frac{\psi_j^i}{\sum_{k=a(i)_1}^{a(i)_n} \psi_k^i} \textit{ such that } j \in a_i \tag{9}$$

where $a(i)$ is the set of allowed actions for CH_i (or S_i), *n* is the number of allowed actions for CH_i, ψ_j^i is the power distance ratio associated with sending information from CH_i through CH_j(as a forwarding node) and $P[S_j|S_i]$ is the state probability associated with sending information from CH_i to the GCS through CH_j. The state transition probabilities for the allowed actions are shown in Fig. 3.

As it rates possible actions in a state on the basis of their power distance ratio, this work uses state transition probability to simultaneously optimize energy efficiency and energy distribution in the network. Clearly, the state transition probabilities of this MDP attempt to maximize the PD of the available actions by giving additional weightage to actions with high PD values.

4) Total Expected Return Function (TER): This function is used to calculate the *total expected return* associated with a transition from a given state to a new one. Most reinforcement learning algorithms use short-term reward functions to generate information about how good a state transition is without considering future transitions and rewards. These "short-term reward" values are further used to calculate the maximum total expected return of an action using the concept of discounted rewards (long-term rewards). However, in this work, *TER directly generates the maximum total expected return associated with a state change*, thereby making decisions easier and reducing the complexity of the MDP compared to a scenario where short-term rewards are used to calculate the long-term rewards. The reason for using long-term rewards only is to focus on arriving at an optimal path. Using short-term rewards may help in choosing a node, which may reduce the delay between two individual CHs but may lead to a non-optimal routing path. TER can be mathematically represented as below:

$$\tau_{S_i}^{S_j} = \frac{\eta_j}{(\log_{10}(\chi_i^j \times 10^5))^4} \tag{10}$$

where $\tau_{S_i}^{S_j}$ is the maximum total expected return associated with sending data from CH_i to CH_j (or S_j), η_j is the energy of the minimum energy node along the optimum path through CH_j and χ_i^j is the expected time taken to send information to the GCS from CH_i along the optimal path through CH_j.

Clearly, TER is directly proportional to the REMEN and inversely proportional to the predicted path delay. Hence, using TER, this work enables CHs to make decisions that improve the energy distribution of the network while simultaneously minimizing the delay associated with data transmissions.

5) MDP Policy (π): This refers to the rule followed by the MDP to determine what action to take in each state. In this work, the MDP follows a greedy policy that tries to maximize the Q-value of the action taken at every stage. The policy can be represented as below:

$$\pi(S_i)\ s.t.\ \text{Maximize}\ Q_{S_i}^{S_j}\ \forall\ S_j, S_i \in SS,\ S_j \in a(i) \tag{11}$$

where $\pi(S_i)$ is the policy followed by the MDP (i.e., the CH chosen by it for the next hop), $Q_{S_i}^{S_j}$ is the Q-value associated with changing state S_i to S_j, $a(i)$ is the allowed set of actions from state S_i, and S_i and S_j are two states in SS. Note that this policy is only used for CHs for whom the GCS is out of range. If the GCS is in range of a CH, the CH will transmit directly to the GCS (which is the terminal state of this MDP). This policy has been summarized in Algorithm 3.

This work uses the concept of Q-learning to compare and periodically update the real-time Q-values of actions for a given state. Q-learning was first proposed by Watkins and Dayan in [42]. It was initially formulated as a means of enabling agents to learn their environments through the concepts of "rewards" and "returns". Within this work, however, agents leverage Q-learning not only to learn their environment, but also to continuously update their understanding of it and make consistently better decisions. The Q-value function $Q_{S_i}^{S_j}$ acts a measure of how "good" a given action is at a given time. It is refreshed for a given (S_j, S_i) when the associated periodic inter-CH data packet is received by CH_i. The new Q-value for that (S_j, S_i) is calculated using the Q-learning update rule given below. Since this work directly generates the maximum total expected reward of an action, the update rule (which is proved in Theorem 1) is mathematically represented as:

$$\begin{aligned} &Q_{S_i}^{S_j} \leftarrow (1-l) \times \tau_{S_i}^{S_j} + l \times Q_{S_i}^{S_j} \\ &\text{where } l = \frac{\gamma \times P[S_j|S_i]}{1 + (\gamma \times P[S_j|S_i])} \end{aligned} \tag{12}$$

where l is the learning rate of the MDP (clearly, $l \in [0, 1)$. Further, $\tau_{S_i}^{S_j}$, $P[S_j|S_i]$ and $Q_{S_j}^{S_i}$ are respectively the TER, state transition probability and Q-values associated with a transition from state S_i to S_j. Additionally, γ is a weighted constant that

represents the importance given to the current and expected return of a given action from a given state. Therefore, this MDP directs a given CH to periodically update its routing information and send information through the node with the *highest* Q-value amongst its neighbors. *The neighbor with the highest Q-value is chosen as an optimal forwarding node by the given CH.*

Theorem 1. *Consider an MDP with state space SS and action space A. Let it have a state-action function $Q(s,a)$ where $s \in SS$ and $a \in A$. Consider a function TER that stores the maximum total expected reward associated with a state change. Hence, the Q-learning update equation can be represented as below:*

$$Q(s,a) = (l) \times Q(s,a) + (1-l) \times TER(s,a)$$

where l is the learning rate and $l \in$ [0,1].

