Dynamic Super Round-Based Distributed Task Scheduling for UAV Networks

Halder, Subir; Ghosal, Amrita; Conti, Mauro

DOI
10.1109/TWC.2022.3200366

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Wireless Communications

Dynamic Super Round-Based Distributed Task Scheduling for UAV Networks

Subir Halder, Amrita Ghosal, and Mauro Conti, *Fellow, IEEE*

***Abstract*— Networks of Unmanned Aerial Vehicles (UAVs) are emerging in many application domains, e.g., military surveillance. To perform collaborative tasks, the involved UAVs exchange several types of information, e.g., sensor data and commands. The major question here is how to schedule the tasks under dynamic traffic flows to provide network services. Existing solutions use the Round-Robin Strategy (RRS), where the tasks are scheduled statistically by dividing the time into fixed-length rounds. However, the RRS wastes significant network and device resources due to task scheduling in each round. This paper proposes DROVE – a novel clustering approach that allows the UAVs for dynamic task scheduling. However, determining the task scheduling is crucial, as it significantly affects several network parameters, e.g., throughput. Therefore, we devise the problem of distributed task scheduling under dynamic traffic flow scenarios to optimize the throughput. We propose a clustering task scheduling algorithm to serve dynamic traffic flows. Particularly, we integrate the dynamic traffic flows into the Lyapunov drift analysis framework, and determine the throughput optimality of our proposed scheduling algorithm. We perform extensive simulations to validate the effectiveness of DROVE. The results show that DROVE outperforms the state-of-the-art solutions in terms of energy consumption, clustering overhead, throughput, end-to-end delay, flow success rate and packet drop rate.**

***Index Terms*— Clustering, data traffic-level dynamics, energy-efficiency, scheduling design, UAV networks.**

I. Introduction

IN RECENT years, the wireless transmission of video data captured by Unmanned Aerial Vehicles (UAVs) has started to receive attention in various cyber-physical systems, including surveillance, public safety, and traffic monitoring [1]. In a typical UAV network, a flying UAV, also known as drone, equipped with a camera captures high-quality images and videos in real-time from the sky for users located at the Ground Station (GS). At present, a flying UAV equipped with a camera is commercially available. For example, DJI Phantom 4 UAV can rise at 6 m/s, fly at 72 km/h, swerve at 250°/s and deliver 2K real-time images and videos [2]. The main advantage of a UAV network is that the deployment of UAVs can be agile and re-configurable due to flexible mobility and the ability to perform tasks autonomously and collaboratively [3]. As a matter of fact, in the UAV network, the involved UAVs cooperate with each other and manage their actions through the exchange of information, e.g., status information, sensor data, and commands.

Recently, efficient placement of UAVs has attracted significant research attention to address challenges like the optimal deployment considering energy efficiency [4], [5], deployment [2], [6], and throughput maximization [7], [8]. Besides, the UAV path planning considering different communication and network constraints is rigorously examined in [9] and [10]. Further, recent research has shown the feasibility of UAV-based small cells [11]. To enhance the network lifespan or even provide consistent service during critical missions, battery charging approaches have been designed. These approaches include charging of battery at night through a high energy laser beam [12] or wireless power transfer [13] and equipping the UAVs with solar panels [14]. However, to the best of our knowledge, none of the existing works focus on the design of distributed task scheduling techniques for the UAVs in response to the dynamics of real-time tracking applications, where objects dynamically arrive and move around the network. To this end, in this work, we focus on an important problem of efficiently scheduling the data traffic of the UAVs to track moving objects in real-time across many geographical areas.

Despite the widespread applications of the UAVs, scheduling data traffic under dynamic scenarios tender many new challenges. Specifically, the lifespan and performance of the UAV networks are primarily limited by the on-board energy, which is limited due to the UAV's size and weight constraints. Therefore, energy-efficient task scheduling for maximizing the information bits per unit energy consumption of the UAV is of utmost importance. Recently, task scheduling under dynamic application scenarios in the UAV networks has received considerable attention, as it is significantly different from the terrestrial networks [15]. In [16], Kong *et al.* revealed that the traditional queue-length based MaxWeight task scheduling has failed to deliver optimal throughput. Subsequently, many novel scheduling techniques have been designed recently in the

Manuscript received 2 January 2022; revised 16 May 2022; accepted 10 August 2022. Date of publication 29 August 2022; date of current version 13 February 2023. The associate editor coordinating the review of this article and approving it for publication was L.-C. Wang. *(Corresponding author: Subir Halder.)*

Subir Halder is with the Department of Electronic and Computer Engineering, University of Limerick, Limerick, V94 T9PX Ireland (e-mail: subir.halder@ul.ie).

Amrita Ghosal is with the Department of Computer Science and Information Systems, University of Limerick, Limerick, V94 T9PX Ireland (e-mail: amrita.ghosal@ul.ie).

Mauro Conti is with the Department of Mathematics, University of Padua, 35121 Padua, Italy, and also with the Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft, 2628 CD Delft, Netherlands (e-mail: conti@math.unipd.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2022.3200366.

Digital Object Identifier 10.1109/TWC.2022.3200366

context of routing protocols by leveraging link duration [17], adaptive Q-Learning [18] or independent of routing by leveraging ant colony optimization [19], meta learning [20]. However, none of these works have considered the dynamic traffic flows while designing their scheduling techniques. Besides, several throughput optimal scheduling techniques have been proposed under dynamic traffic scenarios by leveraging game theory [21] and reinforcement learning [22]. Moreover, in most of these works, Round-Robin Strategy (RRS) has been proposed for task scheduling, where a schedule at the beginning of communication is constructed by dividing the time into fixed-length rounds. By rotating the lead role responsibility among the UAVs, the RRS balances the energy consumption in the UAV networks. However, the cost of message exchanges during the lead role assignment creates a significant overhead. The policy closest to our work is the distributed task scheduling policy designed in [23] that addressed the lifespan enhancement issue for Wireless Sensor Networks (WSNs). But this work does not fulfill the essential performance of the UAV networks, e.g., real-time capturing of various KPIs like throughput, end-to-end delay. This is because irregular network connectivity due to high movement is more prevalent in UAV networks than WSNs.

A. Contributions

This paper aims to design a scalable, distributed and energy-efficient clustering algorithm based on the Dynamic Super Round Policy (DSRP). Our proposed algorithm achieves fine-grain task scheduling in the UAV networks. Particularly, in DSRP, the task scheduling is only accomplished prior to the initiation of each dynamic super round instead of in each round. We define a super round as a sufficiently long time span, within which there are several rounds. Our main objective of introducing the super round concept is to avoid frequent rotation of the lead role and conserve the scarcest resources as much as possible so that the lifespan of the UAV network is enhanced.

The major contributions of this paper are as follows:

- We first propose a Dynamic supeR rOund uaV clustEring (DROVE) algorithm, where the DSRP schedules data traffic according to dynamic traffic flows. This reduces the unnecessary role updates among the UAVs and hence computation and message overheads.
- We then devise a problem of distributed clustering task scheduling in the presence of dynamic traffic flows.
- We propose an optimal clustering task scheduling algorithm to serve dynamic traffic flows. We integrate dynamic traffic flows into the Lyapunov drift analysis framework, and determine the throughput optimality. Our proposed scheduling algorithm not only achieves maximum throughput, but also minimizes the average system traffic load under heavy traffic scenarios.
- We performed extensive simulations and show that DROVE achieves higher energy efficiency, network throughput, flow success rate than the three benchmarks. Moreover, DSRP-based task scheduling in the UAV networks offers greater efficiency in energy savings, and reduced end-to-end delay, packet drop rate compared to the state-of-the-art realizations.

B. Organizations

The rest of this paper is organized as follows. Section II discusses the related works. In Section III, we present the system model considered for the present work. In Section IV, we present DROVE and a clustering task scheduling algorithm. In Section V, we provide an analysis of the proposed clustering technique. Section VI presents the experimental results to confirm our analytical observations. Finally, we conclude our paper and discuss future work in Section VII.

II. Related Work

Nowadays, the UAV networks have attracted significant interest in improving network performance. We present here some works that are more relevant to our context.

A. Task Scheduling

To enhance network performance, task scheduling has been considered as a critical factor. For example, Hong *et al.* [17] first investigated the relationship between the UAV coordination control and the network topology for task scheduling. The authors then proposed a routing mechanism by leveraging the link duration. In [18], the authors proposed a routing strategy based on adaptive Q-learning. Adaptive Q-learning allows UAVs to take distributed, autonomous and adaptive routing decisions. In [19], the authors formulated an optimization problem for task scheduling by considering the energy budget and cache capacity of UAVs. They solved the optimization problem using ant colony optimization algorithm. Mou *et al.* [20] proposed a graph convolutional neural network for real-time task scheduling under limited communication connectivity in a UAV network. They further designed a meta learning scheme to reduce the time complexity of real-time executions. Recently, Tang *et al.* [24] designed a reinforcement learning based task scheduling approach under the node mobility, dynamic network traffic and link state settings. To decide the scheduling strategy, the authors used a double Q-learning algorithm with an improved delay-sensitive replay memory algorithm. To minimize task scheduling and computing delay, Zhou *et al.* [25] proposed a risk-sensitive reinforcement learning based Delay Orientated dynamic Task Scheduling (DOTS). They formulate the dynamic scheduling problem as energy-constrained Markov decision process.

B. Spectrum Sharing

Many spectrum sharing approaches have been proposed for UAV network. For example, Chen *et al.* [21] examined the interference-aware online spectrum access problem for multi-cluster flying ad-hoc network. They formulate the problem as data assisted multistage channel access game with the objective of mitigating the interference of all UAV clusters and reducing the channel switching cost during each slot. In [26], the authors proposed a Time Division Multiple Access (TDMA) based spectrum sharing model to improve the energy

efficiency. They jointly optimized several parameters, including the UAV hovering time and wireless powering duration, to minimize the energy consumption. Zhong *et al.* [22] presented a UAV aided self-organized device-to-device network model. They formulated the problem of joint optimization of relay deployment, channel allocation and relay assignment, with the objective to maximize the total capacity of the relay network. Kong *et al.* [16] formulated a spectrum scheduling problem to maximize the system throughput in the presence of dynamic traffic. To solve this problem, they proposed a model that considers a MaxWeight-type scheduling algorithm, where sharp flow-level dynamics are also taken into account that efficiently redirects the UAVs over a wide range of places. Duan *et al.* [27] addressed a scheduling problem to improve network capacity under dynamic traffic. They used fuzzy mapping among UAVs to reduce the complexity of scheduling. Hanyu *et al.* [28] designed an adaptive channel sharing technique to mitigate data loss caused by the buffer overflow. In the proposed technique, a suitable channel is selected based on the signal-to-noise ratio and data transfer rate. Although most of these proposed schemes achieve maximum throughput, they are unable to avoid unnecessary rescheduling problems. Due to unnecessary rescheduling, the UAVs are bound to consume significant energy, resulting in shortening of their lifespan.

