Environmentally-Aware and Energy-Efficient Multi-Drone Coordination and Networking for Disaster Response

Chengyi Qu, Francesco Betti Sorbelli, *Member, IEEE*, Rounak Singh, Prasad Calyam, *Senior Member, IEEE*, and Sajal K. Das, *Fellow, IEEE*

***Abstract*—In a disaster response management (DRM) scenario, communication and coordination are limited, and absence of related infrastructure hinders situational awareness. Unmanned aerial vehicles (UAVs) or drones provide new capabilities for DRM to address these barriers. However, there is a dearth of works that address multiple heterogeneous drones collaboratively working together to form a flying ad-hoc network (FANET) with air-to-air and air-to-ground links that are impacted by: (i) environmental obstacles, (ii) wind, and (iii) limited battery capacities. In this paper, we present a novel environmentally-aware and energy-efficient multi-drone coordination and networking scheme that features a Reinforcement Learning (RL) based location prediction algorithm coupled with a packet forwarding algorithm for drone-to-ground network establishment. We specifically present two novel drone location-based solutions (i.e., heuristic greedy, and learning-based) in our packet forwarding approach to support application requirements. These requirements involve improving connectivity (i.e., optimize packet delivery ratio and end-to-end delay) despite environmental obstacles, and improving efficiency (i.e., by lower energy use and time consumption) despite energy constraints. We evaluate our scheme with state-of-the-art networking algorithms in a trace-based DRM FANET simulation testbed featuring rural and metropolitan areas. Results show that our strategy overcomes obstacles and can achieve 81-to-90% of network connectivity performance observed under no obstacle conditions. In the presence of obstacles, our scheme improves the network connectivity performance by 14-to-38% while also providing 23-to-54% of energy savings in rural areas; the same in metropolitan areas achieved an average of 25% gain when compared with baseline obstacle awareness approaches with 15-to-76% of energy savings.**

***Index Terms*—drone mobility management, disaster response, reliable network construction, learning-based scheduling**

I. Introduction

Recently, unmanned aerial vehicles (UAVs) or drones have gained attention from both government and industry communities for a plethora of civil applications such as smart agriculture [1], traffic management [2], parcel delivery [3], and disaster response [4]. In these application scenarios, multiple drones with specific capabilities can execute complex and critical tasks, such as: detecting bugs in orchards [5], regulating the traffic in smart cities, delivering small parcels to customers, and saving lives in search and rescue operations. To accomplish these tasks, drones utilize a flying ad-hoc network (FANET), which involves them to be *flying nodes* within a network controlled by a ground control station (GCS). The GCS acts as a main stationary node that also guides the communication and coordination between the fleet of drones to meet application demands.

Barring FAA or local authorities restrictions [6], drones can potentially fly almost everywhere, and execute challenging tasks in a cooperative fashion [7] in order to, e.g., deliver first-aid and relief goods to first responders, capture images and videos above affected areas of interest, and update maps of disaster scenes. However, drones are energy-constrained vehicles and can only operate for a limited amount of time due to the small capacity of batteries, and hence they cannot continuously perform the assigned tasks. Therefore, their flights should be carefully optimized taking into account their number and the surrounding environment conditions (e.g., the terrain or current weather). Moreover, maintenance of the connection between drones and the GCS, as well as the FANET's proper functioning are primary concerns in these applications. State-of-the-art approaches aim to establish and maintain the FANET within energy constraints [8], [9], computation [10], [11], and communication [12], [13] capabilities. However, they do not consider external weather factors such as humidity or wind that could negatively influence the drones' energy consumption during their flights [14]. These factors could indirectly interrupt the air-to-air (A2A) and/or the air-to-ground (A2G) network communications between drones and the GCS. The interruption can be severe if, e.g., the drones remain without available energy from their batteries or are limited in their mobility due to wind conditions.

Another major challenge in the mobility management of drones occurs due to possible obstacles (e.g., trees, buildings) that can obstruct the radio signals and impact the FANET operation/efficiency in terms of connectivity, throughput, and latency. In other words, obstacles can influence the A2A and the A2G network communications between drones and the GCS, respectively, depending on their relative positions [15]. In fact, the presence of obstacles can disconnect the commu-

Manuscript received Month xx, yyyy; revised Month xx, yyyy; accepted Month xx, yyyy. Date of publication Month xx, yyyy; date of current version Month xx, yyyy. (*Corresponding author: Prasad Calyam.*)

This material is based upon work supported by the NSF under Award Number: CNS-1647182. This work was also partially supported by "GNCS – INdAM". Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Chengyi Qu, Rounak Singh, and Prasad Calyam are with the Department of ECE, University of Missouri - Columbia, USA; Francesco Betti Sorbelli is with the Department of Computer Science and Mathematics, University of Perugia, Italy; Sajal K. Das is with the Department of Computer Science, Missouri University of Science & Technology, USA.

Digital Object Identifier XXXXXXXX/TNSM.YYYYYYYY

nication links, increase the packet loss, and therefore increase the drone's energy wastage. In such cases, it is necessary to suitably plan routes for drones in order to keep and maintain reliable communications among the drones and the GCS.

In a disaster response management (DRM) scenario, it is common to have a heterogeneous drone setup to handle different tasks. For instance, large drones can carry goods to assist the victims, small drones can monitor the terrain, and other drones can bring network connectivity to the ground users if the fixed ground infrastructure is destroyed. Basically, the objective of drones in a DRM scenario is to adapt the 'drone assistance paradigm' [16], where they monitor areas in order to *provide assistance* to victims, and provide communications when the existing infrastructure is damaged. Drones spend energy not only to fly, but also to complete their assigned tasks. In DRM scenarios, when earthquakes or tornadoes happen, usually the terrain and/or the weather are not optimal for assisting people, thus affecting the use of drones. Hence, in the presence of obstacles and/or bad weather conditions, drones also have to properly use their residual energy for network establishment by acting as intermediate nodes to forward packets. This is done to keep alive the whole FANET, which is especially critical in DRM scenarios.

To the best of our knowledge, there is a dearth of works that address networking problems of using heterogeneous drones that cooperatively work in a DRM scene with environmental awareness. Particularly, there is a pent-up need for works on improving drones' *geographical obstacles avoidance* and *efficient energy usage*, while maintaining desired FANET connection performance in DRM scenarios.

In this paper, we present a novel environmentally-aware and energy-efficient multi-drone coordination and networking scheme that features a location-aided prediction algorithm coupled with a packet forwarding algorithm for drone-to-ground network establishment. Our novelty is in the approach of using Reinforcement Learning (RL) to estimate future drones' trajectories based on their coordination status and their on-board sensors information. Specifically, once the intermediate drone accurately predicts the position of the destination drone, then a list of preliminary decisions on where to forward packets is made. We consider various drone mobility models such as Gaussian Markov Model (GMM), Mission-Based Plan Model (MBPM), and Random Way Point Model (RWPM) within our prediction technique to address the DRM application requirements. Our packet forwarding algorithm features two drone location-based solutions, i.e., *heuristic greedy* and *learning-based* that can support heterogeneous drone operation requirements under DRM scenarios. The operation requirements involve improving A2A and A2G network connectivity (i.e., optimized packet delivery ratio and reduced end-to-end delay) despite environmental obstacles, and improving efficiency (i.e., by lowering energy use and time consumption) despite battery capacity limitations in re-establishing network connectivity.

We evaluate our proposed scheme by extending the work in [8], and comparing it with state-of-the-art multi-drone networking algorithms in a trace-based DRM FANET simulation testbed [17]. To realize various DRM environment settings, we simulate both rural and metropolitan areas in terms of the relative height between drone swarms and the average physical building height. We analyze the performance by evaluating network (i.e., packet delivery ratio, end-to-end delay) and energy (i.e., energy usage proportion for communication) metrics in diverse experimental settings in terms of different transmission range, amount of drones, physical obstacles densities and heights. Lastly, we present results to show how our strategy overcomes bottlenecks due to obstacles and can achieve better network connectivity performance observed under obstacle-free conditions. Further, we present results that show how our scheme outperforms state-of-the-art algorithms in the presence of obstacles in terms of improving network connectivity performance while also providing significant energy savings.

The rest of the paper is organized as follows: Section II presents related work. Section III provides an overview of our DRM coordination and networking problem. Section IV introduces the wind and obstacles models along with the solution approach. Section V introduces our location prediction model. Section VI details our heuristic-based and learning-based algorithms. Section VII presents our performance evaluation results. Section VIII concludes the paper.

II. Related Work

A. UAV-based Schemes in DRM scenarios

Handling of DRM involves, e.g., search and rescue operations that must be managed as promptly and efficiently as possible to save human lives. In this regard, the major problem is the lack of technologies that can provide the necessary situational awareness for the incident commanders making decisions to deploy first responder resources [18], [19]. State-of-the-art techniques deal with outdoor users assuming the knowledge of their 2D coordinates. In DRM situations such as floods, cyclones, or building fires, users are stuck in small or tall buildings situated in a 3D space. In these cases, it is possible to interact and rescue reliably, if the features are known a priori [12], [20].

Drones, compared to ground-based vehicles such as robots or cars, provide unique advantages such as the ability to observe devastated areas from the sky, flying above possible ruins and avalanches. Additionally, drones can provide monitoring and logistic services to address handling of DRM scenarios in the absence of traditional communication infrastructure [21], [22]. By utilizing a high definition camera on drone, works in [23] utilized pre-and post-disaster images to validate floods in a region to provide timely aid and relief to stranded people. Moreover, with the help of multi-drone and ground devices, works in [24] created a flight plan in the form of the Vehicle Routing Problem (VRP) in order to solve a problem on quickly and efficiently collecting messages for all refugees dispersed in shelters after disasters occur. Our proposed communication and networking scheme can be used in works such as [25], [26] and help with achieving drones' monitoring function in order to provide situational awareness for rapid and effective decision making to handle DRM. Although reliable communication architectures have been studied in [27], [28] in the context of drones, they do not address the underlying multi-

drone coordination and networking aspects that are necessary to deploy DRM applications.

Another usage of drones in DRM scenarios are related to the potential of extending the communication under situations when network failure occurs or there is intermittent network connectivity. Authors in [29] present an optimal model for computing the trajectories of the drones while guaranteeing the total coverage of the ground mobile sensors and connectivity among the drones with a central base station dedicated to data processing. In addition, investigations on efficiently placing drones in affected area where infrastructure is partially broken can be seen in [30], [31] as well. The whole problem is treated as a multi-objective problem of UAV placement, users-UAV connectivity, distance, and cost. Correspondingly, optimal and heuristic approaches are then devised. Although the authors pursue the objective to keep alive the connections among drones, no any energy-efficient algorithms are proposed, as considered in our work. In terms of the various communication purposes, authors in [32] and [33] propose to use intermediate drones to transfer messages from ground-based Long Range (LoRa) nodes to the remote BS. However, intensive streaming data such as video and audio are not supported for the implementation of energy-efficient routing protocols. In our proposed scenario, we utilize embedded devices such as Nvidia Jetson Nano and Raspberry Pi to generate a ac-hoc network between A2A and A2G links.

B. UAV-based Routing Protocols

To achieve reliable GCS/drones communication in a DRM scenario, a suitable packet forwarding algorithm is necessary. In [34], authors proposed a strategy that salvages packets in the presence of void nodes, providing a low-complexity and low-overhead recovery for the Greedy Geographic Forwarding (GGF) failure. In a DRM scenario, it is crucial to consider a location-based packet forwarding protocol that can be used in ad hoc network deployments. The geographic routing protocol, Greedy Perimeter Stateless Routing (GPSR) protocol [35], utilizes GPS information to assist the packet forwarding procedure. To this end, the GPSR algorithm could be applied into every-day surveillance systems since uncertain and unpredictable conditions are not commonly present.