Proof: The optimal Bellman Equation for calculating the Q-value of an action as mentioned in [42] is:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} (P(s,a,s') max_{a'} Q(s',a') \quad (13)$$

where s and a are the current state and action taken in that state respectively. Similarly, s' and a' represent the next state and the next action taken respectively. $R(s,a)$ is the return or reward in that state, $P(s,a,s')$ is the probability of moving from one state to another and $Q(s,a)$ is the Q-value associated with the quality of the action a.
The agent calculates Q-values associated with an action and a starting state. However, changes in the environment may cause this value to change over time. If $Q_{w-1}(s,a)$ denotes the presently saved Q-value of the system and $Q^*_{w-1}(s,a)$ denotes the current optimal Q-value, then the Bellman error [42] is represented as:

$$error = Q^*_{w-1}(s,a) - Q_{w-1}(s,a) \quad (14)$$

The aim of Q-learning is for the agent to adapt to this error and constantly update its Q-values. This adaptation is made using:

$$Q_w(s,a) = Q_{w-1}(s,a) + \alpha * error \quad (15)$$

where w is the current iteration and α is the learning rate. On substituting Eq. (14), this equation becomes :

$$Q_w(s,a) = Q_{w-1}(s,a) + \alpha * (Q^*_{w-1}(s,a) - Q_{w-1}(s,a)) \quad (16)$$

which, on rearranging and substituting Eq. (13), becomes :

$$Q_w(s,a) = (1-\alpha) * Q_{w-1}(s,a) + \alpha * \{R(s,a) + \gamma \sum_{s'} (P(s,a,s') max_{a'} Q(s',a')\} \quad (17)$$

However, as mentioned in [42], the expression $\{R(s,a) + \gamma \sum_{s'} (P(s,a,s') max_{a'} Q(s',a')\}$ represents the *maximum expected discounted reward* of taking an action a in state s. In other words, the expression $\{R(s,a) + \gamma \sum_{s'} (P(s,a,s') max_{a'} Q(s',a')\}$ is essentially the maximum expected return of the action a in state s. As TER(s,a) represents the above quantity, Eq. (17) can be written as :

$$Q_w(s,a) = (1-\alpha) * Q_{w-1}(s,a) + (\alpha) * TER(s,a) \quad (18)$$

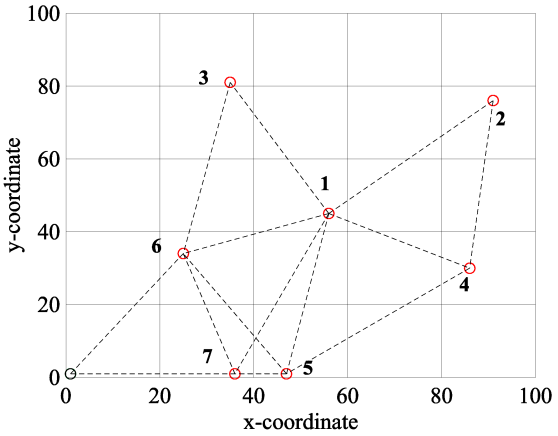


Fig. 4: Network Topology for the Cluster Heads

Let $\alpha \in$ [0,1] and consider a variable $l = 1 - \alpha$. Therefore, on substituting $\alpha = 1 - l$, Eq. (18) becomes :

$$Q_w(s,a) = (l) * Q_{w-1}(s,a) + (1-l) * TER(s,a) \quad (19)$$

thus proving the theorem. Note that as $\alpha \in$ [0,1], $l \in$ [0,1] in this equation too. ■

V. Performance Evaluation

This section presents the results of the proposed routing algorithm MOBMDP. All the simulations were carried out in MATLAB 2020b. The search area is modeled as a grid with cells of dimension 100m x 100m. The routing scheme proposed in this work is compared to four state-of-the-art routing schemes - RARP (Robust and Reliable Routing Protocol) [33], P-OLSR (Predictive Optimised Link State Routing Protocol) [29], PRED (Predictive Routing) [25] and MAB (Multi-Armed Bandit-Based Routing Protocol) [35]. All the simulations consider seven CHs (except for scalability analysis described in the subsection V-G). and their placement in the network is shown in Fig. 4. Further, non-deterministic delays (which include access delay and other MAC layer related delays) are modeled as a Gaussian variable as in [50]. Note that these simulations assume that the UAVs are equipped with the RFD900x Radio Modem [51], which has a transmitting power of 1W and and is running at a baudrate of 1800 bytes per second (or 14,400 bits per second). Note that this radio was specifically chosen as it is capable of radio transmissions upto a range of 80 km. It is assumed to operate at a frequency of 928 MHz as mentioned in the datasheet. It is very important to understand which components consume more energy for energy efficiency. [52]

In our simulations, the following transmissions are accounted for :

1) Periodic Transmissions : These refer to the periodic transmission elaborated in Section IV-B. They have a packet size of 256 bytes and therefore, consume (1*256/1800) J or approximately 142mJ of energy per radio transmission. The bandwidth of the radio used in this work is RFD900x Radio Modem is typically 500 kbits per second [51]. Even if the update of network parameters is performed every

second, the control packets use 2048 bits of this 500 kbits which accounts to 0.4% of the bandwidth which is very insignificant. Thus, the overhead for the periodic transmissions is not very significant.