C. Clustering Technique

Several clustering approaches have been proposed for UAV networks to improve network performance. Recently, Arafat and Moh [29] proposed a swarm-intelligence based clustering scheme for emergency communications in the UAV networks. Specifically, the authors used a particle swarm optimization fitness function to select a cluster head, where the inputs of the fitness function are inter-UAV distance and residual energy. In [30], the authors proposed a RRS-based clustering scheme, where the cluster head is elected based on the residual energy of the UAVs. Whereas, cluster formation is based on connectivity with the ground control station along with the luciferin value. Brust *et al.* [31] proposed a virtual force based clustering algorithm for 3D positioning of the UAVs. Particularly, the authors used a localized virtual force approach to communicate among the UAVs and to form the topology in UAV networks. All these works consider RRS to rotate the cluster head. Although RRS achieves energy efficiency, it overloads the system. Besides, RRS is only suitable for the continuous data delivery model, and is incompatible with other data delivery models, e.g., query- and event-driven. Hence, there is a need for more flexible task scheduling, which avoids unnecessary rescheduling and employs a dynamic scheduling.

D. Difference From Existing Works

Our work differs from the existing works in several ways. First, we design a UAV clustering algorithm DROVE, where the DSRP schedules data traffic according to dynamic traffic flows. Unlike existing works, DROVE features message and processing time complexity of $O(1)$ per UAV. Second, we design an optimal clustering task scheduling algorithm to cater dynamic traffic flow in the UAV network domain. More interestingly, by integrating dynamic traffic flows into the Lyapunov drift analysis framework, DROVE not only achieves maximum throughput, but also minimizes the average system traffic load under heavy traffic scenarios. Last but not the least, we are the first to consider the ray-tracing channel model while designing a dynamic traffic flow aware UAV clustering algorithm.

III. System Model and DSRP Background

In this section, we describe the models used in this work. Particularly, Section III-A presents the network model. We present the background information about the DSRP in Section III-B. Section III-C discusses the channel model. We then introduce the energy model in Section III-D. Section III-E presents the intruder mobility model. Finally, we introduce the use-case scenario to illustrate our concept and outline the problem in Section III-F.

A. Network Model

We assume collaborative autonomous systems, i.e., networks of connected UAVs or the multi-UAV network [19], [32], which interact with each other via wireless communication to execute one or more tasks. Generally, no central entity is required to coordinate the actions of the individual UAVs in this type of network. Nevertheless, a central entity, e.g., GS might be used for data collection, processing and maintenance. For exposition purpose, we consider M homogeneous fixed-wing UAVs flying horizontally at a constant altitude H [24], [25] to monitor and collect data across N ($M > N$) different regions/clusters in the Area of Interest (AoI), and GS is located at the position $\mathrm{x_G} = (0, 0, 0)$ of a 3D Cartesian coordinate system. The UAVs in a particular cluster are denoted as $K = \{K_1, \ldots, K_n\}$, where K_n is the set of UAVs located in the n-th cluster, $n \in \mathbb{N} = \{1, \ldots, N\}$. Therefore, we can say: $K_n \cap K_{n'} = \phi$, where $n \neq n'$ and $n \in \mathbb{N}$, while $K_n = |K_n|$ represents the number of UAVs in the n-th cluster. Additionally, we consider the position of a particular UAV as $\mathrm{x_D} = (\bar{\mathrm{x}}_\mathrm{D}, H)$, where $\bar{\mathrm{x}}_\mathrm{D} \in \mathbb{R}^2$ and H are larger than the maximum height of the obstacle. However, during experimental evaluation, we relax the constant H by implementing dynamic H.

We assume that the UAVs follow a mobility model [24], in which the UAVs fly with a constant speed and randomly redirect themselves within a certain angle. Since the continuous flying and frequent redirection behavior are significantly energy consuming, we also assume that the UAV redirection decisions are taken collaboratively and regularly at predefined fixed time intervals, called rounds. Furthermore, we assume that each round begins with a setup phase, which assigns the roles, e.g., Cluster Head (CH) and Cluster Member (CM), followed by a steady phase, which captures the video/imagery data and transmits them to GS. We divided each steady phase into frames and further each frame is divided into time slots of duration T, as shown in Figure 1. We consider that the energy of the UAVs is limited and supplied by off-the-shelf batteries. Table I provides a list of major notations used in this paper.

TABLE I
SUMMARY OF IMPORTANT NOTATIONS

Notation	Description	Notation	Description
M	Number of UAVs	H	Altitude of UAVs
n	Number of clusters, $n = 1, \ldots, N$	μ_n	Mean of $F_n[k;t]$, $\mu_n > 0$
t	Time slot index within a frame, $t = 1, \ldots, T$	$\mathcal{F}_n^{max}$	Maximum value of $F_n[k;t]$
$\mathbf{q}(t)$	UAV trajectory prediction on the horizontal plane	$\mathbb{C}[k]$	Feasible schedule in frame k
$d(t)$	Distance between the detecting UAV and the intruder at time t	$\mathcal{C}$	Set of all feasible schedules
$\mathcal{F}_n[k;t]$	Set of traffic flows arrived at n-th cluster in t-th time of frame k	$F_n[k;t]$	Cardinality of $\mathcal{F}_n[k;t]$
$C_n[k]$	Service decision for n-th cluster at frame k, $C_n[k] \in \{0,1\}$	s_n^{max}	Maximum value of $\mathcal{S}_{n,f}[k;t]$
$\mathcal{S}_{n,f}[k;t]$	Channel rate of flow f at n-th cluster in t-th slot of frame k	p_n^{min}	Probability that $\mathcal{S}_{n,f}[k;t] = 0$
p_n^{max}	Probability that $\mathcal{S}_{n,f}[k;t]$ achieves the maximum rate	$\mathbb{L}[k]$	Traffic load vector
$L_n[k;t]$	Total traffic load at n-th cluster in t-th slot of frame k	γ	Channel gain
$L_n[k]$	Total traffic load at n-th cluster at the beginning of frame k	l	Length of a message
$\mathcal{L}_n[k;t]$	Newly generated traffic load arrived at n-th cluster in t-th slot of frame k	E_{tx}	Transmitting energy for a message
$D_{n,f}[k;t]$	Residual packets of flow f at n-th cluster in t-th slot of frame k	E_{rx}	Receiving energy for a message
$\mathcal{L}_n[k]$	Total Newly generated traffic load arrived at n-th cluster in frame k	$\mathbb{d}$	Distance between sender and receiver
ρ_n	Average traffic load of all newly arriving flow in n-th cluster at each slot	E_{CR}	Energy consumption rate of a UAV
$\mathcal{N}_n[k;t]$	Set of traffic flow in n-th cluster at t-th slot of frame k	G_{SR}	Number of rounds in a SR
$\lambda_n[k;t]$	Amount of traffic load decrease in n-th cluster at t-th slot of frame k	R_f	Reclustering factor
$\mathcal{U}$	Number of UAVs participating in the clustering procedure	ϵ	Heavy traffic load parameter

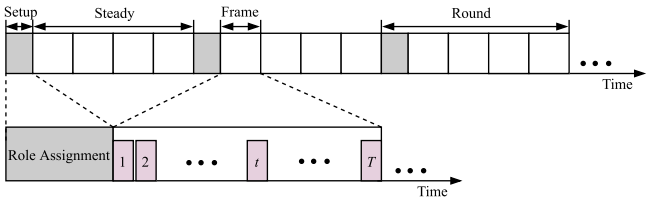


Fig. 1. Time line of a round.

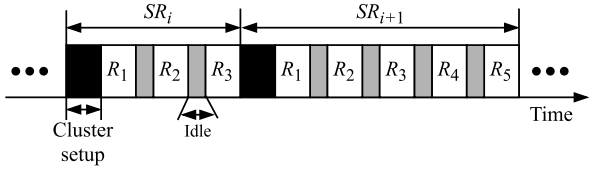


Fig. 2. Illustration of super round.

B. DSRP Background

In this paper, we define the Super Round (SR) as a collection of rounds in which reclustering occurs only at the beginning of the first round in the SR. The i-th SR, i.e., SR_i, consists of three rounds, R_1, R_2 and R_3, and re/clustering is performed before R_1, as shown in Figure 2. We consider that the length of an SR is determined dynamically to reduce the unnecessary reclustering, and hence the computation and message overheads. To realize our SR concept, we must determine the length of an SR, consensus among the UAVs on reclustering time, scheduling intra-cluster data traffic through TDMA frames, and synchronizing the UAVs during reclustering. In Section IV-A, we provide details about each aforesaid step for the realization of the SR. Although the length of a round is fixed, however, consecutive time slots allotted to a UAV and the SR length are variable, as in DROVE reclustering is decided on-demand basis. Particularly, the number of consecutive time slots allotted to a UAV is decided online considering the data traffic load at that time. Whereas, the length of an SR is decided online after satisfying certain conditions, e.g., threshold residual energy, and distance between the intruder and the UAV. In DROVE, as the time slots allotted to a UAV and the length of an SR are determined online, the clustering overhead remains controlled during the entire network lifespan. For example, if all UAVs possess significant remaining energy, the length of an SR is chosen as long as possible since the CHs are capable of managing their clusters over a prolonged duration. Nevertheless, once the remaining energy of a CH reaches a threshold, the SR length reduces to relinquish the energy constraint CH in shorter time intervals.

C. Channel Model

Typically, the channel gain, γ (in dB), for a transmitter-receiver separation distance $\mathbb{d}$ can be expressed as:

$$\gamma = \beta - 10\alpha \log_{10} \mathbb{d} + \xi,$$

where α is the path loss exponent, β is the channel gain at the reference distance $\mathbb{d}_0 = 1$ m, ξ is a Gaussian random variable and usually modeled as $\mathcal{X}(0, \sigma_{SF}^2)$ to capture the shadowing effect. Additionally, α, β and σ_{SF}^2 rely on the propagation scenario, e.g., Line-of-Sight (LoS) or Non Line-of-Sight (NLoS) propagation. This work exploits the spatial structure of ξ, where it might be spatially correlated due to the presence of obstacles and reflectors in the AoI. As a result, ξ might exhibit diverse distributions in distinct areas [33]. Motivated by this fact, we consider an N-segment (or, cluster) ray-tracing channel model [33], which predicts the channel with high precision. For simplicity and tractability, we consider that the Doppler effect due to the UAV's mobility is perfectly compensated using the technique proposed in [34]. Moreover, in [35], the researchers have shown through empirical studies that for a moving UAV with a velocity 10 m/s, the impact of the Doppler effect is negligible. Accordingly, the channel gain (in dB), for a transmitter-receiver can be further expressed as:

$$\gamma(\mathrm{x_R}) = \sum_{n=1}^{N} (\beta_n - 10\alpha_n \log_{10} \mathbb{d}(\mathrm{x}) + \xi_n)\, \mathbb{I}\{(\mathrm{x_R}, \mathrm{x_T}) \in K_n\}, \quad (1)$$

where $\mathrm{x_R}$ is the position of a receiver, $\mathrm{x_T}$ is the position of a transmitter, β_n is the average channel gain at $\mathrm{d}_0 = 1$ m, α_n is the average path loss exponent, ξ_n is modeled as $\mathcal{X}(0, \sigma_n^2)$ to capture the shadowing effect, $\mathrm{d(x)} = \|\mathrm{x_R} - \mathrm{x_T}\|$ is the distance between the transmitter and the receiver, and $\mathbb{I}\{A\}$ is an indicator function, whose value 1 if condition A satisfied, and 0 otherwise.