Our work is highly related to studies in [36], [37], wherein authors describe how a packet forwarding strategy can be used inside these protocols for large density drone-network deployments, through experiments in multi-drone settings. Based on results in prior works, the packet forwarding algorithm described in GPSR outperforms other state-of-the-art packet forwarding strategies described in both proactive or reactive-based protocols. Other solutions that consider only the network traffic congestion based on GPSR utilizing multi-drone orchestrations have been presented in [38], [39], [40]. In these studies, constraints such as wind, and drone's battery are not fully considered in the objective function. Our proposed approach takes environmental features (i.e., wind, obstacles) into account providing solutions at earlier stages (i.e., pre-flight) so that traffic congestion issues do not frequently occur during the establishment of the network links.

The effect of the wind with regards to the A2G communications between drones has been investigated in [41]. Specifically, they study how multiple drones can be effectively used for providing wireless service to ground users. Given the locations of drones, they must expend a control time to adjust their positions dynamically so as to serve multiple users. To minimize this control time, the speed of rotors is optimally adjusted based on both the destinations of the drones and external forces (e.g., wind and gravity). However, no obstacles have been taken into account as done in our work. The wind factor in a DRM scenario has been studied in [42] similar to our work, however they do not consider the presence of obstacles as accounted in our work. Our approach is also motivated by the work in [43], where a suitable packet forwarding algorithm for DRM operations is studied. The authors proposed a Location-Aided Delay-Tolerant Routing protocol (LADTR) that employs location-aided forwarding combined with a Store-Carry-Forward (SCF) method. The goal of the proposed strategy in this routing protocol is to guarantee the connection rate between drone nodes and enable a high packet delivery ratio.

In contrast to these prior works, our proposed approach considers environmental awareness in terms of energy consumption as well as the presence of obstacles in drone path computations at the GCS for A2A and A2G links. To enhance our packet forwarding algorithm by considering both energy efficiency and obstacles awareness, we build upon the recent prior work in [44]. In this work, authors studied energy consumption issues for mobile devices management in the context of a mobile edge computing paradigm; note that drone and related edge device mobility use cases were not addressed in this prior work. The only considered human mobility while handling user requirements of energy conservation over low-latency or vice versa in visual edge-based application data processing. Specifically, they presented the SPIDER algorithm, which was built upon recent advances in the geographic routing area [45]. This study presented a novel AI-augmented geographic routing approach (AGRA) that uses physical obstacle information obtained from satellite imagery by applying deep learning at a network-edge site. The SPIDER algorithm is shown to perform better as a packet forwarding strategy than other stateless geographic packet forwarding solutions, as well as, stateful reactive mesh routing in terms of packet delivery success ratio and path stretch. Our work builds on the SPIDER algorithm, and is suitable for high-density drones network deployments and considers environmental awareness in terms of energy consumption and obstacle avoidance in DRM scenarios.

In Table I we make a comparative study and summarize the key differentiators and novelty of our work contributions in contrast with prior related works.

III. DRM Coordination and Networking

In this section, we present the multi-drone coordination and networking problem for DRM, describing the essential system components, related requirements, and inherent assumptions. Following this, we present our solution overview.

TABLE I
COMPARISON BETWEEN THE EXISTING WORKS.

Work	Comments
[18], [19]	**Objective**: Determine the optimal number of drones and drone chargers **Differences**: No communications among drones
[21], [22]	**Objective**: Compute a suitable path planning for drones **Differences**: No communications
[12], [20]	**Objective**: drones as communication devices in video analytics scenario **Differences**: No communications nor obstacles
[23], [24]	**Objective**: drones aerial image recognition **Differences**: Wind is considered, no communications nor obstacles
[25], [26]	**Objective**: drones aerial image recognition **Differences**: No communications nor obstacles
[27], [28]	**Objective**: Energy-efficient drones-IoT sensors data collection **Differences**: No communications
[29], [30], [31]	**Objective**: Data collection problem using the minimum number of drones **Differences**: No obstacles and energy constrains
[32], [33]	**Objective**: drones as relays for communications, use LoRa **Differences**: No obstacles, no wind, no streaming data transmission
[34], [35]	**Objective**: Computed a constrained path planning **Differences**: In 2D, no drone related communications.
[36], [37]	**Objective**: Architecture formed by IoT devices as edge node carriers, and consider obstacles and energy **Differences**: No multi-drone coordination
[38], [39], [40]	**Objective**: Optimal elevation angle of drones among rescuers **Differences**: No wind and 3D models
[41], [42], [43]	**Objective**: drones as flying base stations considering wind and obstacle **Differences**: Selected drones for network purpose only
[44], [45]	**Objective**: IoT devices routing protocol considering wind, obstacle and energy **Differences**: No 3D models and cooperation with drones

A. Multi-Drone Networking System

We consider a DRM scenario that involves multiple critical tasks (e.g., search and rescue after an earthquake, providing relief goods to people) being executed by heterogeneous drones on a given FANET topology. Figure 1 illustrates the system setup including a GCS and three different types of drones [46] i.e., *delivery drones*, *monitoring drones*, and *map drones*. The delivery drones carry first aid (e.g., medicines) and relief goods (e.g., water) to people at the disaster scenes, the monitoring drones have embedded cameras used for searching for objects and finding missing people, and finally the map drones are in charge of keeping the rescue area map up-to-date. The GCS sends requests to drones for executing specific tasks on certain locations (e.g., video recording above ruins). Furthermore, the drones will send back to the GCS the retrieved situational awareness information.

B. System Requirements and Assumptions

The fundamental requirement that needs to be guaranteed is the establishment of reliable communication links between the drones and GCS. In general, the communication radius is not sufficiently large for allowing single-hop links, due to the fact that the topology can be arbitrary, and also the presence of obstacles can make the links unstable or even blocked. Hence,

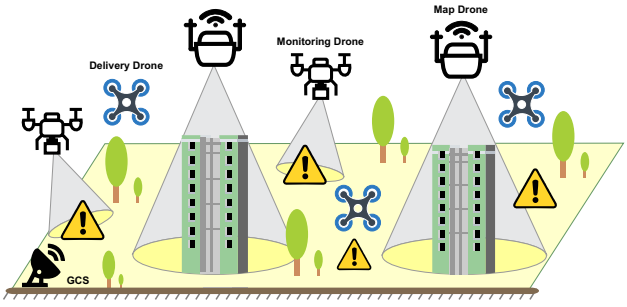


Fig. 1. Coordination of drones in the proposed DRM scenario. Delivery drones carry first aid and relief goods, monitoring drones search for missing people or objects, and map drones keep the rescue area map up-to-date.

communication links should be built according to a multi-hop paradigm. In our DRM scenario, high-throughput links are required only when data transmission is in process. To save energy, delivery drones could individually deliver goods without further notice to the GCS [47], monitoring drones can perform pre-processing functions on-board without guidance from the GCS, and map drones can cache the image/video data inside the embedded storage until the queue is full.

In our scenario, the GCS is not an energy-constrained device and also its computational performance outperforms that of the drones. We assume our system to be comprised of heterogeneous drones with knowledge of their global positions because they are enabled with GPS capability. Delivery drones have pre-computed routes for carrying goods to locations and move with respect to the MBPM [48]. They have sufficient energy for executing delivery tasks and flying safely back to the ground. They can act as forwarding drones for generating connection links only if they have residual energy. Moreover, they can lose the connectivity with GCS due to network bandwidth and flying range constraints. Monitoring drones initially move according to the RWPM [49] flying through many pre-determined points of interest (PoIs). Once they discover objects or people on the ground, they change their mobility model following the GMM when executing tasks on specified target areas, as explored in [50]. Map drones fly at higher altitudes and over longer distances than the other ones, having more computation resources for executing pre-stage map generation and image mosaicing tasks. However, it is not required by them to constantly transmit back the captured data to the GCS, because severe and adverse weather conditions or other unexpected events may result in task failure. Thus, we propose a periodic communication scheme between map drones and the GCS for transferring the stored data on drones to the GCS according to a suitable queuing mechanism for the data (see Section VII-A). Similarly, since the purpose of map drones is to cover the whole area in a short amount of time, a GMM or a MBPM is used to achieve the related task. We assume the delivery and monitoring drones' height to be $100\,\mathrm{m}$, while the same for map drones is $200\,\mathrm{m}$. Also, we consider the average obstacles' height to be $100\,\mathrm{m}$.

IV. Environmentally-Aware Modeling

In this section, we first detail our novel drone energy model influenced by the wind factor i.e., the wind factor affecting the drone's energy consumption during the flight. Following this, we investigate how the presence of obstacles in the area can affect the wind itself, and in turn the drone's energy consumption. Lastly, we present the obstacles model which influences the network communications between drones.

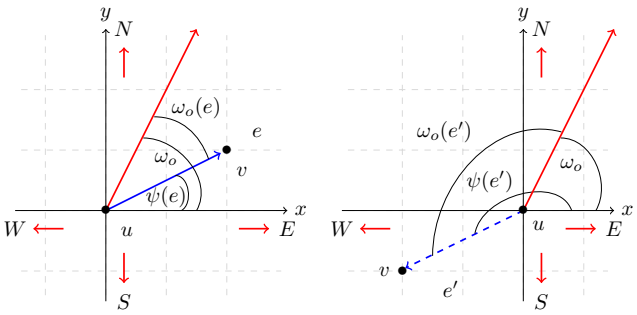
A. Drone Energy Model

In the following, we discuss how the wind influences the drone's energy consumption during the flight. This model has been adapted from [14], [51] and can be applied to calculate, in real-time, the energy consumption of the drone in a given environmentally-aware DRM scenario.

Let the *global wind* $\omega = (\omega_s, \omega_o)$ be the wind in the rescue area, where ω_s and ω_o are, respectively, the speed and the direction (orientation) of ω. While at any instant the entire rescue area is subject to the same wind, the conditions of the global wind can change over time. To define the *relative wind direction* $\omega_o(e)$ experienced by the drone while traversing edge $e = (u, v)$, we build a Cartesian coordinate system with origin in u (two examples are depicted in Figure 2). Let $-90° \leq \arctan(x) < 90°$, for any $x \in \mathbb{R}$. On the straight line e traversed by the drone from $u = (x_u, y_u)$ towards $v = (x_v, y_v)$, the relative wind direction is $\omega_o(e) = \omega_o - \psi(e)$, where:

$$\psi(e) = \begin{cases} \arctan(\frac{y_v}{x_v}) \mod 360° & \text{if } x_v > 0 \\ 180° + \arctan(\frac{y_v}{x_v}) & \text{if } x_v < 0 \end{cases} \quad (1)$$

is the angle direction of edge e. Flying along routes with different angle directions, the drone experiences different relative wind directions. Note that, $\psi(e)$ and $\psi(e')$ differ by 180° because $\omega_o(e') = \omega_o - \psi(e') = \omega_o - \psi(e) - 180° = \omega_o(e) - 180°$. For simplicity, while traversing e the relative wind $\omega_o(e)$ is assumed stable.



(a) $\omega_o(e)$. (b) $\omega_o(e')$.

Fig. 2. Relative wind directions of an edge e (solid blue (a)) and e' (dashed blue (b)), and the global wind direction (red). Note $\psi(e') = \psi(e) + 180°$.