2) Event Based Transmissions : These are event based transmissions from the CMs to the GCS. For the purpose of simulation, we are assuming the transmission of a 1920x1080p image in a 24-bit RGB format, the size of which is 6220800 bytes. Therefore, event based transmissions consume (1*6220800/1800) J, which is 3456 J per transmission.

Also, as mentioned in [53], when hovering at a constant height, UAV energy consumption varies linearly with time. Also, Since the CHs in our model are quasi-static and can therefore, be assumed to hover at a constant height, the power consumption of the UAVs due to its motors and accessories is constant. For the purpose of simulations, UAV CHs are assumed to have an idle hovering flight time of 8 hours and initial energy of 3.5 MJ. Therefore, the power consumption by the UAV motors and accessories is 3.5MJ/(8x3600)s, which is approximately 121 J/s.

Additionally, note that packets are generated by CMs using Gaussian variables of mean 1 and standard deviation 1.

Simulation parameters for the routing mechanism are further summarized in Table III. It must noted here that all simulated algorithms use the same mechanism for CM to CH routing. They differ only in inter-CH routing where the respective algorithm is followed. The forthcoming sections provide an analysis of the network performance metrics.

A. Network Lifetime

Network Lifetime is defined as the time taken for any CH in the network to run out of energy. In this simulation, each CH has been allotted an initial energy of 3.5 *MJ* for transmission. Once a CH runs out of energy, it cannot be used for routing, and the routing protocols consider other available paths. Figure 5 shows the network lifetime of the protocols. Clearly, MOBMDP has the best network lifetime. This can be attributed to the $REMEN$ and PD parameters used in this work, which evenly distribute the energy consumption in the network. Further, since RARP and MAB also attempt to optimize energy consumption, they perform second and third best respectively. Neither P-OLSR nor PRED consider energy as an optimizing parameter, owing to which they perform fourth and fifth best.

TABLE III: Routing Algorithm Simulation Parameters

Parameter	Value
Number of Packets Simulated	2600
Non-deterministic delay standard deviation	93×10^{-7}
γ	0.2
α	0.9
CH_{int} *(MJ)*	3.5
P_E *(Burst Error Probability)*	0.02777
UAV Motor and Accessory Power Consumption (W)	121
Energy Consumed per Periodic Radio Transmission (mJ)	142
Energy Consumed per Event Based Radio Transmission (J)	3456
Operating Frequency (MHz)	928

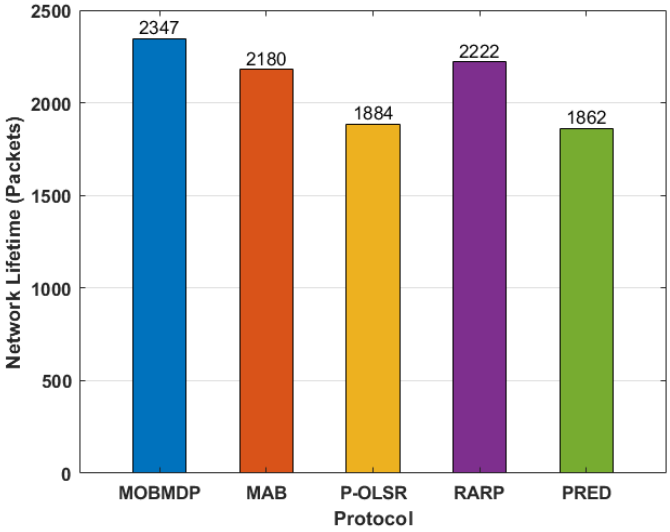


Fig. 5: Network Lifetime

B. Packet Delivery Ratio

Packet Delivery Ratio (PDR) is defined as the ratio of successful transmissions to the total number of transmissions from CM to GCS. To model the burst errors (errors due to random changes in the channel), this work uses the Burst Error model proposed in [54] and [55]. This model defines a two-state Markov chain with probability P_E (named burst error probability), denoting the probability of the transmission to go from a "good" (or successful) state to a "bad" (or unsuccessful) one. Apart from burst errors, packets are considered dropped if no paths to the GCS are available. Figure 6 shows a comparative analysis of the PDRs. Note that this figure shows the variation of the cumulative PDR at every point on the X-axis. For example, the PDR value calculated at packet number 2000 is that of all packets transmitted between the first recorded packet to packet number 2000. Once again, MOBMDP outperforms the other routing algorithms. This can be attributed to MOBMDP having a higher network lifetime than the competing algorithms. This ensures that UAV nodes take longer to run out of energy and packets dropped due to the unavailability of routing paths (arising from UAV nodes running out of energy) are minimized.