Let us assume that the transmission power of the UAV, P_0, is constant. Accordingly, using eq. (1), the instantaneous channel capacity can be expressed as:

$$S(t) = B\log_2\left(1 + \frac{P_0\gamma(\mathrm{x_R})}{\sigma^2}\right) = B\log_2\left(1 + \frac{\mathcal{X}_0}{H^2 + ||\mathbf{q}(t)||^2}\right), \quad 0 < t < T, \tag{2}$$

where B is the channel bandwidth, σ^2 is the white Gaussian noise power at the receiver end, $\mathcal{X}_0 = G_0P_0/\sigma^2$ is the reference received signal-to-noise ratio at the reference distance $\mathrm{d}_0 = 1$m, and $\mathbf{q}(t) = [x(t), y(t)]^T \in \mathbb{R}^{2\times 1}$ is the UAV trajectory prediction on the horizontal plane. Therefore, using eq. (2), the total information bits $\mathcal{S}$ received at the receiver end during the time instance T with a function of the UAV trajectory, $\mathbf{q}(t)$ is given as:

$$\mathcal{S}(\mathbf{q}(t)) = \int_0^T B\log_2\left(1 + \frac{\mathcal{X}_0}{H^2 + ||\mathbf{q}(t)||^2}\right)dt. \tag{3}$$

D. Energy Model

We assume that there are two major components in a UAV for energy consumption [15]. The first component is for communication, specifically, signal processing and radiation. Whereas, the second component is the propulsion energy, which is essential for lifting the UAV and supporting mobility, if required. In practice, the communication energy is significantly less than the propulsion energy, e.g., a few watts versus hundreds of watts [15]. Motivated by [25] and [36], we consider here only the energy consumption related to the communication process, as this work deals with the issue related to task scheduling. Particularly, the energy consumption in the communication process is composed of two parts: the transmission energy E_{tx} and the receiving energy E_{rx}. The transmission energy for sending a message size l at a distance d is $E_{tx}(l, \mathrm{d}) = l\times(E_{elec}+E_{com})$, where E_{elec} (J/bit) is the energy consumption per bit by the electronic circuitry and E_{com} (J/bit) is the communication energy consumption per bit [36]. Let d_{th} be the threshold distance between the sender and receiver UAVs. Therefore, E_{com} is calculated as: $\epsilon_{fs} \times \mathrm{d}^{\eta}$, where ϵ_{fs} (J/bit/m$^{\eta}$) is a constant and $\eta \geq 2$ is the path loss exponent [36]. In contrast, the receiving energy E_{rx} for a message size l is $E_{rx}(l) = l \times E_{elec}$.

In the UAV networks, video/image compression is also assumed as an integral part, thus the energy consumption formula presented above needs necessary modifications. Considering the experiments on energy consumption in video/image compression algorithm [37], a video/image compression enabled UAV can compress the captured videos/images and reduce the size of the packet workload forwarded to the GS. To execute the video/image compression task, a UAV must spend additional processing energy. If k_t is the video/image compression ratio, the energy model considering the video/image compression algorithm in this work can be expressed as:

$$E_{tx}(l, \mathrm{d}) = k_t \times l \times (E_{elec} + E_{com}), \tag{4}$$

$$E_{rx}(l) = k_t \times l \times E_{elec}. \tag{5}$$

Let R_t and R_c be the data transmission and reception rates, respectively. Thus, from eqs. (4) and (5), we can determine the energy consumption rate of a UAV [36] as:

$$E_{CR} = R_tE_{tx}(l, \mathrm{d}) + R_cE_{rx}(l). \tag{6}$$

E. Intruder Mobility Model

In this work, we model the intruder mobility within the AoI by a two-dimensional random walk process [24]. Specifically, we assume that intruders appear uniformly and independently on the boundary of the AoI and tend to move in roughly a straight line before changing direction. Let t_0 be the starting time of the intruder at position $Z_0 = Z(t_0)$. We assume that the intruder makes a step of constant length ζ in each unit of time. However, the direction of the step is uniformly chosen in $\theta = [-\pi/2, \pi/2]$. For each further step, an intruder moves length ζ and the direction angle is chosen uniformly from $[\theta - \varphi, \theta + \varphi]$, where $\varphi \in [0, \pi]$ is the free parameter of the model. Note that if $\varphi = \pi$ the mobility is a random walk without any direction, while $\varphi = 0$ is a straight line. Let Z_s be the intruder position at the s-th step. We estimate the trajectory of the intruder as follows:

$$Z_{s+1} = Z_s + \zeta e_{\theta_{s+1}},$$

where e_θ represents the unit vector of $(\cos\theta, \sin\theta)$. Therefore, the coordinates of the intruder position at Z_{s+1} are given as:

$$x_{Z_{s+1}} = x_{Z_s} + \zeta\cos\theta_{s+1},$$
$$y_{Z_{s+1}} = y_{Z_s} + \zeta\sin\theta_{s+1}.$$

F. Use-Case Scenario and Problem Description

We first introduce an example of a use-case scenario of the UAV networks to demonstrate our proposed concept, which is simplified for the sake of clear illustration. Without loss of generality, the proposed concept also remains valid for large and more complex systems.

Let us assume a set of autonomously flying UAVs that collaborate to perform distributed ground surveillance, e.g., as part of a search and rescue mission in a (possibly dense) urban area. Every UAV has a set of cameras, which allows it to monitor a particular area at a time. In fact, to cover a wide ground area in a short time frame, several UAVs fly in formation. When the UAVs operate, while monitoring the ground area, in close proximity, they must coordinate with each other to track the moving intruder. Figure 3 shows a scenario, where the UAVs must monitor and track a moving intruder in a densely populated area. In Figure 3, the four UAVs, D_1, D_2, D_3, and D_4 (i.e., $|K_n| = 4$) are flying to monitor an area, denoted as n-th cluster. Every UAV is an autonomous unit

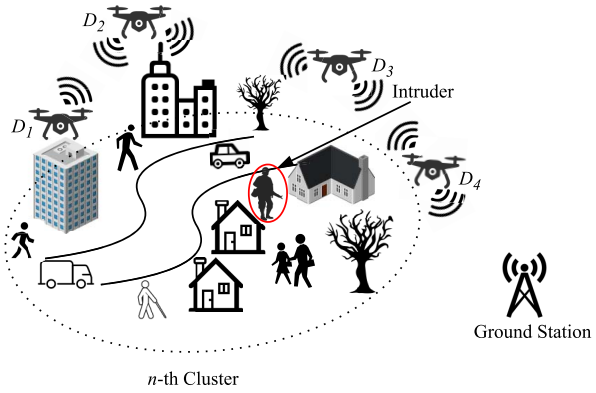


Fig. 3. Example of collaborative monitoring using UAVs.

that can operate on its own. Specifically, every UAV consists of sensors, which enable it to position itself accurately (e.g., GPS), measure its maneuver (e.g., gyroscope, accelerometer), and sense its environment via various cameras, including infrared, ultrasonic, Lidar. Nowadays, all these sensors are common in commercial off-the-shelf systems, and in fact, the customer grade UAVs are equipped with such cameras [2].

During the UAVs' mission, the D_3 detects an intruder using a front facing camera. The D_3 has two options for reacting to this situation: (1) it continues to track the intruder, or (2) it informs the nearby UAVs to track the intruder. The D_3 by itself does not have enough information based on which it can decide in which direction to move for tracking, as the intruder could move in any direction. However, by exchanging information with the nearby UAVs, the D_3 can learn that the intruder is moving towards the D_4. As a consequence, the D_3 decides to transfer the tracking task to the D_4. In this way, the D_3 not only prevents collision with the D_4, but also ensures more precise tracking. We assume here that the D_3 is aware of the D_4's position through sharing of location information among the UAVs. Specifically, UAVs follow a light-weight message passing using the method in [38] to share location information.

Interestingly, with respect to the above use-case scenario, RRS is more severe as D_4 has no provision for sending repeated data traffic, although the moving intruder is being precisely tracked by the D_4 due to close proximity. For sending repeated data traffic, the D_4 must wait for several rounds of data transmission cycles, and by that time, the intruder may detour its route. Therefore, it is crucial to find the length of a round and the number of (successive) time slots assigned to a UAV as the shorter round and time slot increases intrusion tracking performance and energy efficiency at the cost of higher overhead.

In this work, we aim to determine the number of time slots assigned to a UAV for maximizing the tracking performance (in terms of throughput) and energy efficiency. We denote $\mathcal{F}_n[k;t]$ and $\mathbb{C}[k]$ as the set of traffic flow arriving from the n-th cluster at the t-th time slot of the frame k, and the feasible schedule in frame k, respectively. If there are a set of $\mathcal{C}$ feasible schedules, finding the number of time slots τ for a CM to deliver the total traffic load to the CH in cluster n is solving of the following equation:

$$\arg\max_{\mathbb{C}\in\mathcal{C}} \langle \mathbb{L}[k], \mathbb{C}[k] \rangle \tag{7}$$

subject to

$$\mathcal{F}_n[k;t] \leq \mathcal{F}_n^{max}, \forall n, t, k \tag{8}$$

$$L_n[k] > 0, \mathcal{F}_n^{max} > 0, \tag{9}$$

$$C_n^*[k] = 1, \tag{10}$$

where $\mathbb{L}[k]$ and $\mathbb{C}[k]$ are the traffic load vector and feasible schedule in frame k, respectively. The constraint in eq. (8) guarantees that the set of traffic flow arriving from the n-th cluster at the t-th time slot is non-negative. The constraint in eq. (9) specifies that the traffic load in cluster n at the beginning of frame k is supposed to be greater than zero. The constraint in eq. (10) specifies that the flow contribution of the UAVs hovering over a cluster n is 1.

IV. The Proposed Clustering Technique: DROVE

This section discusses the main design of DROVE. In DROVE, a cluster is formed using a distributed algorithm at the beginning of a SR with the following objectives: (i) distribution of all UAVs into various clusters in a distributed way to satisfy our use-case scenario (see Section III-F), e.g., scalability, (ii) minimization of inter- and intra-cluster communication costs to conserve energy, (iii) light-weight CH selection strategy, and (iv) limiting the time complexity to $O(1)$. We divided the proposed clustering procedure into three phases, namely, CH selection, cluster formation and data collection. The following sections describe these three phases in detail.