The motion of drones is regulated by physical properties, such as the gravity and forces due to forward motion and wind [52], [53]. The total required thrust is $T = Wg + F_{\mathrm{D}}$, where W is the total weight of the drone which includes the *payload weight* (mass) m_p, $g = 9.81\,\mathrm{m/s^2}$ is the gravitational constant, and the total drag force F_{D}. The drag force F_{D} depends on the properties of the fluid and on the size, shape, speed, and direction of the movement of the drone, and can be estimated as follows: $F_{\mathrm{D}} = \frac{1}{2}\rho\, s_a(e)^2\, C_D A$, where ρ is the air density, $s_a(e)$ is the drone's relative air speed, $A = \pi R^2$ is the cross sectional area (R is the rotor radius), and C_D is the drag coefficient. The air speed $s_a(e)$ depends on the global wind speed ω_s and the relative wind direction $\omega_o(e)$. The air speed $s_a(e)$ can be calculated as:

$$s_a(e) = \sqrt{s_N^2 + s_E^2}, \quad (2)$$

where:

$$s_N = s_d - \omega_s \cos(\omega_o(e)) \quad (3)$$

$$s_E = -\omega_s \sin(\omega_o(e)), \quad (4)$$

while s_d is the *drone speed*. Note that when $\omega_s = 0$, i.e., no wind, $s_a(e)$ is the same for every drone direction e. Instead, if $\omega_s \neq 0$, when $\omega_o(e) = 0°$, i.e., when the global wind direction and the drone direction have the same orientation, the East component s_E of the friction is null, and the North component s_N has the minimum value, implying $s_a(e)$ is minimum. Also, when $\omega_o(e) = 180°$, i.e., when wind and drone direction are opposite, s_E is null, and s_N is maximum, and thus $s_a(e)$ is maximum.

Having computed the value of the total thrust T, it is possible to estimate the required power P for a steady flight. The theoretical minimum power depends on the area swept by the rotors, which in turn depends on the diameter D of the n rotors. So, given the previous computed values and parameters, the *hovering power* P_H can be calculated as follows [54]:

$$P_H = \frac{T^{3/2}}{\sqrt{\frac{1}{2}\pi n D^2 \rho}}. \quad (5)$$

With forward motion, the *motion power* P_M can be calculated from conservation of momentum and thrust T as follows [55]:

$$P_M = T(s_d \sin(\alpha) + s_i), \quad (6)$$

where $\alpha = \arctan\left(\frac{F_{\mathrm{D}}}{Wg}\right)$ is the pitch angle [55], and s_i is the induced velocity required for a given thrust T. Then, s_i can be obtained by solving the implicit equation [55]:

$$s_i = \frac{s_h^2}{\sqrt{(s_d \cos(\alpha))^2 + (s_d \sin(\alpha) + s_i)^2}}, \quad (7)$$

where $s_h = \sqrt{\frac{T}{2\rho A}}$ is the induced velocity at hover [55]. Note that, even not made explicit in the notation, F_{D} and so α and P_M depend on the drone's direction e. Hence, fixed a global wind $\omega = (\omega_s, \omega_o)$, the energy efficiency (or, unitary energy) $\mu(e)$ of travel along a segment e is calculated as the ratio between motion power consumption P_M and average ground speed s_d of the drone, i.e.,

$$\mu_\omega(e) = P_M / s_d. \quad (8)$$

Therefore, the *energy consumption for motion* $\mathcal{E}_\omega^M$ along one edge e of length $\lambda(e)$ can be expressed as:

$$\mathcal{E}_\omega^M(e) = \mu_\omega(e) \cdot \lambda(e). \quad (9)$$

Finally, the *energy consumption for hovering* $\mathcal{E}_\omega^H$, which depends on the time t the drones hovers at a particular position, and on the hovering power P_H previously calculated, can be

expressed as:

$$\mathcal{E}_{\omega}^{H}(t) = P_H \cdot t. \tag{10}$$

So, with the *energy consumption for motion* $\mathcal{E}_{\omega}^{M}$ and the *energy consumption for hovering* $\mathcal{E}_{\omega}^{H}$ we can estimate, in real-time, the drone's energy consumption when it is employed in a DRM scenario.

B. Wind-Awareness in Energy Consumption

In the previous section, we introduced the wind model which can be used in an obstacle-free environment. However, possible obstacles can influence the drone's energy consumption. Therefore, in this section, we discuss this scenario when physical obstacles could influence the wind direction as well as obstruct the A2A and A2G communication links.

To better formulate this problem, we describe a wind Computational Fluid Dynamics (CFD) model which can be applied in metropolitan areas. The wind CFD model attempts to simulate the interaction of the wind where the surfaces are defined by boundary conditions. State-of-the-art CFD models employ the principles of the Navier-Stokes equations. Simulations are then conducted by solving the equations iteratively in steady-state or transient conditions. There is a dearth of works that consider drone swarm applications applied in metropolitan areas where the average height of drones, during their missions, is lower than the height of the building [56]. In such cases, once the drones fly across a group of skyscrapers, the energy consumption will be locally recalculated according to the wind speed. In addition, the directions experienced by the drones in these places are computed by the adopted (and simulated) wind CFD model. Figure 3 shows the result of a simulation by using the open-source wind CFD model simulator "SimScale" [57] to simulate the local wind directions and speed in a given metropolitan area.

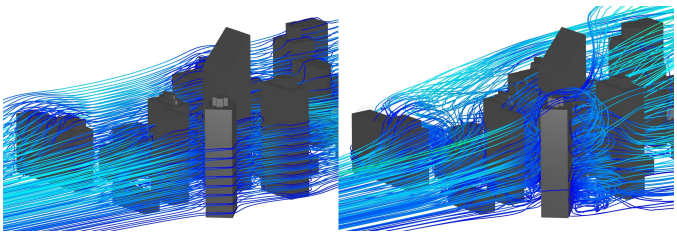


Fig. 3. Results of a simulation when the wind CFD model simulator SimScale is employed in a metropolitan area at run 0 (left) and run 1000 (right).

Figure 3 (left) shows the initial wind direction (orientation of lines) and speed (brightness of lines) at time 0, while Figure 3 (right) shows the results after $10\,\mathrm{mins}$ of simulation (100 runs each minute). As we can observe, the global wind (i.e., direction and speed) is defined to be a constant for an observed time instance on a particular area of the map, and can change along the time/area. Consequently, the local wind direction and speed will also change according to the elapsed time and the drones' positions. For example, if one drone is hovering at a particular position capturing a video for monitoring purposes, the energy consumption will be calculated and estimated under two circumstances (either with global wind or with local wind) based on the relative height between the flight height and the height of the closest obstacle buildings. To this end, it is necessary to predict the real-time location of the *monitoring drones* and *map drones* to update the energy consumption and to re-establish continuous network communications once interrupted (details can be seen in Section V).

To simplify our energy consumption calculation, we assume that: i) the average flight height of the *monitoring drones* will be fixed as $50\,\mathrm{m}$ while the average flight height of the *map drones* will be fixed as $100\,\mathrm{m}$ if no addition information is given, and ii) the projection along the z-axis of the wind direction and speed will be ignored, and therefore only 2 dimensions concerning the drone's flight will be considered. Moreover, for simplicity, we consider the wind as an external factor when computing the drone's energy consumption. In fact, the drones' routes are pre-planned in advance, and these cannot be dynamically modified even if the drones have to fly against the wind i.e., when the wind is a "head-wind". Therefore, the wind energy model is only taken into account for computing the residual energy of the drones, and the proposed packet forwarding algorithms only check that residual energy in order to determine suitable solutions. Thus, our proposed packet forwarding algorithms are indirectly affected by the wind because the residual energy of drones can change according to a given situation.

C. Obstacle-Awareness in Communication

In our proposed scenario, *monitoring drones* and *map drones* have to request the re-establishment of the packet forwarding path in order to maintain the connection to the GCS. Once this request is initiated, a packet forwarding procedure is performed. Notice that, a discontinued communication can be notified by the WiFi chip board embedded on each drone, rather than by the physical communication channel between them. That is to say, there is a situation when the communication tasks discontinue (e.g., no video transmission between links), while the GCS can still have a drone within the line-of-sight. To simplify our experiments, we do not consider the situation when drones are beyond the line-of-sight. The detailed distance choice between drones is reported in Section VII-A. Let us consider a node n that forwards a packet p towards a destination d for re-establishing A2A and A2G communication links. The node n has to decide which neighbor must receive the packet p to progress towards d. Such a decision also needs to balance the neighbor's residual energy and the total throughput of packet p, and should minimize the following weighted energy function f as follows:

$$f(n, d, \theta) = \theta \cdot \tau(n, d) + (1 - \theta) \cdot \epsilon \tag{11}$$

where, $\tau(n, d)$ is the normalized updated shortest path energy [45] with respect to the obstacle blockage, ϵ is the average residual energy at node n, and $\theta \in [0, 1]$ is the balancing parameter.

The purpose is to find the best θ value which gives the minimum energy consumption from node n, and keeps the network connection continuous and stable by calculating the

path. Once n is aware of its propagation Fresnel zone radius and the i^{th} obstacle's center C_i, it computes $\tau(n,d)$ as follows:

$$\tau(n,d) = \sum_{i=1}^{M} \frac{O_i - 1/\sqrt[\delta]{\|n-d\|_2}}{\|n - C_i\|_2^{\delta}} \tag{12}$$

where $\|\cdot\|_2$ represents the Euclidean distance, δ is the attenuation order of obstacles' potential field [45], M is the number of obstacles, and O_i is the intensity of i^{th} obstacle induced by the destination node d, calculated as follows:

$$O_i = \frac{F_i^{\delta}}{\delta(\|d - C_i\|_2 + F_i)^2} \tag{13}$$

where F_i represents the i^{th} Fresnel zone in 3D. We suppose that two devices can communicate if the blockage, due to the presence of obstacles, is up to 20% of the Fresnel zone [58], otherwise, the communication is obstructed.

D. Solution Overview

As shown in Figure 4, our multi-drone coordination and network strategy consists of two main stages i.e., *data collection* (shown left) and *data transmission* (shown right). The DRM application starts with either one map drone (e.g., generating and storing data) or monitoring drone (e.g., flying for PoIs) acting as a target drone. When it requires to communicate with the GCS, a request for establishing a connection is created. At this point, if the GCS is in range, the target drone will finish the packet forwarding and the link is established; otherwise, a list of candidate drones in the neighborhood is generated. From this list, the target drone can use a simple packet forwarding algorithm (SPF algorithm, Section VI-A) to establish links. Alternatively, a more accurate packet forwarding algorithm (HGPF algorithm, Section VI-B) is used to optimally balance the connection link quality and energy consumption. Finally, if there is no restricted task execution time requirement on the target drone, an integrated RL based packet forwarding algorithm (IRLPF algorithm, Section VI-C) is applied. In this case, the system not only achieves the balance on network connection quality and energy consumption, but also schedules task execution such that the optional forwarding drone for the neighbor drone candidate list will appear in the right position when the target drone makes a request.

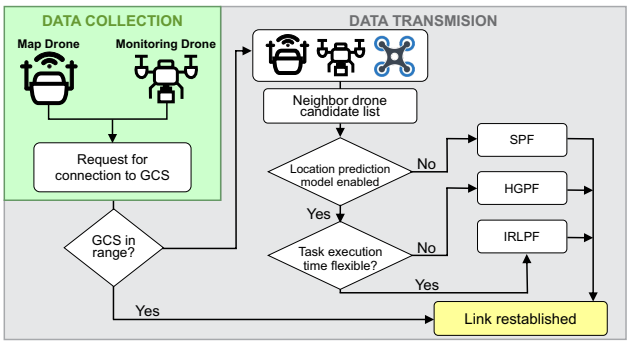


Fig. 4. Obstacle-aware and energy-efficient multi-drone coordination and networking solution steps overview.

V. Location Prediction Modeling

As discussed earlier, we assume that all the drones are connected forming a FANET. By this, they communicate the mapping and monitoring information over the same network to the delivery drones in order to carry out a delivery task. Consequently, the network topology of the multi-drone system keeps on changing based on the mobility of the drones.

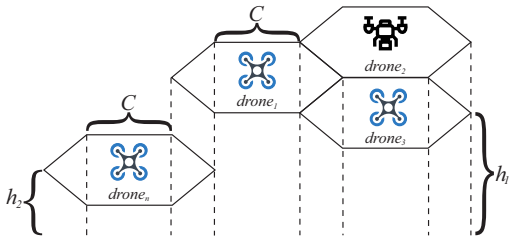


Fig. 5. Multi-drone distribution in the environment state space.