C. Energy Efficiency

Energy efficiency is the total energy expended by the CHs per successful transmission. This can be calculated using the PDR as below:

$$\text{Energy Efficiency} = \frac{\text{Total energy expended by the CHs}}{\text{PDR}} \tag{20}$$

The comparative analysis of the energy efficiency of MOBMDP and the other routing algorithms has been presented in Fig. 7. As mentioned in section V-B, a higher network lifetime contributes to an increase in the number of successful transmissions. Further, MOBMDP uses a novel metric, "PD" (Power Distance ratio), which ensures that the UAV nodes balance maximizing network lifetime and minimizing energy

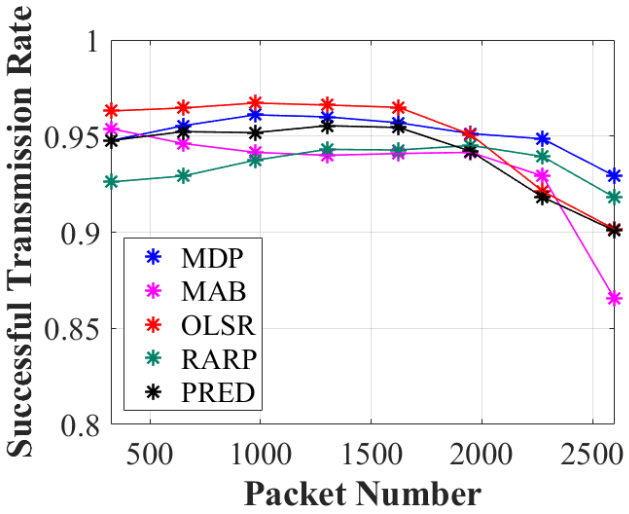


Fig. 6: Packet Delivery Ratio

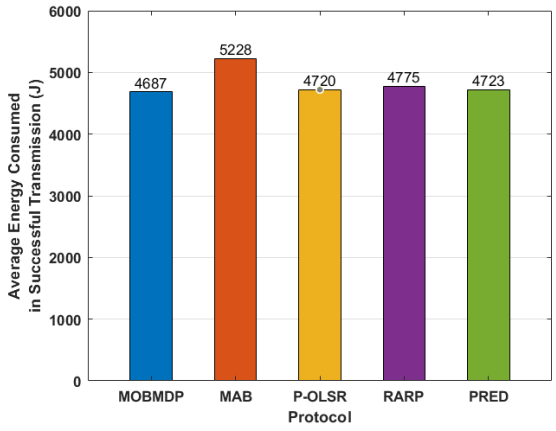


Fig. 7: Average Energy Expended in Successful Transmissions

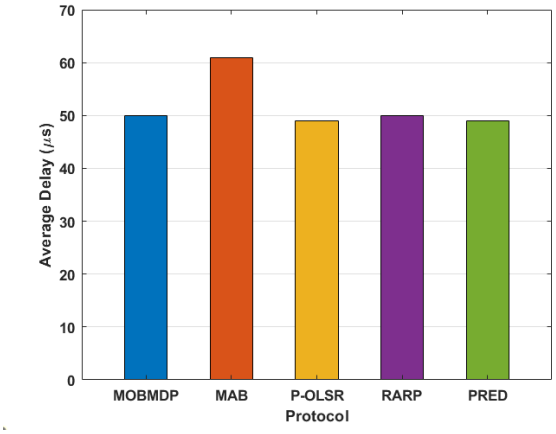


Fig. 8: Average Delay in Successful Transmissions

expended in transmitting data. Hence, MOBMDP expends less energy on transmissions and has more successful transmissions. Therefore, MOBMDP outperforms the remaining routing algorithms in energy efficiency.

D. Average Delay in Data Transmission

Average Delay is calculated by considering the average delay for successful transmissions for each protocol. In other words, delay values were only considered where the packet was successfully delivered to the GCS. The comparative analysis of the average delays of the five algorithms is depicted in Fig. 8. MOBMDP, MAB, P-OLSR, RARP, and PRED all perform comparably with an average difference of 1 μs.