A. CH Selection

In DROVE, CH selection procedure is triggered once an intruder is detected in the AoI. For brevity, we consider that detecting UAV only tracks the intruder with the closest distance if a UAV detects multiple intruders. Since it is unlikely that more than a pair of UAV and intruder can exist at the same location, there will be no unattended intruder in the AoI. In DROVE, detecting UAVs only participate in the CH selection and cluster building phases. The rest of the UAVs remain in the surveillance mode for detecting future intruders. After detecting an intruder, the detecting UAV poses itself as the temporary CH and broadcast a $Hello(ID, E_{re}, d(t))$ message, where ID is the UAV identification number, E_{re} is the remaining energy level of the detecting UAV, and $d(t)$ is the distance from the detecting UAV to the intruder at time t. In DROVE, we calculate E_{re} based on the initial energy budget and energy consumption rate, i.e., eq. (6) of a UAV. Whereas, similar to [39], we calculated $d(t)$ as follows:

$$d(t) = \sqrt{H^2 + [x_u(t) - x_{in}(t)]^2 + [y_u(t) - y_{in}(t)]^2},$$

where (x_u, y_u) and (x_{in}, y_{in}) are the coordinates of a UAV and the intruder, respectively. After receiving the $Hello$ message, the neighboring UAVs who have detected the same

Algorithm 1 CH Selection Phase

while detecting an intruder by the j-th UAV **do**
  Listen to the channel
  if not received $Hello$ message **then**
    Calculate E_{re}^{j}
    Calculate $d_j(t)$
    Send $Hello(ID, E_{re}^{j}, d_j(t))$ to all neighbors
    Receive $Hello\_Ack(ID, E_{re}, d(t))$ from all neighbors
    Calculate AE_{re}
    Calculate $d_{av}(t)$
    if $AE_{re} < \ldots < E_{re}^{max}$ & $d_{av}(t) > \ldots > d(t)^{max}$ **then**
      Advertise CH $Head\_Msg$
      Receive CH_Join from all neighbors
      Send CH_Acpt message to CMs
    else
      Receive $Head\_Msg$ message from CH
      Send CH_Join message to CH
      Receive CH_Acpt message from CH
    end if
  end if
end while

intruder send $Hello\_Ack(ID, E_{re}, d(t))$ message to the temporary CH. Based on the received $Hello\_Ack$ message, the temporary CH determines the average residual energy, AE_{re}, and the average distance, d_{av}, between a detecting UAV and the intruder. The temporary CH nominates the i-th UAV from the neighbors as the new CH whose E_{re} is maximum, i.e., $AE_{re} < \ldots < E_{re}^{max}$, and $d(t)$ is minimum, i.e., $d_{av}(t) > \ldots > d(t)^{min}$. We calculate AE_{re} and $d_{av}(t)$ as follows: $AE_{re} = \sum_{j=1}^{\mathcal{U}} E_{re}^{j}/\mathcal{U}$ and $d_{av}(t) = \sum_{j=1}^{\mathcal{U}} d_j(t)/\mathcal{U}$, where $\mathcal{U}$ is the number of UAVs participating in the clustering procedure. In case, there are more UAVs with the same $(E_{re}^{max}, d(t)^{min})$, one of them is chosen randomly. Algorithm 1 illustrates the action performed by the UAV during CH selection.

Although there are several mechanisms to trigger the reclustering procedure, however, we follow a simple procedure to limit unnecessary reclustering in DROVE, thereby, reducing the computation and implementation complexities. In particular, during the idle time, i.e., the time duration between two successive rounds, all CMs in a cluster estimate the present residual energy of its CH, e_{re}. Whenever a CM finds that e_{re} falls below E_{th} (special condition), signifying possible exhaustion of the CH's energy, the CM broadcasts a recluster scheduling message. After receiving the recluster scheduling message, the CH broadcast a message to inform its CMs for invoking the cluster setup procedure at the end of the upcoming round. In this way, the current SR ends and a new SR starts executing with the cluster setup procedure. In contrast, if the CMs find that $e_{re} > E_{th}$, the same cluster operates in successive rounds until $e_{re} \leq E_{th}$. In DROVE, a CM estimates e_{re} as follows:

$$e_{re} = E_{init} - G_{SR} \times E_{CR},$$

where E_{init} is the initial energy of a UAV, G_{SR} is the number of rounds in a SR and E_{CR} is calculated using eq. (6). Whereas, similar to [23], a CM calculates E_{th} as follows:

$$E_{th} = R_f \times E_{re}^{CM}, \tag{11}$$

where R_f ($0 \leq R_f \leq 1$) is the reclustering factor and E_{re}^{CM} is the CM's residual energy. In DROVE, for a particular network, R_f is constant and decided based on the application's requirement.

In DROVE, as mentioned earlier, reclustering is triggered under a special condition instead of in periodic time intervals. This property of DROVE not only reduces the significant communication overhead, but also reduces the considerable computation overhead through a decrease in the number of reclustering rounds. Note that the worst case scenario occurs in DROVE when a SR consists of a single round. After the CH selection, determining dynamically the length of an SR and number of (successive) time slots assigned to a CM within a round are crucial, these two significantly affect the intrusion tracking efficacy (see Section III-F). To determine these two parameters, we first investigate the variation in SR length in Section IV-A.1. In Section IV-A.2, we then present the analysis of consecutive time slot allotment.

1) SR Length Analysis: In this section, we investigate the length of an SR, G_{SR}. Let $E_{re,i}^{CH}$ and $E_{re,i}^{CM}$ be the residual energy of CH and CM at the start of the steady state phase in SR_i, respectively. Alternatively, $E_{re,i}^{CH}$ and $E_{re,i}^{CM}$ are the residual energy of a CH and CM after the setup phase, respectively. Also, let e^{CH} and e^{CM} be the energy consumption of a CH and CM at the steady state phase, respectively. It is worth mentioning that the values of e^{CH} and e^{CM} depend on the size of the cluster. Let us assume that r and $(r+1)$ are two consecutive rounds of the SR_i, and a UAV has residual energy $E_{re,i}(r)$ and $E_{re,i}(r+1)$ during the steady state phase of rounds r and $(r+1)$, respectively. Therefore, we can express $E_{re,i}(r+1)$ as:

$$E_{re,i}(r+1) = \begin{cases} E_{re,i}(r) - e^{CH}, & \text{if UAV acted as CH in } (r+1) \\ E_{re,i}(r) - e^{CM}, & \text{if UAV acted as CM in } (r+1) \end{cases}.$$

In DROVE, a UAV is chosen as a CH, based on the maximum residual energy and minimum distance from the intruder. Hence, if a UAV is chosen as a CH at the start of r-th round, then:

$$E_{re,i}^{CH}(r) = E_{re,i}(r) + e^{CH}.$$

As mentioned earlier, the reclustering procedure is triggered once the residual energy of a CH crosses a threshold. If G_{SR} is the number of rounds in a SR without any cluster setup phase, we can express E_{th} as:

$$E_{th} \geq E_{re,i}^{CH}(r_f), \tag{12}$$

where $E_{re,i}^{CH}(r_f)$ is the residual energy of a CH at the final round, r_f, of the SR_i. Considering eq. (11), we can write eq. (12) as:

$$R_f \times E_{re,i}^{CM} \geq E_{re,i}^{CH}(r_f). \tag{13}$$

Since SR_i consists of G_{SR} rounds, we can express $E^{CH}_{re,i}(r_f)$ as:

$$E^{CH}_{re,i}(r_f) = E^{CH}_{re,i} - G_{SR} \times e^{CH}. \tag{14}$$

Using eq. (14), we can express eq. (13) as:

$$R_f \times E^{CM}_{re,i} \geq E^{CH}_{re,i} - G_{SR} \times e^{CH}.$$

After basic transformation, we can write the above equation as:

$$G_{SR} \geq \frac{E^{CH}_{re,i} - R_f \times E^{CM}_{re,i}}{e^{CH}}.$$

Hence, the length of an SR can be determined dynamically by finding the lowest integer of the above equation, i.e.,

$$G_{SR} = \left\lceil \frac{E^{CH}_{re,i} - R_f \times E^{CM}_{re,i}}{e^{CH}} \right\rceil. \tag{15}$$

From eq. (15), it is clear that G_{SR} is decided by $E^{CH}_{re,i}$, $E^{CM}_{re,i}$, e^{CH} and R_f. Among these parameters, e^{CH} and R_f are fixed. Thus, $E^{CH}_{re,i}$ and $E^{CM}_{re,i}$ are the parameters that directly influence the value of G_{SR}. It is worth noting that as the surveillance operation progresses, UAV's energy level will decrease gradually, resulting in the shortening of the SR length, i.e., G_{SR}. Eventually, the shortening of SR length will increase the number of reclustering procedures. However, such an increase in reclustering procedure is essential to distribute the work load evenly among the UAVs, leading to subsequent enhancement in network lifespan.

2) Time Slot Allotment Analysis: This section provides a theoretical analysis to determine the number of consecutive time slots to be assigned to a CM in a cluster. Let $\mathcal{F}_n[k;t]$ be the set of traffic flows arriving from the n-th cluster at the t-th time slot of the frame k. The cardinality of the set $\mathcal{F}_n[k;t]$ is represented by $F_n[k;t]$, i.e., $F_n[k;t] = |\mathcal{F}_n[k;t]|$. We consider that $\{\mathcal{F}_n[k;t], t = 1, \ldots, T, k \geq 0\}$ are independently and identically distributed (i.i.d.) over time with mean $\mu_n > 0$, and $\mathcal{F}_n[k;t] \leq \mathcal{F}^{max}_n, \forall n, t, k$ for some $\mathcal{F}^{max}_n > 0$.

Without loss of generality, we consider that each cluster is served by at most one data flow in each time slot. Further, we consider that at least one UAV is hovering over a cluster in each frame. Furthermore, we consider that all M UAVs hovering over various clusters do not interfere with each other and hence, data traffic can flow simultaneously. To avoid interference, UAVs employ an inter-UAV interference coordination technique, e.g., orthogonal resource block allocation [28]. Finally, we assume that a set of clusters monitored by the UAVs simultaneously in frame k is a feasible schedule and designated by: $\mathbb{C}[k] \triangleq (C_n[k])^N_{n=1}$, where exact flow contribution of the UAVs hovering over a cluster n is 1. Let $\mathcal{C}$ be the set of all feasible schedules. We assume that all M UAVs are directly connected with the GS via point-to-multipoint micro/millimeter wave backhauling with adequate bandwidth [16].