A. The Position Estimation of Drones

The position estimation of the drones must be performed in very short intervals of time using the new coordinates being updated rapidly within the FANET. Each drone in the FANET is considered to have a GPS module and an Inertial Measurement Unit to record its current location. This information is broadcast to the FANET so that the other drones in the vicinity are recognized for packet or information transfers when needed. In prior works such as [59], the Extended Kalman Filter was considered as the best algorithm choice to predict the location coordinates of a drone. However, the location information of the drones can be misleading if the sensors malfunction or the accuracy of the information gets compromised. Therefore, we have developed a novel multi-agent deep RL approach that predicts the drones' coordinates in the FANET for effective communication and packet transfer.

We allot a hexagonal area as in [60] of side C for each drone as shown in Figure 5, in order to maneuver inside them and change their heading to any direction to efficiently cover a surveillance area and discover PoIs. For initiating a packet transfer between the drones in the environment, we consider the location of them as $P_n = (x_n, y_n, h_n)$, where x_n, y_n are the location coordinates, and h_n denotes the altitude. Moreover, D_{ij} is the distance among any two drones $i \neq j$.

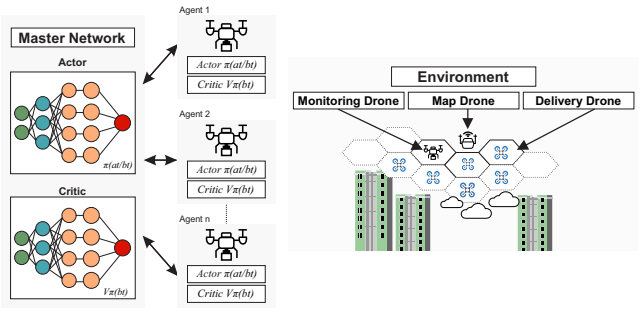


Fig. 6. Location prediction using multi-agent A3C network.

Due to the potential uncertainties in the environment that affect the drones' localization and orientation, the location

prediction of the drones can be formulated as a Partially-Observable Markov Decision Process (POMDP) defined by the tuple $[S, A, P, Z, O, R]$, where S, A, P, Z, O, and R denote the states, actions, probability of transition, probability distribution function for observing states, observations, and rewards, respectively. The environment states are defined as $s_t = (P_i, \phi_i, P_j, \phi_j, D_{ij})$ where P_i and P_j are the positions, while ϕ_i and ϕ_j are the headings of the packet transferring and receiving drones. The actions performed by the drones (agents) are defined as a set of 7 flight operations a_t, i.e., hover, forward, backward, up, down, yaw-left, and yaw-right. Moreover, the rewards R_t are defined as follows: $+100$: successful packet transfer; $+10$: every drone action; -50: going out of Hexagonal cell; and -100: collision with obstacles/drones;

Let T be the episode in which the agent performs action in the state space. The agent does not track the exact states $s_t \in S$, but uses the observations $o \in O$ in any given episode T. Therefore, it has to rely on the history of actions and observations $S_t = (a_t, o_t; a_{t-1}, o_{t-1}; \ldots; a_0, o_0)$ to perform intelligent actions that allow higher rewards. However, this history S_t exponentially grows with every action taken and every state observed. The agent rather chooses to utilize the belief states b which are single valued and represent the probability distribution $Z = p(o_t|s_t, a_t)$ over all possible states s_t in a given episode. These belief states are a sufficient measure of history, and given the current belief state b_t, the POMDP aims to find an optimal policy π^* that maximizes a value function V^π while following a sequence of actions and observations. The value function V^π is thus given by:

$$V^\pi(b) = \mathbb{E}[\sum_{i=0}^{\infty} \gamma^{i-t} R_i | b, \pi]. \tag{14}$$

Our approach, aims to solve the POMDP problem using a multi-agent Asynchronous Advantage Actor Critic (A3C) Network such that the best possible actions are chosen in given states, and the cumulative reward G_t (i.e., the accumulated discounted return) gets maximized, as follows:

$$G_t = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(b_t, a_t)] \tag{15}$$

where γ is the discount factor. Finally, the action value function is given by:

$$Q_\pi(b_t, a_t) = \mathbb{E}[\sum_{n=0}^{\infty} \gamma^n R(b_t, a_t) | b_t, a_t] \tag{16}$$

B. A3C based solution

Compared to other deep RL approaches, an A3C network allows multiple agents to interact in a parallel fashion with the environment in order to generate individual policies that are the outputs. Each episode in the A3C network stochastically progresses, and each corresponding action is probabilistically sampled. The *actor* and related *critic* networks within the A3C network are deep neural networks (DNNs) and have target networks. In the following, we detail the proposed A3C network functioning. A master network has an actor network that generates a policy $\pi(a_t|b_t; \theta_a) = P(a_t|b_t; \theta_a)$ and a critic network that generates the state value function to test the expected return under belief state $V_\pi(b_t; \theta_v)$, where, b_t is the belief state of an episode, θ_a denotes weights of actor network, and θ_v denotes the weight of critic network. The weights in these cases are updated using back-propagation. Each copy of the Master Network is sent to the agents as shown in Figure 6. The actor network is trained with the loss function:

$$L(\theta_a) = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{n} [-Q(b_t(i), \pi_b(i); \theta_a)]^2 \tag{17}$$

where, N_{batch} is the batch size and the critic network is trained with the loss function given by:

$$L(\theta_v) = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{n} [y_t(i) - Q(b_t(i), a_t(i); \theta_v)]^2 \tag{18}$$

where $y_t(i) = R_t + \gamma \cdot Q^{'}(b_{t+1}, \pi^{'}(b_{t+1}; \theta_a^{'})|\theta_v^{'})$ and $\pi^{'}(b|\theta_a^{'})$ and $Q^{'}(b, a|\theta_v^{'})$ are target networks of actor and critic networks. Due to the fact that gradient methods are used to optimize the network weights, there are chances of high variance occurrence in the critic network. Therefore, we employ an advantage function $\Omega(b_t, a_t) = Q(b, a) - V^\pi(b_t, \theta_v)$ to overcome this high variance problem. Upon solving the POMDP using the proposed A3C network, we get a policy $\pi : b \rightarrow a$ that maps the actions to belief states b. The optimal policy obtained from the actor and critic network operations is given by: $\pi^*(b) = \arg\max_b V^\pi(b)$.

As detailed earlier in Section IV-C, an RL technique is used along with a FANET through which the drones communicate their mapping and monitoring information. The location estimates $P_n = (x_n, y_n, h_n)$ of two drones are provided as state constraints in the A3C Network that generates a policy; following this policy, the cumulative reward G_t increases. This results in each drone's location getting updated to follow an optimized path in their own trajectory. As the update happens very rapidly, the optimized path information of each drone changes in the FANET in very short intervals of time. The optimized location information of the drones along with the previous location information is broadcast in the FANET. Thereby, the drones place themselves in close proximity to transfer information packets among themselves. Hence, our RL technique (A3C) predicts the mobility of drones by using the location information.

VI. Packet Forwarding Solution Approach

In this section, we initially describe a simple technique called Simple Packet Forwarding (SPF), and two novel algorithms called Heuristic Greedy Packet Forwarding (HGPF) and Integrated RL-based Packet Forwarding (IRLPF).

A. The SPF Algorithm

To address the trade-off in energy-efficiency versus continuous connection, there is a need for flexible packet forwarding algorithms among the drones and GCS. A choice among *stateful* and *stateless* packet forwarding design has to be carefully determined based on the cost of maintaining forwarding paths or infinite loops in paths [44]. Consequently, we rely on a stateless instead of stateful design, and we consider on-demand communications when requested. By design of SPF, if any drone is within the transmission range of a target drone, then it will be selected as a candidate forwarding drone for

forwarding packets. If there are no candidate drones, the target drone will hover and wait in the current position, thus pausing the task. As a second attempt, if no forwarding drone candidate is within the transmission range, the target drone forwards the data to the farthest drone in range. Note that any drone can act as an intermediate or forwarding drone. The data will be transferred through multi-hop drones to the GCS. Since SPF does not use any predicted drones' positions, it can be implemented in any drone.

TABLE II
SPF ALGORITHM PERFORMANCE COMPARISON

	Throughput (Mbps)				Energy (J)	
	GPSR	AODV	HWMP	Our	GPSR	Our
low	2.48±1.7	1.94±1.8	0.84±0.2	2.18±0.6	522±46	519±42
	2.35±0.9	0.87±0.3	0.77±0.2	2.26±0.6	515±13	506±18
high	2.06±1.8	1.26±1.5	0.77±0.3	2.05±1.1	497±47	498±40
	1.95±0.8	0.62±0.4	0.72±0.2	2.32±0.5	499±16	495±40

Despite its simplicity, it outperforms in terms of energy consumption and network throughput, other existing stateless algorithms which do not rely on predictions. As a baseline, we have chosen Greedy Perimeter Stateless Routing (GPSR) [61], Ad-Hoc On Demand Distance Vector (AODV) [62], and the Hybrid Wireless Mesh Network (HWMP) [63]. Table II shows the average throughput and energy consumption on target drone for link establishment with different density of obstacles (10%: low, 60%: high) and number of drones (10, 30) with the simulation settings described in Section VII. Since each protocol follows different path forwarding strategies (e.g., AODV and HWMP do not consider energy constraints), it is hard to compare side-by-side performance with each of these approaches. Hence, in this comparison, some part of the results are omitted. However, if we do not consider predicting the future positions of drones, both SPF and GPSR are Pareto-optimal, i.e., they have no alternative solutions. In order to have improvements, more features need to be considered with additional information as detailed in the following algorithms.

B. The HGPF Algorithm

As we can observe from SPF, if the prediction of the future position of the drones is not used, the target drone may have a lesser chance to find the potential candidate forwarding drone when it flies beyond the GCS's transmission range. Consequently, in HGPF design we use the RL-based location prediction A3C method to enhance the packet forwarding performance in the presence of environmental obstacles, while using the residual energy from the forwarding drone candidates. Based on the drone position prediction information, we can obtain the local relative distance between the drones in advance. Thus, to increase the accuracy of the drone path computation procedure, we use HGPF that proactively performs a greedy calculation of the local-optimal path solutions.

To simplify our algorithm and save time in real-world experiments, we periodically run the RL location prediction every $10\,\mathrm{s}$ according to state-of-the-art drone position model evaluation metrics [64]. For every $10\,\mathrm{s}$, we found that the corresponding time is within acceptable range ($2.20 \pm 0.05\,\mathrm{s}$), and the estimation error of the model is relatively small ($0.43\,\mathrm{m}$). Therefore, for each time slot, we can estimate the $f(n, d, \theta)$ results given $\theta \in \{0, 0.2, 0.5, 0.8, 1\}$ values. Corresponding to these results, we select the minimum value f with the related θ value to guide the next $10\,\mathrm{s}$ flight. Thus, by such a greedy finding of the discrete local minimum value, we enhance the overall performance in terms of the total residual energy and overall throughput.

The main purpose of HGPF is to utilize the drone position model for predicting the relative position of drones. The time period t in the algorithm is by default set to $10\,\mathrm{s}$. For each time period, all θ values are calculated for each alternate neighbor node. Following this, we calculate the performance gain on the drone (i.e., throughput gain by processed energy) of each θ, and perform a greedy select of the highest performance gain. In addition, we will use this θ until the next time period t, when HGPF is invoked again. Consequently, HGPF will always select the local optimal choice, which however may not be the overall best for the entire system.