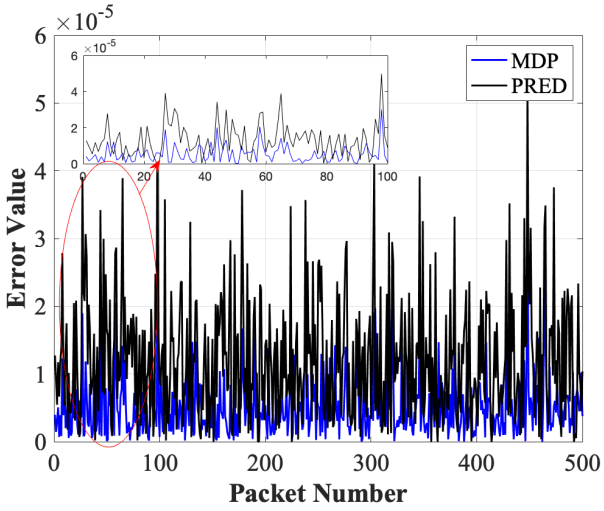


Fig. 9: Error in Predicted Delay

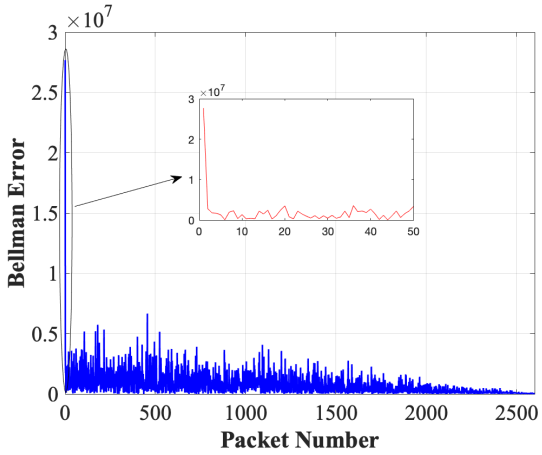


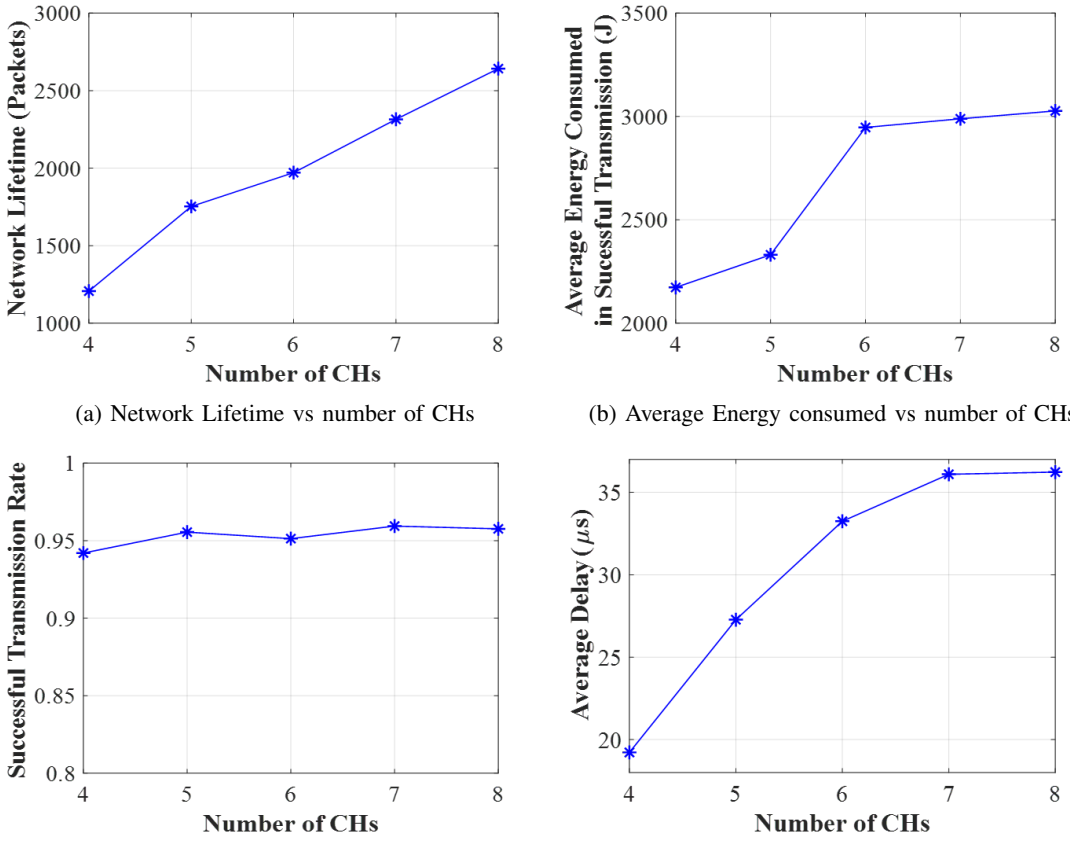
Fig. 10: MOBMDP Convergence

E. Error in Delay Estimation

Among the five algorithms compared, MOBMDP and PRED attempt to predict the delay associated with all possible paths and use that as a parameter to decide their forwarding nodes. Therefore, Fig. 9 shows a comparative analysis of the error in predicted delay of MOBMDP and PRED. Error is defined as $Error = |Delay_{Predicted} - Delay_{actual}|$ MOBMDP gives a more accurate delay prediction than PRED. In fact, it reduces the error in the delay prediction by a factor of 10. This can be attributed to the RL-based mechanism proposed in MOBMDP using real-time delay information to update its prediction in contrast to PRED, which simply makes estimations based on the distance between the UAV nodes.

F. Convergence Analysis

Figure 10 depicts the absolute Bellman error of CH1 in MOBMDP for every packet transmission. In Q-learning,



(a) Network Lifetime vs number of CHs

(b) Average Energy consumed vs number of CHs

(c) Packet Delivery Ratio vs number of CHs

(d) Average Delay vs number of CHs

Fig. 11: Analysis of various performance parameters with varying network size (number of CHs)

Bellman error is defined as the change in the Q-value of an agent in the next observed state [42]. Clearly, after the first packet, there is a considerable change in the Q-value of CH1. However, the Bellman error remains relatively low and stable until the 2600^{th} packet. Therefore, it can be concluded that MOBMDP converges. Further, since the Bellman error stabilizes almost instantly, it can be concluded that MOBMDP shows fast convergence.

G. Scalability Analysis

The performance of MOBMDP is analysed for various parameters with varying number of CHs. Figure 11 presents the results of this analysis for Network lifetime in Fig. 11a, Average energy consumed in Fig. 11b, PDR in Fig. 11c and average delay in Fig. 11d.

It can be seen from Fig. 11a that the network lifetime increases with the number of CHs. The reason for this trend can be attributed to the fact that as the network size increases, more CHs are available for forwarding the packets in the network. Thus, it takes longer for the first CH to run out of energy, thereby increasing the network lifetime.