Since we use a limited number of coding and modulation techniques while sending data traffic, each data traffic flow has a finite transmission rate. Let $\mathcal{S}_{n,f}[k;t]$ be the channel rate (i.e., packets per time slot) of the traffic flow f in cluster n in the t-th time slot of frame k, which is i.i.d. with the minimum and maximum channel rates as 0 and s^{max}_n, respectively. It is worth reminding that we can derive $\mathcal{S}_{n,f}[k;t]$ using eqs. (2) and (3). We further consider that both probabilities of each traffic flow with the minimum and maximum channel rates are positive, particularly, $p^{min}_n \triangleq \Pr\{\mathcal{S}_{n,f}[k;t] = 0\} > 0$ and $p^{max}_n \triangleq \Pr\{\mathcal{S}_{n,f}[k;t] = s^{max}_n\} > 0$. Without loss of generality, we assume that the data traffic characteristic is cluster dependent.

To characterize the traffic flow under a dynamic application scenario, we use *traffic load* to determine the minimum number of time slots (see Figure 1) required to deliver the newly generated (or, existing) traffic flows. Precisely, we designate: $L_n[k;t] \triangleq \sum_{f \in \mathcal{N}_n[k;t]} \lceil D_{n,f}[k;t]/s^{max}_n \rceil$ and $\mathcal{L}[k;t] \triangleq \sum_{f \in \mathcal{F}_n[k;t]} \lceil F_{n,f}[k;t]/s^{max}_n \rceil$ as the total traffic load and the newly generating traffic load in cluster n in the t-th time slot of frame k, respectively, where $D_{n,f}[k;t]$ is the number of residual data packets of traffic flow f and $\mathcal{N}_n[k;t]$ is the set of traffic flows in the cluster n in the t-th time slot of the frame k. It is worth mentioning that upon the transmission of data traffic by the CH from any cluster, the amount of traffic load will decrease gradually. Let $\lambda_n[k;t]$ be the amount of traffic load decrease in the cluster n in the t-th time slot of the time frame k. Therefore, $\lambda_n[k;t]$ depends on the number of UAVs hovering over the cluster n to deliver the traffic load, the amount of remaining traffic load and the associated channel rate. We assume that there is at least one UAV hovering over a cluster and collectively serve one flow, i.e., $0 \leq \lambda_n[k;t] \leq 1$. Accordingly, during the monitoring in cluster n in the t-th time slot of time frame k, we can write the total traffic load as:

$$L_n[k] \triangleq L_n[k;1].$$

Likewise, we can express the total newly generated traffic loads as:

$$\mathcal{L}_n[k] \triangleq \sum\nolimits_{t=1}^{T} \mathcal{L}_n[k;t],$$

and the amount of remaining traffic load in the time frame k as:

$$\lambda_n[k] \triangleq \sum\nolimits_{t=1}^{T} \lambda_n[k;t].$$

In summary, we can describe the progress of the traffic load as:

$$L_n[k+1] = L_n[k] + \mathcal{L}_n[k] - C_n[k]\lambda_n[k],$$

for $n = \{1, \ldots, N\}$.

In this work, our target is to determine the number of successive time slots required by a CM in each cluster to serve its data traffic with the following aims: (1) maximizing the throughput, and (2) minimizing the end-to-end latency. We assume that, in each time slot, the traffic load can be reduced by at most 1 in each cluster and hence the data traffic of the newly arriving flows in each cluster can be at most 1. Considering our target and earlier analysis, we develop an optimal time slot assignment algorithm for CM, as given in Algorithm 2. In our optimal time slots allotment algorithm,

Algorithm 2 Optimal Time Slots Allotment

For each time frame k, under the present traffic load vector $\mathbb{L}[k] \triangleq (L_n[k])_{n=1}^{N}$,

(1) Determine and assign τ consecutive time slots to a CM for delivering the total traffic load to the CH in cluster n, i.e., select $\mathbb{C}^*[k;\tau] \triangleq (C_n^*[k])_{n=1}^{N}$ such that

$$\mathbb{C}^*[k;\tau] \in \arg\max_{\mathbb{C}\in\mathcal{C}} \langle \mathbb{L}[k], \mathbb{C}\rangle.$$

(2) In a time frame k, a CM reserves τ consecutive time slots for delivering the traffic load, i.e., $C_n^*[k] = 1$, to the CH with the maximum channel rate.

the CH of a cluster initially collects the traffic load information from all CMs in every time frame and accordingly assigns time slots to the CMs for delivering the traffic load. Here, it is worth mentioning that our proposed Algorithm 2 is significantly different from the traditional queue length based MaxWeight algorithm. Particularly, in the MaxWeight algorithm, the system uses a first-come-first-server queuing mechanism. In contrast, DROVE uses a flow-level dynamic mechanism, while delivering the traffic load.

B. Cluster Formation

In DROVE, the cluster building procedure initiates after the CH selection. During the cluster building procedure, the newly elected CH broadcasts a $Head\_Msg$ message within the communication range. The $Head\_Msg$ message consists of the UAV ID and the remaining energy level information. Upon receiving the $Head\_Msg$ message, a UAV replies to the newly elected CH by sending the CH_Join message. In case a UAV receives several $Head\_Msg$ messages from its neighboring CHs, it selects the closest CH by sending the CH_Join message as the smallest power level is required to reach [38]. In this work, similar to [40], we consider that a UAV determines the inter-UAV distance using on-board ultra-wideband ranging measurement. After receiving the CH_Join messages, the CH acknowledges the joined UAVs by sending the CH_Acpt messages and designates them as the CMs.

C. Data Collection

In DROVE, after the cluster setup phase using Algorithm 1, each CH generates a TDMA schedule for its CMs. Due to the application's requirement, the CH dynamically schedules the time slot(s) among the CMs using Algorithm 2. We vividly discussed the dynamic time slots assignment problem in Section IV-A.2. Upon receiving the schedule, each CM transmits its captured data to its CH. As the data packets collected at the CH's end are vastly correlated, therefore, each CH aggregates the received data packets into a single data packet. Finally, the CH sends the aggregated data packets to the GS at the end of the frame.

V. Theoretical Analysis of DROVE

In this section, we analyze the message and time complexities of DROVE in Section V-A. We then show that our time slot allotment algorithm, i.e., Algorithm 2 is optimal in Section V-B.

A. Message and Time Complexities

In this section, we first analyze the worst-case message complexity of our proposed algorithm in Lemma 1. We then analyze the worst-case time complexity of our proposed algorithm in Lemma 2.

Lemma 1: In DROVE, the cluster setup phase has a worst-case message complexity of $O(M)$ in the network, i.e., $O(1)$ per UAV.

Proof: In our work, we assume that M UAVs are flying in the AoI and N clusters are formed. It is trivial to determine the number of messages exchanged during the cluster set in Section IV-A. Particularly, the maximum number of messages exchanged in the network is given by:

$$3N + 2(M - N) = 2M + N.$$

Therefore, the worst-case message complexity in the network is $O(M)$. Alternatively, in DROVE, the order of message exchange complexity is $O(1)$ per UAV, and $O(M)$ in the network. Note that the complexity of the message passing method used in DROVE for sharing location information is $O(1)$ per UAV [38]. □

Lemma 2: The proposed clustering algorithm has a worst-case time complexity of $O(M)$ in the network, i.e., $O(1)$ per UAV.

Proof: We present the proposed clustering algorithm in Algorithm 1. It is trivial to calculate the time complexity of various parameters in Algorithm 1. Specifically, as M UAVs are participating in the CH selection phase and all operations within the algorithm take $O(1)$ time, the total time complexity is obtained as $O(M)$ or $O(1)$ per UAV. □

B. Optimality

In this section, we first show that the CH selection policy in DROVE is optimal. Second, we show that DROVE can achieve optimal throughput. Our optimal throughput analysis is based on the Lyapunov drift function [16], [25], as it is one of the most promising approaches for delay-aware resource control problem in wireless systems.

Theorem 1: In DROVE, the CH selection policy P is the optimal selection policy to maximize the network lifespan.

Proof: Since the proof is similar to that of [Ghosal *et al.* [41], Theorem 3], we do not repeat it here. □

Theorem 2: The time slot allotment algorithm is throughput optimal, i.e., it stabilizes the system for any arrival traffic load strictly within the AoI.

Proof: To prove the stability of the system, we modeled the time slot allotment as the Lyapunov function $V(k) \triangleq \|\mathbb{L}[k]\|$ and its conditional expected Lyapunov drift as:

$$\begin{aligned}
&\mathbb{E}[\Delta V(\mathbb{L})|\mathbb{L}[k] = \mathbb{L}] \\
&\quad \triangleq \mathbb{E}[\|\mathbb{L}[k+1]\| - \|\mathbb{L}[k]\||\mathbb{L}[k] = \mathbb{L}[k]] \\
&\quad = \mathbb{E}[\sqrt{\|\mathbb{L}[k+1]\|^2} - \sqrt{\|\mathbb{L}[k]\|^2}|\mathbb{L}[k] = \mathbb{L}[k]] \\
&\quad \leq \frac{1}{2\|\mathbb{L}[k]\|}\mathbb{E}[\|\mathbb{L}[k+1]\|^2 - \|\mathbb{L}[k]\|^2|\mathbb{L}[k] = \mathbb{L}[k]], \quad (16)
\end{aligned}$$

where we derive the last step from the fact that $f(x_2) - f(x_1) \leq (x_2 - x_1)f'(x_1) = (x_2 - x_1)/2\sqrt{x_1}$ due to the concavity of the function $f(x) \triangleq \sqrt{x}$ for $x > 0$ [16]. For the sake of conciseness, we omit the frame index $[k]$ in the rest of the proof. Accordingly, the expected Lyapunov drift can be derived as:

$$\begin{aligned}
&\mathbb{E}[\Delta U(\mathbb{L})|\mathbb{L}] \\
&\quad \triangleq \mathbb{E}[\|\mathbb{L}[k+1]\|^2 - \|\mathbb{L}[k]\|^2 | \mathbb{L}[k] = \mathbb{L}] \\
&\quad = \mathbb{E}\left[\sum_{n=1}^{N}(L_n + \mathcal{L}_n - C_n^*\lambda_n)^2 - \sum_{n=1}^{N} L_n^2 | \mathbb{L}\right] \\
&\quad = \mathbb{E}\left[\sum_{n=1}^{N}\left(2L_n(\mathcal{L}_n - C_n^*\lambda_n) + (\mathcal{L}_n - C_n^*\lambda_n)^2\right)|\mathbb{L}\right] \\
&\quad \overset{\text{(i)}}{=} \mathbb{E}\left[\sum_{n=1}^{N}\left(2L_n\sum_{t=1}^{T}(\mathcal{L}_n[k;t] - C_n^*\lambda_n[k;t]) \right.\right. \\
&\qquad\qquad \left.\left. +(\mathcal{L}_n - C_n^*\lambda_n)^2\right)|\mathbb{L}\right] \\
&\quad \overset{\text{(ii)}}{\leq} 2T\sum_{n=1}^{N} L_n\rho_n - 2\sum_{n=1}^{N}\sum_{t=1}^{T}\mathbb{E}[L_n C_n^*\lambda_n[k;t]|\mathbb{L}] + \mathbb{B}_1,
\end{aligned} \tag{17}$$

where ρ_n is the average traffic load of all newly arrived flows in the cluster n per time slot. In eq. (17), we derive step (i) from the definitions of $\mathcal{L}$ and λ_n (see Section IV-A.2). Since the newly arrived traffic load is independent of the present system state, step (ii) is true for $\mathbb{B}_1 \triangleq N(\mathcal{L}_{max}^2 + T^2)$ and $\mathcal{L}_{max} \triangleq T_n^{max}\mathcal{F}_n^{max}\lceil F_n^{max}/s_n^{max}\rceil$ in eq. (17).