C. The IRLPF Algorithm

Since HGPF provides the local optimal choice by time period, it may lack information on the global optimal performance gain over the entire system. However, during DRM, the position of the geographical obstacle may change frequently. In this case, it is not reasonable to obtain predictions of exact positions of obstacles that may block the connection. Thus, we propose an alternative way to indirectly provide forwarding drone candidates to target drone positions by scheduling the take-off time for different tasks. For example, once a monitoring drone takes off and starts the task, it will require a connection when PoIs are in sight. If we plan to take-off a delivery drone in the monitoring drone's transmission range, the monitoring drone will have a higher chance to search and find the delivery drone and select it as a forwarding drone. To achieve this, we abstract the multi-drone multi-task packet forwarding procedure as a Markov-decision process (MDP), which can provide a mechanism to evaluate and learn the potential take-off time and the θ value by using rewards. Thus, the optimization problem can be redefined as: *find the optimal task schedule which minimizes the global performance gain for the packet forwarding link generation*. This finite-time MDP problem can be expressed as $M = (S, A, P, R, T)$ where S is the state space, A is the action space, P is the probability function that indicates the probability of action $a \in A$ in state $s \in S$ at time $t \geq 0$ will lead to state $s' \in S$ at time $t + 1$, R is a reward function, and T represents the last acceptable time when the drone has to take-off and process the task.

To ensure that the near optimal θ can be chosen at every step, we consider the dynamic decision-making problem to be used for downstream tasks at time t. The situation will turn to an energy-oriented case, when action goes to -1. At the same time, the situation will turn to a throughput-oriented case, when action goes to 1 in terms of the chosen step size δ value. In the case where the δ value is small, the system will take more resources (i.e., energy and time) to calculate the optimal value, and each action change may only produce small reward gains. Thus, we require the δ value selected to

be at most 0.5, and greater than 0.15 to prevent the void and eliminate redundant calculations.

Specifically, we obtain the state s in the MDP problem using information about the actions a with a given time period t. The state s thus contains the stored previous prediction f results, and the remaining query budget on both energy-oriented as well as throughput-oriented cases. The state will also record the time when the potential forwarding drone takes-off and starts the task. Thus, the state s for a given time period t can be formalized as follows: $s^t = [\theta_t, f_{\text{previous}}, f_{(\theta-\delta,\theta+\delta)}, T-t]$.

The reward function is given to minimize the cost of drone-based flight energy ϵ and obstacle-awareness recovery time τ given in Equation (11). Consequently, we considered the reward function to be the same as in Equation (11), with respect to the energy consumption on a single drone and considering theoretical guarantees on packet delivery in obstacle situations. Thus, we can ultimately define the reward function as follows:

$$R^t(s^t, a^t) = -\alpha \cdot \underbrace{\epsilon(f^t, \hat{f}^t)}_{\text{residual energy}} - \beta \cdot \underbrace{\tau(cost(a^t))}_{\text{obstacle}} \tag{19}$$

Having defined the obstacle-aware drone delivery scenario as an MDP, we can evaluate the overall performance by minimizing the expected total reward the system achieves. In this context, we can state: Given the choice of optimized θ values for highest gains of f, a finite horizon of T time slot, and an MDP problem M, find the optimal routing policy $\pi : s \rightarrow a$ that maximizes the expected cumulative reward R:

$$\pi \in \arg\max \mathbb{E}\Big(\sum_{T} R(s,a)\Big) \tag{20}$$

Although we framed this problem as an MDP, it is not easy to apply conventional techniques such as dynamic programming for solving the problem. This is because, many of the aspects of this problem are hard to analytically characterize, especially the dynamics of the sensory input stream (e.g., GPS, obstacle, energy). This motivates our integration of a soft optimal solution that uses a model-free RL technique. The reason to use such a technique is as follows: it is capable of learning optimal discrete policies based solely on the features included in the state, and avoids the need to predict the future states (as considered in HGPF). Specifically, we use the state-of-the-art Q-learning algorithm [65], which is easy to deploy, efficient to evaluate in terms of dynamics, and is amenable to effectively perform optimization-based action selection.

VII. Performance Evaluation

In this section, we first discuss our experiments setup, then we discuss the baseline approaches used for comparison, and finally we present the detailed performance results.

A. Experiment Setup

We initialized a simulation environment using 3D building models in the ns-3 simulator based on a set of rural and metropolitan road maps. The simulation script randomly generates task locations in the range of the simulated DRM scene. The simulator scripts can change the density of the buildings to simulate various DRM scenarios i.e., from rural areas (with the average height $< 50\,\mathrm{m}$) to metropolitan areas (with the average height $< 100\,\mathrm{m}$), with different building obstacle densities. Table III shows the simulation settings including both the application-side and network-side settings. The flying altitudes vary from $50\,\mathrm{m}$ for *delivery drones* and *monitoring drones*, and $100\,\mathrm{m}$ for *map drones*. In terms of the drones' parameters, we considered uniform speed of either $0\,\mathrm{m/s}$ or $20\,\mathrm{m/s}$ depending on the circumstances when a drone is assigned to be hovering or in motion, with the various payloads of $2\,\mathrm{kg}$, $7\,\mathrm{kg}$, and $10\,\mathrm{kg}$ for *delivery drones*, *monitoring drones*, and *map drones*, respectively.

Before starting the simulation experiments, we first evaluate the network parameters in realistic experiments between real drones and flights. The parameters are listed in Table III under network settings. To ensure the accuracy, security and speed, we utilize RUDP and QUIC for the video data transmission. More specifically, for the delivery drone and monitoring drone, we choose QUIC as the transport protocol, and for the map drone, we choose RUDP. The reason for why we choose QUIC on small devices is that QUIC can provide encryption by default during streaming, and provides high-speed connections even in cases where there are network failures or links experience low throughputs [66], [67]. On the other hand, for the map drone, we choose the RUDP for delivering continuous video streaming without the need for data encryption.

For the choice of the delivery drones and monitoring drones, we considered realistic flights pertaining to off-the-shelf drones i.e., the three DJI Mavic drones in our study. Each of them, is independently deployed in regions of the emulated disaster field. For the map drones, we assume settings from a commercial UAV, i.e., the DJI Matrice 600 equipped with a multispectral camera RedEdge (Micasense), an RGB camera (Sony a6300), a thermal image ICI 8640P (ICI company), a high-performance Real-Time Kinematic (RTK)-GPS receiver, and a light sensor. Each drone is assumed to be controlled by Android tablets running *SoftwarePilot* to manage all the drone flights. The tablets are connected via a $1\,\mathrm{GB/s}$ router to our GCS. In our experiments, the GCS is a Lenovo Thinkpad T470 computer running Ubuntu 18.04, which has been used to plan each mission, dispatch swarm members, and aggregate swarm results. We also provisioned a custom desktop PC with an Intel Xeon processor and Nvidia 2070TI GPU for offline RL purposes. Transmission data types are categorized as video stream data and text-based stream data. In most cases, video stream data is transmitted from the drone to the ground, and text-based stream data is transmitted oppositely.

For the wind parameters, we generate the global wind speeds at random among three distinct values, i.e., $0\,\mathrm{m/s}$, $5\,\mathrm{m/s}$, and $10\,\mathrm{m/s}$, which correspond to the usual wind speed according to the weather forecast [68]. Global wind direction is generated according to the uniform distribution from 0° to 359°. Although official weather stations report the recorded wind speed and direction with a granularity of $1\,\mathrm{hour}$ [69], in this article the experiment time-slot is $15\,\mathrm{mins}$, and within each time-slot, the global wind directions and speed will be kept the same. The wind CFD model will create local wind properties specifically for a height of $50\,\mathrm{m}$ and $100\,\mathrm{m}$. The changes of the wind above these two heights will be ignored.

Figure 7 shows an example of the two types of experiments (i.e., rural and metropolitan experiments).

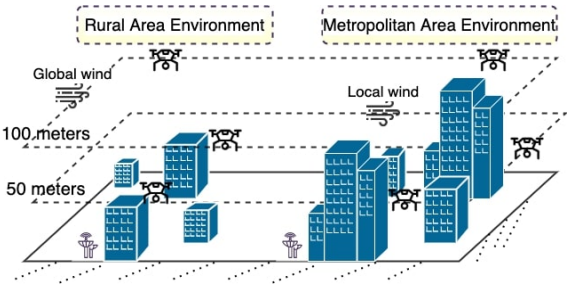


Fig. 7. Rural and Metropolitan experiment types illustration showing the differences in obstacle heights and wind conditions.

To run the experiments in a reproducible and reliable testbed, we leverage a trace-based drone-edge simulation platform on top of ns-3 [17]. This platform integrates simulation of both drones and networks for DRM scenarios, and provides flexibility in adding plugins, e.g., changing the mobility model of the drones, adding multi-sensor simulations, and applying realistic map interfaces. As mentioned in Section IV-C, the physical communication channel between the drones follows a simplistic "disk model", meaning that $100\%$of the communications succeed if the distance between the two nodes is $\leq R$, and 0% succeed if the distance $> R$. We set $R = 300\,\mathrm{m}$ according to the guidance in [70], [71]. To this end, the transmission range for A2A and A2G links is selected between 50–$250\,\mathrm{m}$ to avoid to beyond the line of sight for each drone.

To achieve a real-time energy consumption ϵ, we provide CFD models to mark out the various wind features on different Cartesian coordinates (with the precision of $1\,\mathrm{m}^2$), and calculate the energy consumption on that coordinate based on the wind model we provided in Section IV-A. After all the calculations, we get the residual energy of the specific drone in that location, and then apply our packet forwarding algorithms for the reestablishment of communications. To this end, in the ns-3 simulator, we do not directly use the original CFD model and only use the energy consumption from that CFD model. For each setting, we run the experiments in: (i) 5 different transmission settings ranging from 50–$250\,\mathrm{m}$, (ii) various number of drones ranging from 10–30, and (iii) various density percentages of obstacles in DRM ranging from 0–60%. Settings are summarized in Table III.

TABLE III
ENVIRONMENT SETTINGS USED IN THE SIMULATION EXPERIMENTS

Application Settings		Network Settings	
Number of drones:	10–30	Trans Protocol:	RUDP, QUIC
Disaster area:	7–10 km	Bit rate:	6 Mbps
Obstacle height:	$\leq$ 50–100 m	Tx power:	32–48 dBm
Transmission range:	50–250 m	Tx/Rx gain:	3 dB
Simulation time:	1000–3000 s	WiFi protocol	802.11 ax
Avg. drone speed:	10 m/s	Modulation:	OFDM
Prop. Model:	TWO RAY	Data rate:	65 Mbps

For evaluating our algorithms in contrast with state-of-the-art energy-aware or location-aware packet forwarding algorithms. In this context, we used the following performance metrics in our simulation environment:

Packet Delivery Ratio: Defined by the ratio of the packets that are successfully delivered to the target. A higher packet delivery ratio means a more reliable network connection.
End-to-end Delay: Calculated as the sum of the link re-establishment duration when a drone carries its generated data, and the data transmission time to the GCS. A lower end-to-end delay implies better performance on network links.
Energy Usage Proportion: For the energy measurement, we check the distribution of the average residual energy percentage used for the networking procedure of all drones in the DRM scenario. Given that the total energy usage varies for each kind of drone, we calculate the energy usage proportion in order to normalize the metrics. A lower energy usage proportion means the drone could use more energy for its task.