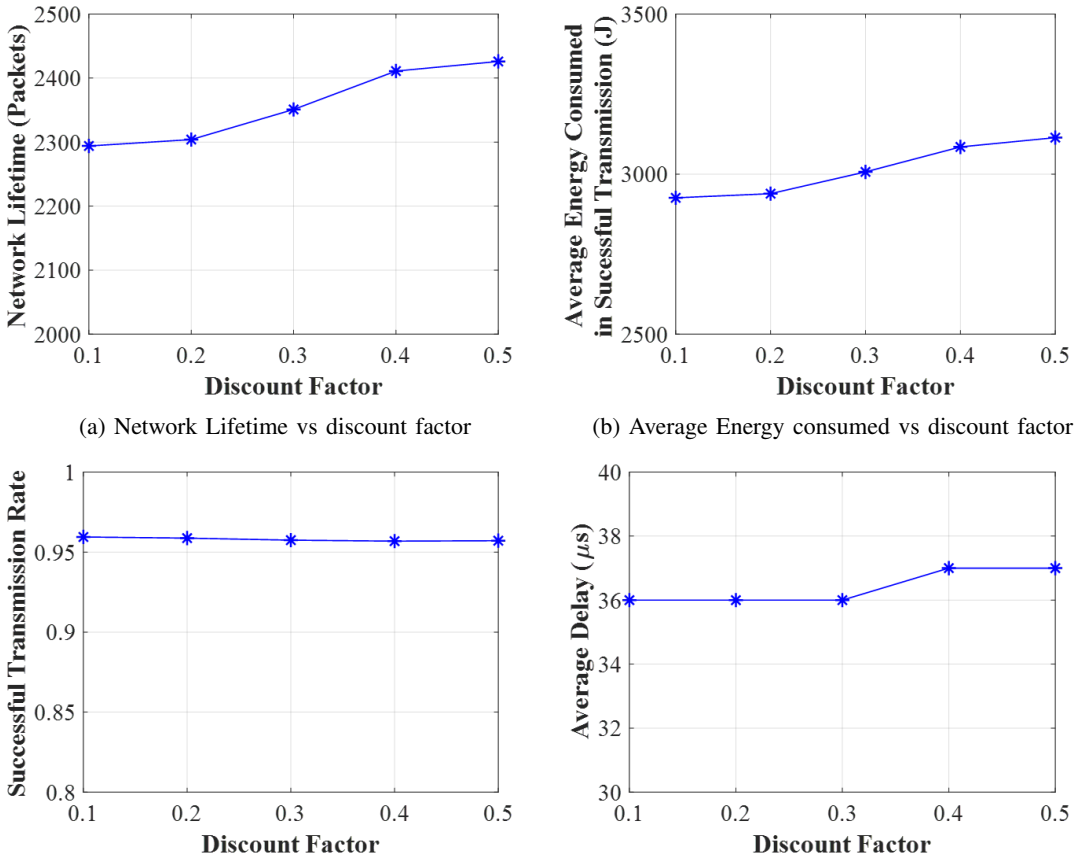
Figure 11b shows that the network's average energy consumed (in Joules) increases with network size. This is because more CHs are present in the network, increasing the total energy consumed by the mechanical parts of the CH UAVs (like motors). Further, as the network lifetime increases with network size, data packets transmitted increases with network size. This, in turn, further increases the energy consumed as network size increases. Note that only CHs were considered for this energy calculation because this paper focuses on the routing algorithm of the CHs.

The Packet Delivery Ratio (PDR) (or successful transmission rate) variation with network size is shown in 11c. This graph shows the PDR remains greater than 95% for all network sizes considered, thereby showing the efficiency of MOBMDP. This shows that network size does not affect the PDR and MOBMDP ensures a high packet delivery ratio. Further, MOBMDP's PDR is higher than the competing algorithms, as shown in Fig. 6.

Figure 11d shows the variation of average transmission delay with network size. This figure shows that the average delay increases with network size as the CHs farther from the GCS take more hops to reach the GCS.

H. Effect of Discount factor on the Network Performance

The effect of discount factor γ on various network performance parameters is shown in Fig. 12. In this experiment, the number of CHs is kept constant at seven. The value of γ varies from 0.1 to 0.5 in the steps of 0.1. Figure 12a shows the variation of network lifetime with varying values of γ. A small increase in network lifetime value is observed with an increase in γ. The effect of γ on average energy consumed by the CHs is shown in Fig. 12b. The variation of energy consumption is very slight from 2926.16 J for $\gamma = 0.1$ compared to 3113.61 J which is an increase of 6.4%. This shows that the effect of γ on average energy consumption is fairly minimal. Variation



(a) Network Lifetime vs discount factor

(b) Average Energy consumed vs discount factor

(c) Packet Delivery Ratio vs discount factor

(d) Average Delay vs discount factor

Fig. 12: Analysis of various performance parameters with varying discount factors γ for N=7 (CHs)

of PDR with γ is shown in Fig. 12c. This shows change in γ does not have an appreciable effect on PDR. Figure 12d shows the effect of γ on the average (transmission) delay of the CHs. Again the increase in average delay is fairly minimal, i.e., 1.85% from $\gamma = 0.1$ to $\gamma = 0.5$.

VI. Conclusion

This work presents MOBMDP (Multi-Objective Markov Decision Process Based Routing), a novel Q-learning and MDP-based routing algorithm explicitly designed for multi-UAV system facilitated search operations. This algorithm outperforms four state-of-the-art algorithms in terms of network lifetime, energy efficiency and packet delivery ratio while maintaining similar performance in data transmission delay . In future works, the aim will be to apply this routing protocol in a novel end-to-end secure UAV network framework on a hardware testbed.