We now analyze the first term on the right hand side of step (ii) in eq. (17). Based on our Algorithm 1 and the fact that the data traffic of the newly arrived flows in each cluster can be at most 1, we have:

$$2T\sum_{n=1}^{N} L_n\rho_n \overset{\text{(a)}}{\leq} -2\epsilon\sum_{n=1}^{N} L_n + 2T\sum_{n=1}^{N} L_n C_n^*, \tag{18}$$

where ϵ is the heavy traffic load parameter [16]. Similarly, the second term on the right hand side of step (ii) in eq. (17) can be expressed as:

$$\begin{aligned}
&2\sum_{n=1}^{N}\sum_{t=1}^{T}\mathbb{E}[L_n C_n^*\lambda_n[k;t]|\mathbb{L}] \\
&\qquad \geq 2\left(1 - \frac{1}{2T}\epsilon\right)\sum_{n=1}^{N} L_n C_n^* T - \mathbb{B}_2 \\
&\qquad \geq \epsilon\sum_{n=1}^{N} L_n + 2T\sum_{n=1}^{N} L_n C_n^* - \mathbb{B}_2,
\end{aligned} \tag{19}$$

where $\mathbb{B}_2 \triangleq 2T\lceil F_n^{max}/s_n^{max}\rceil\sum_{n=1}^{N}(N_n^{max} + T\mathcal{F}_n^{max})$. By substituting eq. (18) and eq. (19) into eq. (17), the expected Lyapunov drift can be derived as:

$$\begin{aligned}
\mathbb{E}[\Delta U(\mathbb{L})|\mathbb{L}] &\leq -\epsilon\sum_{n=1}^{N} L_n + \mathbb{B}_1 + \mathbb{B}_2 \\
&\leq -\epsilon\|\mathbb{L}\| + \mathbb{B}_1 + \mathbb{B}_2.
\end{aligned} \tag{20}$$

The above eq. (20) is true as $\|\mathbb{L}\|_1 \geq \|\mathbb{L}\|$. By substituting eq. (20) into eq. (16), the expected Lyapunov drift can be expressed as:

$$\mathbb{E}[\Delta V(\mathbb{L})|\mathbb{L} \leq -\frac{\epsilon}{2} + \frac{\mathbb{B}_1 + \mathbb{B}_2}{2\|\mathbb{L}\|}. \tag{21}$$

Here, eq. (21) implies that if the value of Lyapunov function $V(\mathbb{L}) = \|\mathbb{L}\|$ is sufficiently large, its conditional expected drift is strictly negative. This indicates that our proposed time slot allotment algorithm is throughput optimal. Theorem 2 is thus proved. □

VI. Performance Evaluation

In this section, we present our evaluation results for DROVE under realistic application scenarios. We compare DROVE with the thee benchmarks, namely, DOTS [25], SIL [29] and SOCS [30]. Additionally, we implement a dynamic version of DROVE, called D-DROVE, where UAVs fly horizontally at different altitudes and use a light-weight message passing method [38] to share location information. Section VI-A discusses the experimental setup. We then present the experimental results in Section VI-B.

A. Experimental Setup

We evaluate all schemes on realistic network settings using an integrated UAV network simulator–FlyNetSim [42]. The core components of FlyNetSim are NS-3 and Ardupilot. Particularly, we use NS-3 to emulate various realistic models, e.g., ray-tracing channel, traffic generation and mobility. In contrast, we use Ardupilot to model various characteristics of UAV, e.g., navigation, control and mission planning, and Micro Air Vehicle Link protocol [43] stack as a communication protocol. To implement a real-world scenario, the position and velocity of an intruder and UAV are derived following the method in [24]. The amount of traffic flows arriving at any cluster in each time slot follows the Bernoulli distribution with mean 0.05. We assume that once an event of interest is captured by a UAV, it generates packets at the rate of 250 kbps (constant bit rate) and transmits over bandwidth 1 MHz to the GS. We also assume that the coordinates of the UAVs follow a 2D Gaussian distribution with standard deviation in both x-axis and y-axis as 3, and mean as 0. The energy model parameters are used as $E_{elec} = 50$ nJ/bit, $R_f = 0.8$, $k_t = 55\%$ and $\epsilon_{fs} = 0.1$ nJ/bit/m^2 [36]. We consider two propagation segments for exposition: LoS and NLoS. The model parameters used in the simulation are $\alpha_1 = 2.27$, $\beta_1 = -40$, $\alpha_2 = 3.64$, $\beta_2 = -30$, $\sigma_1^2 = 1$ and $\sigma_2^2 = 3$ [33]. The initial battery energy of the UAV is 14000 mAh. The altitude of UAV in D-DROVE changes from 50 m to 100 m. We consider a UAV with similar specifications as DJI Phantom 4 Pro [44]. Unless specified otherwise, we listed all the parameters and their corresponding values used during the experiment in Table II. We perform extensive simulation with a 95% confidence level and each data point in the figures is computed by averaging the results of 100 independent runs.

TABLE II
KEY PARAMETER VALUES USED IN THE SIMULATION

Parameter	Value	Parameter	Value
Network area	25×25 km^2	Drone speed	30 m/s
l	200 kB	Intruder moving speed	1.4 m/s
H	100 m	σ^2	-110 dBm
B	1 MHz	G_0	-50 dB
M	40	Traffic load	5 messages/s
Size of computing queue	20	Transmission range	2 km

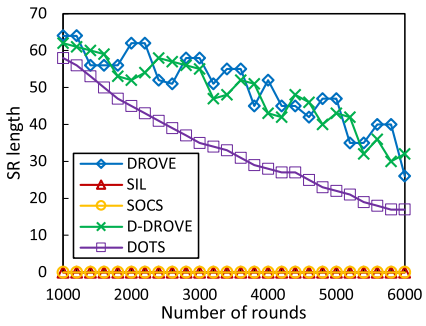


Fig. 4. Super round length versus number of rounds.

B. Performance Comparison

In this section, we discuss the performance of DROVE and the comparative results with D-DROVE, DOTS, SIL and SOCS. We consider to use the SR length, energy consumption, clustering overhead, throughput, end-to-end delay, flow success rate and packet drop rate as the appropriate evaluation metrics.

1) Clustering Task Scheduling: We plot the dynamic changes of the SR length with the number of rounds in Figure 4. For both DROVE and D-DROVE, the plot shows that the SR length is long at the beginning of the simulation time. Particularly, the first SR length is 62 and 57 rounds in DROVE and D-DROVE, respectively. It signifies that the UAV selected as CH in the first round can perform the CH role for 62 rounds in DROVE. Interestingly, we observe that varying altitude in D-DROVE has an effect on the SR length. The plot also reveals that the SR length is more stable in both DROVE and D-DROVE than DOTS because the reclustering triggering factors in DOTS is not prudent. We further notice that the SR length gradually decreases with the increase in the number of rounds. It is because, as the simulation time increases, the UAV's residual energy decreases, resulting in reduction of reclustering interval to evenly distribute the traffic load. Additionally, we measure the average number of reclustering up to 6000 rounds for both SIL and SOCS. As expected, the plot shows that the DSRP considerably reduces DROVE's and D-DROVE's reclustering number to 69 and 95, respectively resulting in the reduction of the clustering overhead. However, this value is 2703 in DOTS. Moreover, for SIL and SOCS, this value is significantly high, i.e., 6000, since both SIL and SOCS recluster the network in every round.

2) Energy Conservation: We present the experimental results in terms of energy conservation in Figure 5. Precisely, Figure 5(a) illustrates the amount of energy consumed by the UAVs only during the clustering process. Note that the energy consumed for executing Algorithm 2 is included in both DROVE and D-DROVE while plotting. The plot exhibits that the best performance is achieved by DROVE, followed by D-DROVE, DOTS, SIL and SOCS. The primary reason for this significant reduction of energy consumption in DROVE's plot is that the number of reclusterings triggered by DSRP and the number of messages exchanged in the clustering operation are considerably reduced. However, the differences in energy consumption between DROVE and D-DROVE is due to the differences in SR length for varying altitude in D-DROVE. Whereas both SIL and SOCS consume significant energy because the network-wide broadcasting of control messages during the clustering process.

Figure 5(b) illustrates the total energy consumed per increasing number of rounds. As Figure 5(b) shows, as the number of round increases, the total energy consumption increases linearly since more number of UAVs participate to serve higher traffic flow. However, the performance of DROVE is superior than DOTS, SIL and SOCS. This is because, the cluster formation mechanism, time slot allotment, and reclustering at suitable times with DSRP are the main energy conservation attributes of DROVE. We observe that DROVE reduces the total energy consumption by 16.47%, 36.32%, 41.80% and 54.48% compared to D-DROVE, DOTS, SIL and SOCS, respectively.

Lastly we verify the clustering overhead of the five schemes in Figure 5(c). We can see that, as the number of round increases, the clustering overhead remains stable in all the schemes. However, the differences in clustering overhead among the schemes are because each of the scheme follow different principle for CH selection, cluster building and data collection. It can be seen from Figure 5(c) that DROVE outperforms the three benchmarks because the UAVs have more freedom to dynamically adjust the traffic schedule, resulting lesser reclustering. Moreover, D-DROVE exhibits more clustering overhead than DROVE because additional message passing overhead for frequent sharing of location information. We observe that DROVE reduces the clustering overhead by 49.93%, 64.70% and 66.95% compared to DOTS, SIL and SOCS, respectively. In summary, DSRP is suitable for reducing the clustering overhead by managing the reclustering procedure judiciously.