B. Baseline Solution Approaches

To design a better packet forwarding approach for the multi-drone coordination problem, we compare both HGPF and IRLPF algorithms with several existing state-of-the-art algorithms. All experiments use the same settings previously detailed. We used three different baselines (BLs) approaches detailed below based on their different perspectives:
BL 1 – TQNGPSR [39]: Similar to how we considered GPSR with SPF in Section VI, we selected one variant of the GPSR based on a location-aided and traffic-awareness approach, which is called TQNGPSR. Compared to the traditional GPSR, this new variant enforces a traffic balancing strategy using the congestion information of neighbors, and evaluates the quality of a wireless link by applying a Q-learning algorithm. Both TQNGPSR and our proposed approach use RL when applied in experiments on drone networks. However, TQNGPSR does not provide solutions when wind and obstacles could directly influence the drone communications, whereas in our proposed solution we considered both the wind and obstacle factors.
BL 2 – SPIDER [44]: This algorithm solves the packet forwarding problem by considering edge devices' energy consumption as well as presence of obstacles, in a manner that is very close to our system formulation. However, in terms of obstacle awareness, our approach considers three dimensional obstacles with consideration of the Fresnel Zone on communication blockage, which is not involved in the design of the SPIDER algorithm.
BL 3 – LADTR [43]: It consists of a location-aided drone packet forwarding algorithm design. One difference between LADTR and our location prediction model is that LADTR only considers GMM, while ours considers various drone mobility models and uses RL to enhance the location prediction precision. In addition, LADTR does not consider obstacles that may block the communication, whereas we consider obstacles in our DRM packet forwarding solutions.

C. Performance Results

In this section, we evaluate the performances under two metrics, i.e., network quality and energy usage. In addition, two types of experiment environments are considered, i.e., with an average building height of $h \leq 50$ meters (*rural area*),

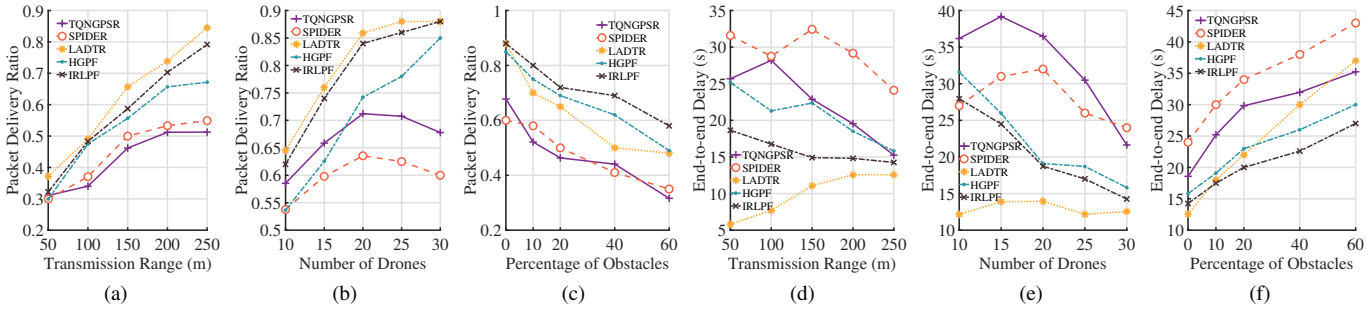


Fig. 8. Performance metrics on packet delivery ratio (a,b,c) and average end-to-end delay (d,e,f) in terms of varying: 1) transmission range, 2) amount of drones, and 3) percentage of physical obstacles in a realistic DRM scenario (with $h \leq 50\,\mathrm{m}$).

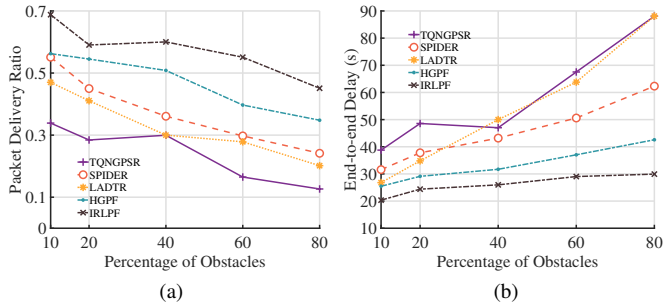


Fig. 9. Performance results based on packet delivery ratio (a), and average end-to-end delay (b), under conditions of various percentage of physical obstacles in a realistic DRM scenario (with $50 < h \leq 100\,\mathrm{m}$).

and with an average building height of $50 < h \leq 100$ meters (*metropolitan area*).

Network Quality: Figure 8 shows the results on the network performance for all the algorithms in a rural area environment. Considering various transmission ranges of the nodes (Figures 8a and 8d), we can see that all the algorithms perform better in larger range cases. This is because a large range provides more contact opportunities for creating transmission links, which will in turn increase the packet delivery ratio and decrease the delay. HGPF and IRLPF outperform TQNGPSR and SPIDER under all transmission settings. LADTR performs better than other state-of-the-art strategies due to its use of ferrying drones that use up all of their energy on networking procedures instead of focusing on their own tasks. On the contrary, IRLPF which only uses residual energy on transmission, achieves $\approx 90\%$ of the performance of LADTR. This result shows that the performance of IRLPF is comparable to the LADTR, which although it has a better performance, has the problem of energy wastage in drone ferrying. From Figure 8b it is possible to observe that when the number of drones increases, SPIDER and GPSR do not provide better results due to the unpredictable drone positions and unplanned forwarding node selection. By combining Figures 8d and 8e, we can state that LADTR's results do not provide lower end-to-end delay when the transmission range increases or when more drones are added. This is because LADTR follows a stateful setting rather than a stateless one, which is used in other approaches. Thereby, the forwarding node for transmission will not change once selected, and the delay for LADTR remains unchanged once a network connection is established.

Both the above experiments do not consider any obstacles in the DRM application scenario in a given rural environment setting. However, if we consider obstacle blockage of the connections, both HGPF and IRLPF outperform the other baseline approaches. Figures 8f and 8c depict the network performance with the increase in the percentage of obstacles in the DRM scenario. Note that, under the conditions in which the average obstacle building heights is less than the average drone swarm flight height, there still exist continuous interruption of networks due to the fixed ground control station and the existence of the Fresnel zone. In Figures 8f and 8c, it is remarkable to observe that - with the increase of the density of the obstacles, the network quality decreases, and this aspect influences all the algorithms. Given that we considered algorithms that can deal with obstacles, other solutions without obstacle awareness (e.g., LADTR and TQNGPSR) experience poor performance than HGPF or IRLPF. Although LADTR can achieve similar end-to-end delay under low obstacle settings, this condition is not always promising in complex metropolitan DRM scenarios with high building obstacle densities as seen in Figure 9. To conclude, without considering obstacles, LADTR achieves better network performance due to its stateful connection settings, while our solutions can achieve $\approx$ 81–90% of the LADTR performance. However, LADTR is not robust considering that obstacle blockage is very common in DRM scenarios even under rural environment settings, where the average obstacle objects height is less than or equal to $50\,\mathrm{m}$. In such situations, an increase of $\approx$ 14–38% of network performance is obtained by using our baseline approaches.

The second experiment considers the situation when the height of the obstacle is between $50\,\mathrm{m}$ and $100\,\mathrm{m}$. We assume this setup as a metropolitan area according to the average roof height in major cities around the world [72]. Figure 9 shows the results on the network performance for all the algorithms in various metropolitan areas. We simulate a default transmission range of $150\,\mathrm{m}$ for each drone, and employ a total of 30 drones to run this set of experiments. From the results, we can observe that generally the network quality decreases significantly, thus influencing all the algorithms. This trend can be especially observed when density of the obstacles increases (compared with the rural area environments). This is because - unlike the experiments in a rural area environment, a network interruption situation happens frequently according to the uncertain conditions when the A2A link may be blocked by

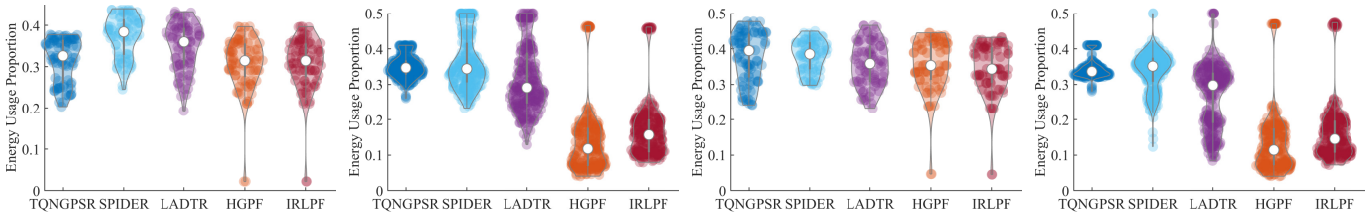


Fig. 10. Energy usage proportion for communication in 40% of obstacles (with $h \leq 50\,\mathrm{m}$)) with four different experiments with varying transmission range and drone amounts for the cases: a) 50 m, 10 drones; b) 50 m, 30 drones; c) 250 m, 10 drones, and d) 250 m, 30 drones.

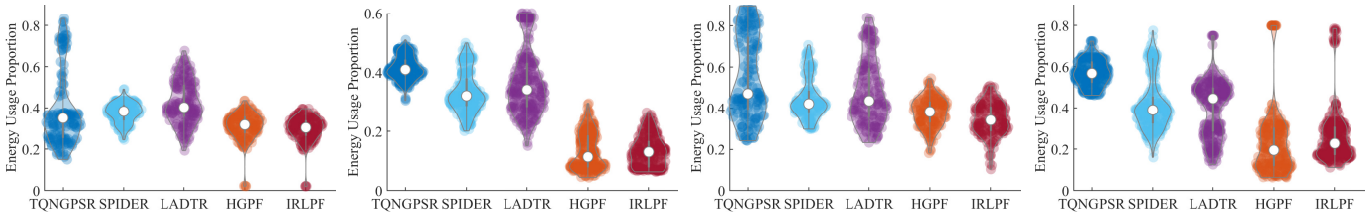


Fig. 11. Energy usage proportion for communication in 40% of obstacles (with $50 < h \leq 100$ meters) with four different experiments with varying transmission range and drone amounts for the cases: a) 50 m, 10 drones; b) 50 m, 30 drones; c) 250 m, 10 drones, and d) 250 m, 30 drones.

the skyscrapers (with $\approx 60\%$ higher chance based on our trace-based simulator). As discussed earlier, LADTR can no longer achieve similar end-to-end delay under low obstacle density circumstances as it does in rural environment experiments. The reason is due to the election of the ferrying drone in LADTR that requires the knowledge of all the neighbour drones' positions. However, such information is not always obtainable if any obstacles are present between the drones. Similar suboptimal results are given by TQNGPSR due to the fact that unreliable congestion information is generated before the network interruption. This information will result in the Q-learning procedure given by the TQNGPSR algorithm trapping into the local optimizer according to the QoS requirements without querying for the alternative route. On the contrary, since the SPIDER algorithm applies shortest path approximation procedures with respect to the obstacle blockage regardless of the A2A or A2G link, it achieves $\approx$ 85–95% of performance compared with rural environment experiments. By only comparing HGPF and IRLPF, we can observe that IRLPF always performs better than HGPF in both rural and metropolitan settings. On the other hand, HGPF provides a network performance of $\approx$ 12–18% lower than IRLPF while comparing with other baseline approaches, while other baseline approaches can only achieve $\approx 75\%$ of the network performance of IRLPF at the most. When considering the obstacle densities, we can derive the same conclusion as we did in the rural experiments. In summary, when considering obstacles that may block the A2A communication, our solutions can achieve an average gain of $\approx 25\%$ compared with the other baseline obstacle awareness approaches.