Acknowledgment

This work is supported by BITS-Pilani through the Additional Competitive Research Grant under Project Grant Reference No. PLN/AD/2020-21/2.

References

[1] C. Guo, Z. Guo, Q. Zhang, and W. Zhu, "A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 5, pp. 834–848, 2004.

[2] A. Saci, A. Al-Dweik, and A. Shami, "Direct Data Detection of OFDM Signals Over Wireless Channels," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12432–12448, 2020.

[3] D. M. Manias and A. Shami, "The Need for Advanced Intelligence in NFV Management and Orchestration," *IEEE Network*, vol. 35, no. 1, pp. 365–371, 2021.

[4] E. Uchiteleva, A. R. Hussein, and A. Shami, "Lightweight Dynamic Group Rekeying for Low-Power Wireless Networks in IIoT," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4972–4986, 2020.

[5] A. Moubayed, A. Shami, P. Heidari, A. Larabi, and R. Brunner, "Edge-Enabled V2X Service Placement for Intelligent Transportation Systems," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1380–1392, 2021.

[6] 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 Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[7] D. Bein, W. Bein, A. Karki, and B. B. Madan, "Optimizing Border Patrol Operations Using Unmanned Aerial Vehicles," in *2015 12th International Conference on Information Technology - New Generations*, pp. 479–484, 2015.

[8] V. R. Khare, F. Z. Wang, S. Wu, Y. Deng, and C. Thompson, "Ad-hoc network of unmanned aerial vehicle swarms for search destroy tasks," in *2008 4th International IEEE Conference Intelligent Systems*, vol. 1, pp. 6–65–6–72, 2008.

[9] J.-H. Park, S.-C. Choi, I.-Y. Ahn, and J. Kim, "Multiple UAVs-based Surveillance and Reconnaissance System Utilizing IoT Platform," in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–3, 2019.

[10] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the Sky: Leveraging UAVs for Disaster Management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[11] N. Falco, H. Wainwright, C. Ulrich, B. Dafflon, S. S. Hubbard, M. Williamson, J. D. Cothren, R. G. Ham, J. A. McEntire, and M. McEntire, "Remote Sensing to Uav-Based Digital Farmland," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 5936–5939, 2018.

[12] H. El-Sayed, M. Chaqfa, S. Zeadally, and D. Puthal, "A Traffic-Aware

Approach for Enabling Unmanned Aerial Vehicles (UAVs) in Smart City Scenarios," *IEEE Access*, vol. 7, pp. 86297–86305, 2019.

[13] T. Yu, X. Wang, and A. Shami, "UAV-Enabled Spatial Data Sampling in Large-Scale IoT Systems Using Denoising Autoencoder Neural Network," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1856–1865, 2019.

[14] M. Samir, D. Ebrahimi, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Leveraging UAVs for Coverage in Cell-Free Vehicular Networks: A Deep Reinforcement Learning Approach," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2835–2847, 2021.

[15] A. H. Arani, M. Mahdi Azari, W. Melek, and S. Safavi-Naeini, "Learning in the Sky: Towards Efficient 3D Placement of UAVs," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–7, 2020.

[16] A. Al-Hilo, M. Samir, C. Assi, S. Sharafeddine, and D. Ebrahimi, "UAV-Assisted Content Delivery in Intelligent Transportation Systems-Joint Trajectory Planning and Cache Management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5155–5167, 2021.

[17] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, "Age of Information Aware Trajectory Planning of UAVs in Intelligent Transportation Systems: A Deep Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12382–12395, 2020.

[18] G. Varela, P. Caamaño, F. Orjales, Deibe, F. López-Peña, and R. J. Duro, "Swarm intelligence based approach for real time UAV team coordination in search operations," in *2011 Third World Congress on Nature and Biologically Inspired Computing*, pp. 365–370, 2011.

[19] M. Erdelj and E. Natalizio, "UAV-assisted disaster management: Applications and open issues," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1–5, 2016.

[20] D. Erdos, A. Erdos, and S. E. Watkins, "An experimental UAV system for search and rescue challenge," *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 5, pp. 32–37, 2013.

[21] Y. Bao, X. Fu, and X. Gao, "Path planning for reconnaissance UAV based on Particle Swarm Optimization," in *2010 Second International Conference on Computational Intelligence and Natural Computing*, vol. 2, pp. 28–32, 2010.

[22] D. Avola, G. L. Foresti, N. Martinel, C. Micheloni, D. Pannone, and C. Piciarelli, "Aerial video surveillance system for small-scale UAV environment monitoring," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, 2017.

[23] X. Chen, J. Tang, and S. Lao, "Review of Unmanned Aerial Vehicle Swarm Communication Architectures and Routing Protocols," *Applied Sciences*, vol. 10, no. 10, p. 3661, 2020.

[24] D. Shumeye Lakew, U. Sa'ad, N.-N. Dao, W. Na, and S. Cho, "Routing in Flying Ad Hoc Networks: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1071–1120, 2020.

[25] A. Rovira-Sugranes and A. Razi, "Predictive Routing for Dynamic UAV Networks," in *2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pp. 43–47, 2017.