3) Throughput: In this section, we first measure the impact of the deployed number of UAVs on the throughput. Figure 6(a) shows that throughput increases rapidly with the number of UAVs. We observe that the growth rate of the throughput slows down after the number of UAVs reach 30 as the interference becomes significant with the increase of number of UAVs. Figure 6(a) further shows the superiority of DROVE against the three benchmarks as the traffic intensity is suitably handled to reduce the possibility of congestion and packet collision. We notice that both DROVE and DOTS outperform D-DROVE as the varying altitude in D-DROVE increases interference among UAVs. The overall throughput performance of DROVE outperforms D-DROVE, DOTS, SIL and SOCS on average by 12.37%, 10.52%, 23.44% and 34.02%, respectively. Meanwhile, we also measure the throughput under varying traffic arrival rate. As shown in

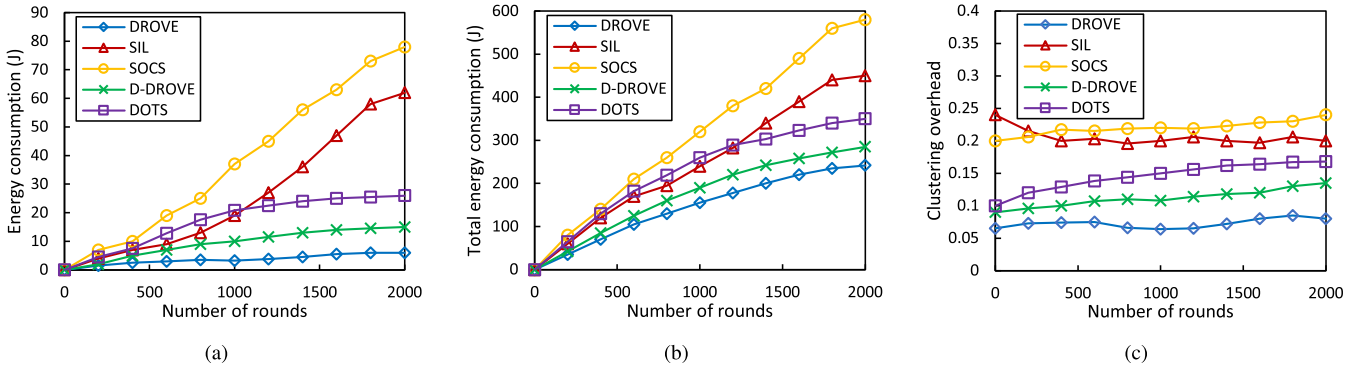


Fig. 5. Energy consumption and clustering overhead performance: (a) Energy consumption during clustering process; (b) Total energy consumption; (c) Clustering overhead.

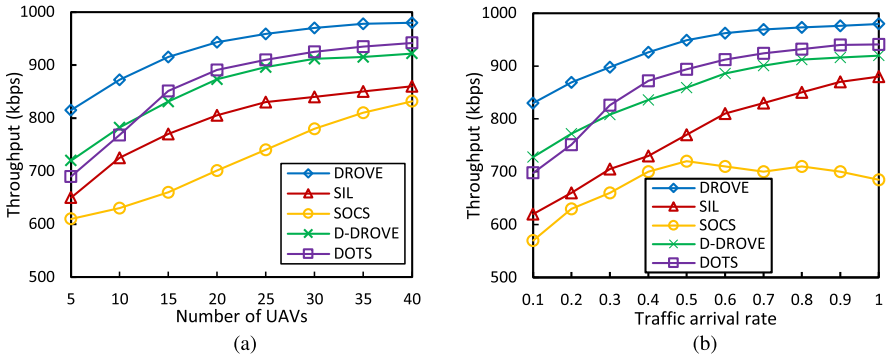


Fig. 6. Throughput performance: (a) Under varying number of UAVs; (b) Under varying traffic arrival rate.

Figure 6(b), DROVE can stabilize the system for any traffic arrival rate. This verifies that the proposed DROVE indeed achieves the maximum system throughout (cf. Theorem 2). Similar to Figure 6(a), DROVE achieves the best performance followed by DOTS, D-DROVE, SIL and SOCS due to the efficient consecutive time slots allotment to the UAVs. The inferior performance of D-DROVE compared to DROVE and DOTS is mainly because of the effect of varying altitude results in imperfect UAV's positions for delivering traffic. Moreover, it can be seen that the throughput of SOCS increases when the traffic arrival rate is low. Once the traffic arrival rate is sufficiently large, the throughput decreases slightly. This is mainly because SOCS uses RRS to assign time slots for a UAV, which might have no data to transmit, resulting in a reduction in throughput.

4) End-to-End Delay: We plot the end-to-end delay of our scheme along with the competing schemes in Figure 7. The plot confirms the performance improvement by using DROVE and D-DROVE. Particularly, both DROVE and D-DROVE can consistently maintain the average end-to-end delay increase to near linear increase, whereas the three benchmarks expand exponentially. We observe that the end-to-end delay in D-DROVE is slightly higher than DROVE as UAV may take a longer time to find a forwardee UAV due to altitude variation for delivering the data to the GS. Moreover, the variations in end-to-end delays among the schemes are due to traffic scheduling and routing, causing different magnitudes of congestion and resulting in

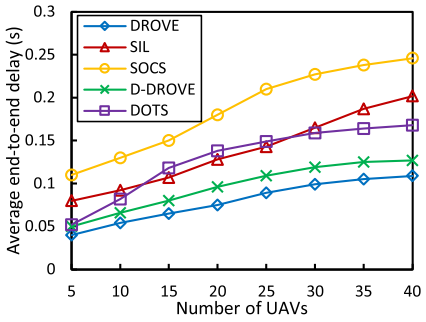


Fig. 7. Average end-to-end delay performance under varying number of UAVs.

different end-to-end delays. We finally observe that the end-to-end delay of DROVE is 17.69%, 38.27%, 49.27% and 61.13% less than that of D-DROVE, DOTS, SIL and SOCS, respectively.

5) Flow Success Rate: Figure 8(a) illustrates the flow success rate under different numbers of UAVs. We notice that the flow success rate of benchmarks, DOTS, SIL and SOCS are decreasing with the increasing number of UAVs as the overhead of the proposed methods prevents them from successful delivering their data traffic, in highly dynamic scenario. As Figure 8(a) shows, the flow success rate of DROVE is on average 17%, 19%, 57% and 79% higher than that of D-DROVE, DOTS, SIL and SOCS, respectively. Besides, we measure the flow success rate under different

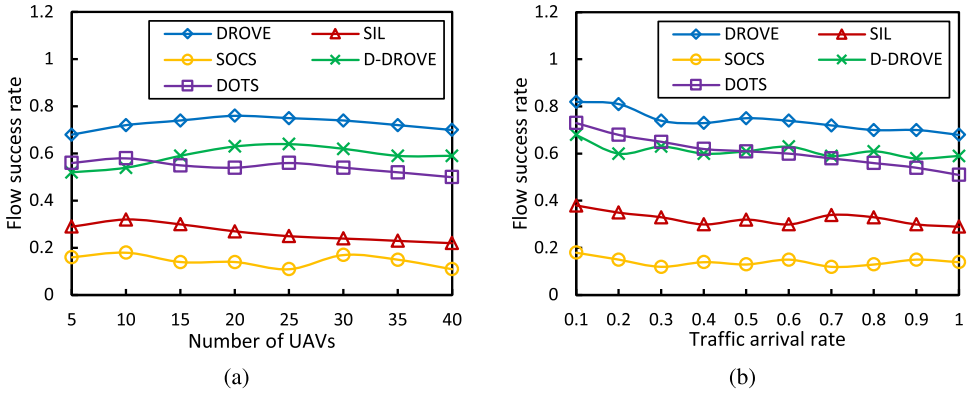


Fig. 8. Performance of flow success rate: (a) Under varying number of UAVs; (b) Under varying traffic arrival rate.

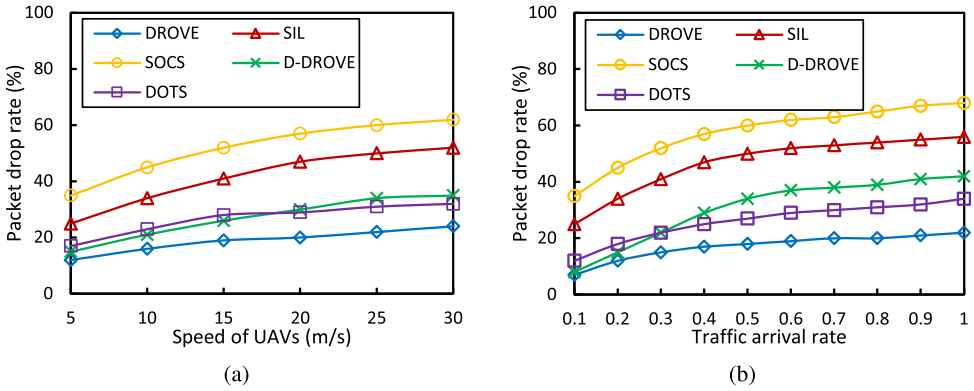


Fig. 9. Performance of packet drop rate: (a) Under varying speed of UAVs; (b) Under varying traffic arrival rate.

traffic arrival rate. As Figure 8(b) shows, as the traffic arrival rate increases, the flow success rate of the five schemes shows a downward trend. However, the flow success rate of DROVE is still highest. Unlike the three benchmarks, as the traffic arrival rate increases, the UAV can efficiently schedule the traffic via DSRP, so the overall flow success rate improves.

6) Packet Drop Rate: Figure 9 shows the evaluation result of the packet drop rate. As Figure 9(a) shows, the average packet drop rate significantly improves by utilizing DROVE, followed by DOTS, D-DROVE, SIL and SOCS due to the efficient consecutive time slots allotment to the UAVs. Specifically, using DROVE, we can keep a more steady packet drop rate increase rate than the three benchmarks. We also notice that D-DROVE exhibits a higher packet drop rate than DROVE as the unfair link-state more dominant in D-ROVE due to varying altitude. Similar to Figure 9(a), Figure 9(b) further shows the superiority of DROVE against the three benchmarks as the traffic intensity is judiciously managed to reduce the packet drop rate. We notice that an increase in the traffic arrival rate seems to burst both SIL and SOCS such that they can not prevent congestion and packet collision. Although, DOTS exhibits better performance than D-DROVE due to improved link-state, fails to beat the performance of DROVE. In summary, we understand the overall efficiency and superiority when applying DROVE to solve task scheduling in UAV networks.

VII. Conclusion

In this paper, we propose DROVE – a dynamic clustering technique for the UAV networks. To conserve energy, DROVE follows a novel dynamic super round policy, where reclustering is triggered only at the required time, i.e., this policy avoids the unnecessary reclustering as performed in RRS. We then analyzed the distributed traffic scheduling problem to optimize throughput in the presence of dynamic traffic flows. Based on our analysis, we developed a scheduling algorithm, where time slots are assigned dynamically to the CMs for achieving optimal throughput. The use of dynamic time allotment not only reduces the clustering overhead, but also conserves energy significantly. Finally, we provide extensive simulation results to validate both the energy conservation and throughput optimality of our proposed algorithm.

In the future, it would be interesting to apply different machine learning and bio-inspired algorithms to minimize clustering problems, like CH selection, cluster formation. We further seek to investigate the trade-off between the maximum residual energy and minimum distance for nominating a CH.