Energy Usage: The violin distribution diagrams shown in Figures 10 and 11 depict the results concerning the four experiments with varying transmission ranges (50 m, 250 m), 40% of obstacle settings, and varying number of drones (10 drones, 30 drones), under two types of obstacle heights (either within 50 m or between 50 to 100 m). Notice that, under these circumstances where the average obstacle height is less than 50 m, the energy consumption calculation will only consider the global wind properties. This directly comes from the assumption that the height of the drone swarm is always higher than the physical obstacles. From all four experiments in Figure 10, we can observe that both HGPF and IRLPF save $\approx$ 23–54% of energy. This is due to the fact that the establishment of network connectivity is not performed if there are no transmission tasks, and only residual energy is used to establish network links for generating forwarding drone candidates. Moreover, as we can observe from Figures 10b and 10d, HGPF saves more energy in larger drone number situations. Due to the fact that IRLPF presents more forwarding drone candidates by intelligently scheduling the task execution in presence of increasing number of drones, an increased energy consumption occurs while simultaneously providing guarantees on better network connection quality in comparison with HGPF. In terms of the scenario when the energy consumption is calculated based on the local wind according to the wind CFD model, we conclude that both HGPF and IRLPF save $\approx$ 15–76% of energy. This comes by utilizing the wind CFD model where we can achieve more precise wind directions and speeds information than utilizing the static global wind model. Those wind properties, provided by the wind CFD model, are given in real-time to update the original energy consumption for each in-flight drone. As a result, the required energy consumption is decreased when there is a need to re-establish the A2A or A2G links. In addition, as we can observe from Figures 10 and 11, the baseline methods perform much worse in a metropolitan environment than in a rural environment due to the lack of strategies to handle the uncertainty of the local wind change influenced by the physical obstacle buildings.

Execution Time: Figure 12 shows the results in terms of experimental running time of all the compared packet forwarding algorithms. The y-axis (logarithmic scale in this

case) represents the average CPU time (in seconds) for each drone on-board. Specifically, Figure 12a describes several experiments with various number of drones (10-30) under a DRM scenario with no obstacles, while Figure 12b shows the CPU time with a group of 20 drones under various percentage of obstacles in the environment.

Under the simplest scenario with no obstacles, our proposed algorithms HGPF and IRLPF execute quickly, varying from 0.2–1.0 s when the number of drones varies from 10 to 30. Recalling that these algorithms also exhibit good performance in terms of latency and throughput, we can say that both HGPF and IRLPF can be easily performed on board on off-the-shelf drones. Also the TQNGPSR algorithm quickly executes as the previous two. LADTR seems to have a constant experimental running time regardless of the number of drones (i.e., ≈ 4 s, while the worst algorithm here is SPIDER, whose CPU time rapidly grows when the number of employed drones increases (i.e., up to $\approx$ 16 s with 30 drones). Obviously, this makes the use of these last two algorithms unfeasible on-board the drones. On the other hand, by fixing the number of drones to 20 and varying the number of obstacles, we see that our two HGPF and IRLPF outperform TQNGPSR in terms of CPU time. In fact, in a relatively sparse environment, with 20–40% of obstacles, our proposed algorithms take < 3 s, while TQNGPSR takes $\approx 5\times$ more. When the environment changes to dense, i.e., with 60–80% of obstacles, HGPF and IRLPF take < 15 s, while the others take up to 1 min. Hence, in a DRM scenario that is full of obstacles, algorithms such as SPIDER, LADTR, and TQNGPSR are not particularly suitable to be performed on off-the-shelf drones.

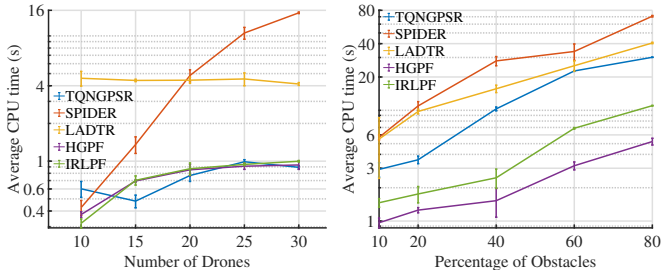


Fig. 12. CPU time (seconds) on average per embedded device (on each drone) in terms of: (a) various number of drones (with no obstacles), (b) and percentage of obstacles (with 20 drones).

Discussion of Results: Herein, we discuss the choice for the most appropriate packet forwarding algorithm in realistic DRM scenarios in terms of the rural area and the metropolitan area. First, if drones have no restrictions on take-off time, IRLPF is the most suitable because it will find the optimal task schedule. Consequently, it provides better network performance compared to HGPF, while also saving energy. Second, if drones need to be scheduled in advance for accomplishing specific tasks in accordance with a restricted time table while saving energy, HGPF is preferred. This is because HGPF can provide acceptable network quality performance with considerable energy savings. Third, as the number of drones increases, both HGPF or IRLPF are acceptable in terms of conserving energy usage, irrespective of the number of drones. Consequently, they can provide a pertinent solution for DRM application operations involving large number of drones in the presence of obstacles. Lastly, in DRM scenarios where there are no obstacle blockages and energy consumption is not an issue, LADTR will be the best one in terms of network performance. However, the links generated by LADTR are unstable and unreliable when they are blocked due to obstacles. Thus, in cases where the environmental condition of a DRM scenario is unknown, HGPF or IRLPF algorithm are more pertinent for DRM application operations.

VIII. Conclusion

In this paper, we proposed a novel *environmentally-aware packet forwarding scheme* for multi-drone cooperation in various applications related to disaster response management (DRM). We also considered dynamics of wind conditions integrated via a wind model into the obstacles formulation. Our contributions advance current knowledge on the design and development of location-aided packet forwarding for FANETs with consideration of geographical obstacles blockage, and energy efficiency for coordinated drone flights in DRM application scenarios. Specifically, we devised an accurate drone location position prediction method using the deep RL to suit multi-drone DRM application scenarios.

We also proposed two different algorithms, i.e., HGPF and IRLPF algorithms that utilized the drone location predictions for improving the network connectivity, while also providing better energy efficiency compared to state-of-the-art approaches such as TQNGPSR, SPIDER, and LADTR. Specifically, our proposed schemes outperformed the state-of-the-art approaches in terms of network connectivity performance (e.g., packet delivery ratio, end-to-end delay) and energy usage (i.e., energy usage proportion for communication) in experiments with a trace-based drone-edge simulation platform featuring different transmission ranges, amount of drones, and obstacle density. In terms of the algorithm execution time, our HGPF and IRLPF algorithms are also comparable with RL-based multi-drone routing algorithms and outperform all other state-of-the-art approaches in the scenario where obstacles exist.

In practice, our proposed architecture can be potentially implemented in real-world DRM scenarios due to its relative simplicity. Nowadays, commercial top-tier drones have impressive characteristics in terms of flight time endurance, computational edge capabilities, additional weight to be carried, and extremely powerful cameras. These aspects make them suitable for being employed in assisting a team of rescuers in a DRM scenario. Moreover, our proposed packet forwarding algorithms (that deal with wind and obstacles) are effective and lightweight, and can be easily run on the edge, or even on-board of drones.

Future work can consider DRM applications that require sustained video throughput in A2G links, while also avoiding disruptions in A2A and A2G connectivity. Study of scenarios where wheeled vehicles assist the drones in DRM applications can also be considered in the design of extended multi-drone coordination and networking schemes.

REFERENCES

[1] P. M *et al.*, "Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges," *IEEE Sensors Journal*, 2020.

[2] R. Rumba and A. Nikitenko, "The wild west of drones: a review on autonomous-uav traffic-management," in *2020 Int. Conf. on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1317–1322.

[3] A. Khanda, F. Corò, F. Betti Sorbelli, C. M. Pinotti, and S. K. Das, "Efficient route selection for drone-based delivery under time-varying dynamics," in *2021 IEEE 18th Intl. Conf. on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2021, pp. 437–445.

[4] T. Calamoneri, F. Corò, and S. Mancini, "A realistic model to support rescue operations after an earthquake via uavs," *IEEE Access*, vol. 10, pp. 6109–6125, 2022.

[5] F. Betti Sorbelli, F. Corò, S. K. Das, L. Palazzetti, and C. M. Pinotti, "Drone-based optimal and heuristic orienteering algorithms towards bug detection in orchards," in *18th Int. Conf. on Distr. Computing in Sensor Systems (DCOSS)*. IEEE, 2022.

[6] S. Calandrillo, J. Oh, and A. Webb, "Deadly drones: Why faa regulations miss the mark on drone safety," *Stan. Tech. L. Rev.*, vol. 23, p. 182, 2020.

[7] A. Khochare, Y. Simmhan, F. B. Sorbelli, and S. K. Das, "Heuristic algorithms for co-scheduling of edge analytics and routes for uav fleet missions," in *IEEE INFOCOM*, 2021.

[8] C. Qu, R. Singh, A. E. Morel, F. Betti Sorbelli, P. Calyam, and S. K. Das, "Obstacle-aware and energy-efficient multi-drone coordination and networking for disaster response," in *17th Intl. Conf. on Network and Service Management (CNSM)*. IEEE, 2021, pp. 446–454.

[9] T. Nguyen and T.-C. Au, "Extending the range of delivery drones by exploratory learning of energy models," in *Autonomous Agents and MultiAgent Systems*. IFAAMAS, 2017, pp. 1658–1660.

[10] C. Qu, J. Boubin, D. Gafurov, J. Zhou, N. Aloysius, H. Nguyen, and P. Calyam, "Uav swarms in smart agriculture: Experiences and opportunities," *IEEE eScience*, 2022.

[11] B. Wang, J. Bao, L. Zhang, and Q. Sheng, "Uav autonomous path optimization simulation based on radar tracking prediction," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–8, 2018.

[12] C. Qu, P. Calyam, J. Yu, A. Vandanapu, O. Opeoluwa, K. Gao, S. Wang, R. Chastain, and K. Palaniappan, "Dronecoconet: Learning-based edge computation offloading and control networking for drone video analytics," *Future Generation Computer Systems*, vol. 125, pp. 247–262, 2021.

[13] C. Qu, R. Singh, A. Esquivel-Morel, and P. Calyam, "Learning-based multi-drone network edge orchestration for video analytics," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 1219–1228.

[14] F. Betti Sorbelli, F. Corò, S. K. Das, and C. M. Pinotti, "Energy-constrained delivery of goods with drones under varying wind conditions," *IEEE Trans. on Intelligent Transportation Systems*, 2020.

[15] F. Betti Sorbelli, C. M. Pinotti, S. Silvestri, and S. K. Das, "Measurement errors in range-based localization algorithms for *UAVs*: Analysis and experimentation," *IEEE Trans. on Mobile Computing*, 2021.

[16] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "Uav assistance paradigm: State-of-the-art in applications and challenges," *Journal of Network and Computer Applications*, vol. 166, p. 102706, 2020.

[17] C. Qu, A. E. Morel, D. Dahlquist, and P. Calyam, "Dronenet-sim: a learning-based trace simulation framework for control networking in drone video analytics," in *6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2020, pp. 1–6.

[18] 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.

[19] B. D. Song, H. Park, and K. Park, "Toward flexible and persistent uav service: Multi-period and multi-objective system design with task assignment for disaster management," *Expert Systems with Applications*, vol. 206, p. 117855, 2022.

[20] A. Khan, S. Gupta, and S. K. Gupta, "Unmanned aerial vehicle-enabled layered architecture based solution for disaster management," *Trans. on Emerging Telecommunications Technologies*, vol. 32, no. 12, p. e4370, 2021.

[21] V. Mayor, R. Estepa, A. Estepa, and G. Madinabeitia, "Deploying a reliable uav-aided communication service in disaster areas," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[22] X. Yu, C. Li, and G. G. Yen, "A knee-guided differential evolution algorithm for unmanned aerial vehicle path planning in disaster management," *Applied Soft Computing*, vol. 98, p. 106857, 2021.

[23] H. S. Munawar, F. Ullah, S. Qayyum, S. I. Khan, and M. Mojtahedi, "Uavs in disaster management: Application of integrated aerial imagery and convolutional neural network for flood detection," *Sustainability*, vol. 13, no. 14, p. 7547, 2021.