[26] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," in *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.*, pp. 62–68, 2001.

[27] C. Perkins and E. Royer, "Ad-hoc On-Demand Distance Vector Routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, 1999.

[28] M. Song, J. Liu, and S. Yang, "A Mobility Prediction and Delay Prediction Routing Protocol for UAV Networks," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, 2018.

[29] S. Rosati, K. Krużelecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic Routing for Flying Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1690–1700, 2016.

[30] V. Bhardwaj and N. Kaur, "Optimized Route Discovery and Node Registration for FANET," in *Evolving Technologies for Computing, Communication and Smart World*, (Singapore), pp. 223–237, Springer Singapore, 2021.

[31] F. Wang, Z. Chen, J. Zhang, C. Zhou, and W. Yue, "Greedy Forwarding and Limited Flooding based Routing Protocol for UAV Flying Ad-Hoc networks," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 1–4, 2019.

[32] M. Y. Arafat and S. Moh, "Location-Aided Delay Tolerant Routing Protocol in UAV Networks for Post-Disaster Operation," *IEEE Access*, vol. 6, pp. 59891–59906, 2018.

[33] G. Gankhuyag, A. P. Shrestha, and S.-J. Yoo, "Robust and Reliable Predictive Routing Strategy for Flying Ad-Hoc Networks," *IEEE Access*, vol. 5, pp. 643–654, 2017.

[34] F. Z. Bousbaa, C. A. Kerrache, Z. Mahi, A. E. K. Tahari, N. Lagraa, and M. B. Yagoubi, "GeoUAVs: A new geocast routing protocol for fleet of UAVs," *Computer Communications*, vol. 149, pp. 259–269, 2020.

[35] A. Mukherjee, S. Misra, V. S. P. Chandra, and M. S. Obaidat, "Resource-Optimized Multiarmed Bandit-Based Offload Path Selection in Edge UAV Swarms," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4889–4896, 2019.

[36] V. Bhardwaj and N. Kaur, "An efficient routing protocol for FANET based on hybrid optimization algorithm," in *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 252–255, 2020.

[37] A. Khan, F. Aftab, and Z. Zhang, "BICSF: Bio-Inspired Clustering Scheme for FANETs," *IEEE Access*, vol. 7, pp. 31446–31456, 2019.

[38] I. U. Khan, I. M. Qureshi, M. A. Aziz, T. A. Cheema, and S. B. H. Shah, "Smart IoT Control-Based Nature Inspired Energy Efficient Routing Protocol for Flying Ad Hoc Network (FANET)," *IEEE Access*, vol. 8, pp. 56371–56378, 2020.

[39] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, and Y. Xu, "QMR:Q-learning based Multi-objective optimization Routing protocol for Flying Ad Hoc Networks," *Computer Communications*, vol. 150, pp. 304–316, 2020.

[40] L. A. L. da Costa, R. Kunst, and E. Pignaton de Freitas, "Q-FANET: Improved Q-learning based routing protocol for FANETs," *Computer Networks*, vol. 198, p. 108379, 2021.

[41] M. Y. Arafat and S. Moh, "A Q-Learning-Based Topology-Aware Routing Protocol for Flying Ad Hoc Networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1985–2000, 2022.

[42] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[43] J. Lansky, S. Ali, A. M. Rahmani, M. S. Yousefpoor, E. Yousefpoor, F. Khan, and M. Hosseinzadeh, "Reinforcement Learning-Based Routing Protocols in Flying Ad Hoc Networks (FANET): A Review," *Mathematics*, vol. 10, no. 16, 2022.

[44] 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.

[45] H. Wu, X. Tao, N. Zhang, and X. Shen, "Cooperative UAV Cluster-Assisted Terrestrial Cellular Networks for Ubiquitous Coverage," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2045–2058, 2018.

[46] P. Mahajan, A. Kumar, G. Chalapathi, and R. Buyya, "EFTA: An Energy-efficient, Fault-Tolerant, and Area-optimized UAV Placement Scheme for Search Operations," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2022.

[47] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D Placement of an Unmanned Aerial Vehicle Base Station (UAV-BS) for Energy-Efficient Maximal Coverage," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, 2017.

[48] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers Operations Research*, vol. 40, no. 6, pp. 1616–1624, 2013.

[49] Y. Chen, Y. Gao, C. Jiang, and K. J. R. Liu, "Game Theoretic Markov Decision Processes for Optimal Decision Making in Social Systems," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 268–272, 2014.

[50] B. Etzlinger, H. Wymeersch, and A. Springer, "Cooperative Synchronization in Wireless Networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 11, pp. 2837–2849, 2014.

[51] "RFD 900x-US Modem." https://store.rfdesign.com.au/rfd-900x-us-modem-fcc-approved/. Accessed: February 22, 2024.

[52] Q. Wang, M. Hempstead, and W. Yang, "A Realistic Power Consumption Model for Wireless Sensor Network Devices," in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 1, pp. 286–295, 2006.

[53] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Comprehensive Energy Consumption Model for Unmanned Aerial Vehicles, Based on Empirical Studies of Battery Performance," *IEEE Access*, vol. 6, pp. 58383–58394, 2018.

[54] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell system technical journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[55] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *The Bell System Technical Journal*, vol. 42, no. 5, pp. 1977–1997, 1963.