References

[1] C. W. Chen, "Internet of video things: Next-generation IoT with visual sensors," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6676–6685, Aug. 2020.

[2] W. Wang *et al.*, "Placement of unmanned aerial vehicles for directional coverage in 3D space," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 888–901, Apr. 2020.

[3] A. Baltaci, E. Dinc, M. Ozger, A. Alabbasi, C. Cavdar, and D. Schupke, "A survey of wireless networks for future aerial communications (FACOM)," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2833–2884, 4th Quart., 2021.

[4] W. Wang, H. Dai, C. Dong, F. Xiao, X. Cheng, and G. Chen, "VISIT: Placement of unmanned aerial vehicles for anisotropic monitoring tasks," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–9.

[5] A. Rahmati *et al.*, "Dynamic interference management for UAV-assisted wireless networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2637–2653, Apr. 2022, doi: 10.1109/TWC.2021.3114234.

[6] Z. Wang and L. Duan, "Chase or wait: Dynamic UAV deployment to learn and catch time-varying user activities," *IEEE Trans. Mobile Comput.*, early access, Aug. 24, 2021, doi: 10.1109/TMC.2021.3107027.

[7] S.-F. Chou, A.-C. Pang, and Y.-J. Yu, "Energy-aware 3D unmanned aerial vehicle deployment for network throughput optimization," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 563–578, Jan. 2020.

[8] B. Hu, L. Wang, S. Chen, J. Cui, and L. Chen, "An uplink throughput optimization scheme for UAV-enabled urban emergency communications," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4291–4302, Mar. 2022.

[9] N. Bartolini, A. Coletta, and G. Maselli, "SIDE: Self drIving DronEs embrace uncertainty," *IEEE Trans. Mobile Comput.*, early access, Dec. 16, 2021, doi: 10.1109/TMC.2021.3135894.

[10] J. Yoon, A.-H. Lee, and H. Lee, "Rendezvous: Opportunistic data delivery to mobile users by UAVs through target trajectory prediction," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2230–2245, Feb. 2020.

[11] S. T. Muntaha, S. A. Hassan, H. Jung, and M. S. Hossain, "Energy efficiency and hover time optimization in UAV-based HetNets," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5103–5111, Aug. 2021.

[12] D. He, H. Liu, S. Chan, and M. Guizani, "How to govern the non-cooperative amateur drones?" *IEEE Netw.*, vol. 33, no. 3, pp. 184–189, May 2019.

[13] J. Xu, Y. Zeng, and R. Zhang, "UAV-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5092–5106, Aug. 2018.

[14] P. Oettershagen *et al.*, "Perpetual flight with a small solar-powered UAV: Flight results, performance analysis and model validation," in *Proc. IEEE Aerosp. Conf.*, Mar. 2016, pp. 1–8.

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

[16] X. Kong, N. Lu, and B. Li, "Optimal scheduling for unmanned aerial vehicle networks with flow-level dynamics," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 1186–1197, Mar. 2021.

[17] L. Hong, H. Guo, J. Liu, and Y. Zhang, "Toward swarm coordination: Topology-aware inter-UAV routing optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10177–10187, Sep. 2020.

[18] Y. Cui, Q. Zhang, Z. Feng, Z. Wei, C. Shi, and H. Yang, "Topology-aware resilient routing protocol for FANETs: An adaptive Q-learning approach," *IEEE Internet Things J.*, early access, Mar. 29, 2022, doi: 10.1109/JIOT.2022.3162849.

[19] S. Shen, K. Yang, K. Wang, G. Zhang, and H. Mei, "Number and operation time minimization for multi-UAV enabled data collection system with time windows," *IEEE Internet Things J.*, vol. 19, no. 2, pp. 10149–10161, Jun. 2022, doi: 10.1109/JIOT.2021.3121511.

[20] Z. Mou, F. Gao, J. Liu, and Q. Wu, "Resilient UAV swarm communications with graph convolutional neural network," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 393–411, Jan. 2022.

[21] J. Chen, Y. Xu, Q. Wu, Y. Zhang, X. Chen, and N. Qi, "Interference-aware online distributed channel selection for multicluster FANET: A potential game approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3792–3804, Apr. 2019.

[22] X. Zhong, Y. Guo, N. Li, and Y. Chen, "Joint optimization of relay deployment, channel allocation, and relay assignment for UAVs-aided D2D networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 804–817, Apr. 2020.

[23] P. Neamatollahi, M. Naghibzadeh, M.-H. Yaghmaee, and S. Abrishami, "Distributed clustering-task scheduling for wireless sensor networks using dynamic hyper round policy," *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 334–347, Feb. 2018.

[24] F. Tang, H. Hofner, N. Kato, K. Kaneko, Y. Yamashita, and M. Hangai, "A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN)," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 276–289, Jan. 2022.

[25] C. Zhou *et al.*, "Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 911–925, Feb. 2021.

[26] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10187–10200, Dec. 2019.

[27] T. Duan, W. Wang, T. Wang, X. Chen, and X. Li, "Dynamic tasks scheduling model of UAV cluster based on flexible network architecture," *IEEE Access*, vol. 8, pp. 115448–115460, 2020.

[28] A. Hanyu, Y. Kawamoto, and N. Kato, "Adaptive channel selection and transmission timing control for simultaneous receiving and sending in relay-based UAV network," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2840–2849, Oct. 2020.

[29] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in UAV networks for emergency communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8958–8976, Oct. 2019.

[30] A. Khan, F. Aftab, and Z. Zhang, "Self-organization based clustering scheme for FANETs using glowworm swarm optimization," *Phys. Commun.*, vol. 36, Oct. 2019, Art. no. 100769.

[31] M. R. Brust, M. I. Akbaç, and D. Turgut, "VBCA: A virtual forces clustering algorithm for autonomous aerial drone systems," in *Proc. Annu. IEEE Syst. Conf. (SysCon)*, Apr. 2016, pp. 1–6.

[32] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1274–1285, Jun. 2020.

[33] J. Chen, U. Yatnalli, and D. Gesbert, "Learning radio maps for UAV-aided wireless networks: A segmented regression approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[34] R. Essaadali and A. Kouki, "A new simple unmanned aerial vehicle Doppler effect RF reducing technique," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 1179–1183.

[35] P. Zhou, X. Fang, Y. Fang, R. He, Y. Long, and G. Huang, "Beam management and self-healing for mmWave UAV mesh networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1718–1732, Feb. 2019.

[36] M. Thammawichai, S. P. Baliyarasimhuni, E. C. Kerrigan, and J. B. Sousa, "Optimizing communication and computation for multi-UAV information gathering applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 2, pp. 601–615, Apr. 2018.

[37] A. Ghosal, S. Halder, and M. Conti, "DISC: A novel distributed on-demand clustering protocol for internet of multimedia things," in *Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2019, pp. 1–9.

[38] S. Hosseinalipour, A. Rahmati, and H. Dai, "Interference avoidance position planning in dual-hop and multi-hop UAV relay networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7033–7048, Nov. 2020.

[39] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019.

[40] K. Guo, X. Li, and L. Xie, "Ultra-wideband and odometry-based cooperative relative localization with application to multi-UAV formation control," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2590–2603, Jun. 2020.

[41] A. Ghosal, S. Halder, and S. K. Das, "Distributed on-demand clustering algorithm for lifetime optimization in wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 141, pp. 129–142, Jul. 2020.

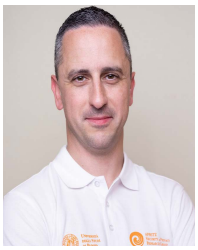
[42] S. Baidya, Z. Shaikh, and M. Levorato, "Flynetsim: An open source synchronized uav network simulator based on ns-3 and ardupilot," in *Proc. 21st ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst. (MSWiM)*, Oct. 2018, pp. 37–45.

[43] L. Meier *et al. Mavlink: Micro Air Vehicle Communication Protocol*. Accessed: Jul. 20, 2020. [Online]. Available: http://qgroundcontrol.org/mavlink/start

[44] *PHANTOM 4 PRO Data Sheet*. Accessed: Jul. 15, 2020. [Online]. Available: https://www.dji.com/it/phantom-4-pro/info

Subir Halder received the Ph.D. degree in computer science and technology from the Indian Institute of Engineering Science and Technology, India, in 2015. He has worked as an Assistant Professor with the Department of CSE, Dr. B. C. Roy Engineering College, India, from 2007 to 2017. He is currently a Marie Skłodowska Curie Fellow with the University of Limerick, Ireland. Prior to that, he was a Post-Doctoral Researcher with the University of Padua, Italy. His research interests include security and privacy in cyber physical systems, the IoT, autonomous vehicle, controller area network, and industry 4.0. He has coauthored more than 40 papers in reputed international peer-reviewed conferences and journals in these fields.

Amrita Ghosal received the Ph.D. degree in computer science and engineering from the Indian Institute of Engineering Science and Technology, India, in 2015. After her Ph.D. degree, she was a Post-Doctoral Researcher with the Department of Mathematics, University of Padua, Italy. She is currently a Marie Skodowska-Curie Fellow with the Department of Computer Science and Information Systems, University of Limerick, Ireland. Her research interests include security and privacy for mobile and wireless networks. Particularly, she is interested in detection, prevention, and mitigation of different DoS style attacks for smart grid, V2X, connected vehicle, cyber-physical systems, and the IoT. In these areas, she has published more than 40 papers in high quality journals and refereed conference proceedings. She also has coauthored a number of book chapters.



Mauro Conti (Fellow, IEEE) received the Ph.D. degree from the Sapienza University of Rome, Italy, in 2009. After his Ph.D. degree, he was a Post-Doctoral Researcher with Vrije Universiteit Amsterdam, The Netherlands. In 2011, he joined as an Assistant Professor with the University of Padua, Italy, where he became an Associate Professor in 2015 and a Full Professor in 2018. He has been a Visiting Researcher with GMU in 2008 and 2016; UCLA in 2010; UCI in 2012–2014 and 2017; TU Darmstadt in 2013; UF in 2015; and FIU in 2015, 2016, and 2018. He is currently a Full Professor with the University of Padua and an Affiliate Professor with the University of Washington, Seattle, USA. His main research interests include security and privacy. In this area, he has published more than 250 papers in topmost international peer-reviewed journals and conferences. He is a Senior Member of ACM. He has been awarded with a Marie Curie Fellowship by the European Commission in 2012 and a Fellowship by the German DAAD in 2013. His research is also funded by companies, including Cisco, Intel, and Huawei. He is an Area Editor-in-Chief for IEEE COMST and an Associate Editor for several journals, including IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (COMST), IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS), IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC), and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT (TNSM). He was the Program Chair of TRUST 2015, ICISS 2016, and WiSec 2017. He was the General Chair of SecureComm 2012 and ACM SACMAT 2013.