[24] A. Danjo, A. Murata, S. Hara, T. Matsuda, and F. Ono, "Construction of a temporary message collection system using a drone for refugees in a large-scale disaster," in *2020 IEEE 91st Vehicular Technology Conf. (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[25] B. Shishkov, S. Hristozov, and A. Verbraeck, "Improving resilience using drones for effective monitoring after disruptive events," in *9th Int. Conf. on Telecommunications and Remote Sensing*, 2020, pp. 38–43.

[26] D. Q. Tran, M. Park, D. Jung, and S. Park, "Damage-map estimation using uav images and deep learning algorithms for disaster management system," *Remote Sensing*, vol. 12, no. 24, p. 4169, 2020.

[27] J. Xu, K. Ota, and M. Dong, "Big data on the fly: Uav-mounted mobile edge computing for disaster management," *Trans. on Network Science and Engineering*, vol. 7, no. 4, pp. 2620–2630, 2020.

[28] W. Ejaz, A. Ahmed, A. Mushtaq, and M. Ibnkahla, "Energy-efficient task scheduling and physiological assessment in disaster management using uav-assisted networks," *Computer Communications*, vol. 155, pp. 150–157, 2020.

[29] I. D. Da Silva, C. Caillouet, and D. Coudert, "Optimizing fanet deployment for mobile sensor tracking in disaster management scenario," in *2021 Intl. Conf. on Information and Communication Technologies for Disaster Management (ICT-DM)*. IEEE, 2021, pp. 134–141.

[30] R. Masroor, M. Naeem, and W. Ejaz, "Efficient deployment of uavs for disaster management: A multi-criterion optimization approach," *Computer Communications*, vol. 177, pp. 185–194, 2021.

[31] R. Tanaka, G. Papadimitriou, S. C. Viswanath, C. Wang, E. Lyons, K. Thareja, C. Qu, A. Esquivel, E. Deelman, A. Mandal *et al.*, "Automating edge-to-cloud workflows for science: Traversing the edge-to-cloud continuum with pegasus," in *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2022, pp. 826–833.

[32] O. S. Oubbati, A. Lakas, and M. Guizani, "Multi-agent deep reinforcement learning for wireless-powered uav networks," *IEEE Internet of Things Journal*, 2022.

[33] O. A. Saraereh, A. Alsaraira, I. Khan, and P. Uthansakul, "Performance evaluation of uav-enabled lora networks for disaster management applications," *Sensors*, vol. 20, no. 8, p. 2396, 2020.

[34] M. Biomo *et al.*, "Routing in unmanned aerial ad hoc networks: A recovery strategy for greedy geographic forwarding failure," in *2014 Wireless Comm. and Networking Conf.* IEEE, 2014, pp. 2236–2241.

[35] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual Int. Conf. on Mobile computing and networking*, 2000, pp. 243–254.

[36] M. Hyland, B. E. Mullins, R. O. Baldwin, and M. A. Temple, "Simulation-based performance evaluation of mobile ad hoc routing protocols in a swarm of unmanned aerial vehicles," in *AINAW'07*, vol. 2. IEEE, 2007, pp. 249–256.

[37] R. Shirani *et al.*, "The performance of greedy geographic forwarding in unmanned aeronautical ad-hoc networks," in *Ninth Annual Communication Networks and Services Research Conf.* IEEE, 2011, pp. 161–166.

[38] N. Lyu, G. Song, B. Yang, and Y. Cheng, "Qngpsr: a q-network enhanced geographic ad-hoc routing protocol based on gpsr," in *2018 IEEE 88th Vehicular Technology Conf. (VTC-Fall)*. IEEE, 2018, pp. 1–6.

[39] Y.-n. Chen *et al.*, "A traffic-aware q-network enhanced routing protocol based on gpsr for unmanned aerial vehicle ad-hoc networks," *Frontiers*, vol. 21, no. 9, pp. 1308–1320, 2020.

[40] A. Saif, K. Dimyati, K. A. Noordin, S. H. Alsamhi, and A. Hawbani, "Multi-uav and sar collaboration model for disaster management in b5g networks," *Internet Technology Letters*, p. e310, 2021.

[41] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Communications and control for wireless drone-based antenna array," *IEEE Trans. on Communications*, vol. 67, no. 1, pp. 820–834, 2018.

[42] G. Bielsa *et al.*, "Performance assessment of off-the-shelf mm wave radios for drone communications," in *Symposium on A World of Wireless, Mobile and Multimedia Networks*. IEEE, 2019, pp. 1–3.

[43] T. Akram, M. Awais, R. Naqvi, A. Ahmed, and M. Naeem, "Multicriteria uav base stations placement for disaster management," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3475–3482, 2020.

[44] H. Trinh, P. Calyam, D. Chemodanov, S. Yao, Q. Lei, F. Gao, and K. Palaniappan, "Energy-aware mobile edge computing and routing for low-latency visual data processing," *IEEE Trans. on Multimedia*, vol. 20, no. 10, pp. 2562–2577, 2018.

[45] D. Chemodanov, F. Esposito, A. Sukhov, P. Calyam, H. Trinh, and Z. Oraibi, "Agra: Ai-augmented geographic routing approach for iot-based incident-supporting applications," *Future Generation Computer Systems*, vol. 92, pp. 1051–1065, 2019.

[46] S. M. S. Mohd Daud, M. Y. P. Mohd Yusof, C. C. Heo, L. S. Khoo, M. K. Chainchel Singh, M. S. Mahmood, and H. Nawawi, "Applications of drone in disaster management: A scoping review," *Science & Justice*, vol. 62, no. 1, pp. 30–42, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1355030621001477

[47] R. Kellermann, T. Biehle, and L. Fischer, "Drones for parcel and passenger transportation: A literature review," *Transportation Research Interdisciplinary Perspectives*, vol. 4, p. 100088, 2020.

[48] M. B. Yassein, N. Alhuda *et al.*, "Flying ad-hoc networks: Routing protocols, mobility models, issues," *Int. Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 6, 2016.

[49] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun, "Fanet: Communication, mobility models and security issues," *Computer Networks*, vol. 163, p. 106877, 2019.

[50] A. Bujari, C. T. Calafate, J.-C. Cano *et al.*, "Flying ad-hoc network application scenarios and mobility models," *Int. Journal of Distributed Sensor Networks*, vol. 13, no. 10, p. 0, 2017.

[51] F. Betti Sorbelli, F. Corò, L. Palazzetti, C. M. Pinotti, and G. Rigoni, "How the wind can be leveraged for saving energy in a truck-drone delivery system," *IEEE Trans. on Intelligent Transportation Systems*, 2023.

[52] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nature communications*, vol. 9, no. 1, pp. 1–13, 2018.

[53] M.-D. Hua *et al.*, "A control approach for thrust-propelled underactuated vehicles and its application to vtol drones," *IEEE Trans. on Automatic Control*, vol. 54, no. 8, pp. 1837–1853, 2009.

[54] S. Driessens and P. E. Pounds, "Towards a more efficient quadrotor configuration," in *2013 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. IEEE, 2013, pp. 1386–1392.

[55] W. Johnson, *Helicopter theory*. Courier Corporation, 2012.

[56] S. Al-Emadi and A. Al-Mohannadi, "Towards enhancement of network communication architectures and routing protocols for fanets: A survey," in *2020 3rd Int. Conf. on Advanced Communication Technologies and Networking (CommNet)*. IEEE, 2020, pp. 1–10.

[57] Simscale, "Simscale - academic cfd software," Jan 2022. [Online]. Available: https://www.simscale.com/academic-program/

[58] "Fresnel zone," Jan 2021. [Online]. Available: https://en.wikipedia.org/wiki/Fresnel_zone

[59] C. Luo, S. I. McClean, G. Parr *et al.*, "Uav position estimation and collision avoidance using the extended kalman filter," *IEEE Trans. on Vehicular Technology*, vol. 62, no. 6, pp. 2749–2762, 2013.

[60] P. Sujit and D. Ghose, "Search using multiple uavs with flight time constraints," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 491–509, 2004.

[61] C. Lochert, H. Hartenstein, J. Tian *et al.*, "A routing strategy for vehicular ad hoc networks in city environments," in *IV2003 Intelligent Vehicles Symposium*. IEEE, 2003, pp. 156–161.

[62] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, 1999, pp. 90–100.

[63] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "Ieee 802.11 s: the wlan mesh standard," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, 2010.

[64] J. Peng, P. Zhang, L. Zheng, and J. Tan, "Uav positioning based on multi-sensor fusion," *IEEE Access*, vol. 8, pp. 34 455–34 467, 2020.

[65] D. Tamagawa, E. Taniguchi, and T. Yamada, "Evaluating city logistics measures using a multi-agent model," *Procedia-Social and Behavioral Sciences*, vol. 2, no. 3, pp. 6002–6012, 2010.

[66] A. Langley *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the Conf. of the ACM special interest group on data communication*, 2017, pp. 183–196.

[67] M. Seufert, R. Schatz, N. Wehner, and P. Casas, "Quicker or not?-an empirical analysis of quic vs tcp for video streaming qoe provisioning," in *2019 22nd Conf. on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 2019, pp. 7–12.

[68] J. Thomson, E. D'Asaro, M. Cronin, W. Rogers, R. Harcourt, and A. Shcherbina, "Waves and the equilibrium range at ocean weather station p," *Journal of Geophysical Research: Oceans*, vol. 118, no. 11, pp. 5951–5962, 2013.

[69] N. N. C. for Environmental Information, "Office for coastal management," 2022. [Online]. Available: https://www.fisheries.noaa.gov/inport/item/66216

[70] A. S. Abdalla and V. Marojevic, "Communications standards for unmanned aircraft systems: The 3gpp perspective and research drivers," *IEEE Communications Standards Magazine*, vol. 5, no. 1, pp. 70–77, 2021.

[71] S. J. Maeng, I. Güvenç, M. Sichitiu, B. A. Floyd, R. Dutta, T. Zajkowski, Ö. Özdemir, and M. J. Mushi, "National radio dynamic zone concept with autonomous aerial and ground spectrum sensors," *arXiv preprint arXiv:2203.09014*, 2022.

[72] Wikipedia, "List of cities with the most skyscrapers," Jan 2022. [Online]. Available: https://en.wikipedia.org/wiki/List_of_cities_with_the_most_skyscrapers

Chengyi Qu received his B.S. degree from the Department of Software Engineering at Northeast University, China in 2016. He is currently a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia. His current research interests include distributed and cloud computing, edge computing, ad-hoc networks and drone video analytics.

Francesco Betti Sorbelli is currently an assistant professor at the University of Perugia, Italy. He received the Bachelor and Master degrees *cum laude* in Computer Science from the University of Perugia in 2007 and 2010, respectively, and his Ph.D. in Computer Science from the University of Florence, Italy, in 2018. He was a postdoctoral researcher at the Missouri University of S&T, USA, in 2020, and at the University of Perugia in 2018, 2021–2022. His research interests include algorithms design, combinatorial optimization, unmanned vehicles.

Rounak Singh received his B.Tech. degree in Electronics and Communication Engineering from G.I.T.A.M Deemed to be University, Hyderabad, India in 2019. He was a graduate research assistant at University of Missouri-Columbia and received his M.S degree in Electrical Engineering in 2021. His research interests include: machine learning, deep learning, reinforcement learning, computer vision and cloud computing.

Prasad Calyam received his MS and PhD degrees from the Department of Electrical and Computer Engineering at The Ohio State University in 2002 and 2007, respectively. He is currently an Associate Professor in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia. His current research interests include distributed and cloud computing, computer networking, and cyber security. He is a Senior Member of IEEE.

Sajal K. Das is a professor of computer science and Daniel St. Clair Endowed Chair at Missouri University of Science and Technology. His research interests include wireless sensor networks, mobile and pervasive computing, cyber-physical systems and IoT, smart environments, cloud computing, cyber security, and social networks. He serves as the founding Editor-in-Chief of Elsevier's Pervasive and Mobile Computing journal, and as Associate Editor of several journals. He is an IEEE Fellow.