doi.org/10.1016/j.comnet.2019.107038

Distributed Aerial Processing for IoT-Based Edge UAV Swarms in Smart Farming

Anandarup Mukherjee[1], Sudip Misra

Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India, 721302

Anumandala Sukrutha

Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Telangana State, India, 502285

Narendra Singh Raghuwanshi

Department of Agriculture & Food Engineering, Indian Institute of Technology Kharagpur, India, 721302

Abstract

This work addresses the challenges of a decentralized and heterogeneous Unmanned Aerial Vehicle (UAV) swarm deployment – some fitted with multimedia sensors, while others armed with scalar sensors – in resource-constrained and challenging environments, typically associated with farming. Subsequently, we also address the resulting problem of sensing and processing resource-intensive data aerially within the Edge swarm in the fastest and most efficient manner possible. The heterogeneous nature of the Edge swarm results in under-utilization of the available computation resources due to unequal data generation within its members. To address this, we propose a Nash bargaining-based weighted intra-Edge processing offload scheme to mitigate the problem of heavy processing in some of the swarm members. We do this by distributing the data to be processed to all the swarm members. Real-life hardware tuned simulation of a large UAV swarm shows that by increasing the number of UAVs in the swarm, our scheme achieves better scalability and reduced processing delays for intensive processing tasks. Additionally, in comparison to regular star and mesh topologies, our scheme achieves an increase in collective available network processing speeds by 100% for only 25% of the number of UAVs in a star topology.

Keywords: UAV swarm, collaborative processing, aerial mesh network, heterogeneous swarm, Edge computing, smart farming.

1. Introduction

Internet of Things (IoT) is in the process of revolutionizing agriculture through smart farming. The involvement of IoT in farming applications such as precision agriculture, livestock management, inventory management, and others has increased the productivity, yield, and raised economic benefits to farmers through connected sensors, actuators, and networked systems. UAVs – one of the prime examples of such networked system – has become quite popular in smart farming applications,

Email address: anandarupmukherjee@gmail.com (Anandarup Mukherjee)

with applications ranging from monitoring of crop health, farmland demarcation and mapping, to spraying fertilizers and pesticides periodically and autonomously.

Networked UAVs [1] are in extensive use for a range of solutions with far-reaching implications in the domains of agriculture, remote sensing, surveillance, security, law enforcement, disaster management [2], and others. Most of these domains deal with UAV-based multimedia data for various tasks such as target tracking, information gathering, and path planning [3]. The real-time processing of multimedia data in constrained environments is an inherent problem, which is often encountered by UAVs in precision agriculture tasks. The data gathered from the farmlands, as well as the ones generated within the UAVs for its flight controls and navigation, are quite massive. Commonly, the data is stored within the UAVs and retrieved later for processing and analysis. However, this results in a loss of real-timeliness, which also prevents the implementation of complete UAV automation for agricultural practices. The biggest challenge faced during the implementation of a real-time UAV-based sensing solution by making use of multimedia data is the low computational power and limited energy resources of these UAVs.

Various solutions are proposed to address the problems of low computation capability of such UAVs. Solutions such as cloud-based data processing offloading from single UAVs [4], processing offloading from a UAV to a ground server [5], and others [6] offer limited respite from the challenges at hand as these are heavily dependent on network connectivity, bandwidth, and quality of service for reliable and timely operation. Additionally, the areas of implementation of such multi-UAV networked solutions may not always promise the availability of network connectivity, network quality, or bandwidth, especially in applications involving operations in remote and infrastructure-constrained applications such as agriculture and disaster management.

UAV deployment strategies for farming applications such as crop monitoring, field surveys, and others range from a single standalone powerful UAV to swarm of smaller, less powerful UAVs working in tandem. However, the use of multiple smaller UAVs has proved to be more efficient than a single large one regarding scalability, survivability, speed, cost, and bandwidth requirements [5]. Star and mesh network configurations are the commonly used topologies used for multi-UAV networks. In a star topology formation, each UAV connects to a central UAV, which, however, restricts direct communication between the UAVs in the network. Whereas, multi-UAV networks following a mesh topology allow direct or hop-based intra-member UAV to UAV communication within the network, however, at the cost of increased network load and traffic [5].

(a) Sensing and communication range

(b) A swarm of autonomous UAVs in flight

(c) Aerial imagery of agricultural plots

Figure 1: Edge UAV swarm-based operations and its possible applications in smart farming.

In this work, we propose a two-pronged approach to address the need for time-critical observation and tracking of ground-based tasks such as crop health and stress monitoring, farmland mapping (refer to Fig. 1(c)), and others by a heterogeneous collaborative sensing approach, which uses both multimedia and scalar sensor armed UAVs in the swarm (refer to Fig. 1(b)). We additionally devise a scheme to mitigate the processing overheads of each swarm member, essentially an Edge computing platform, using distributed collaborative processing within the Edge UAV swarm itself.

Assumption 1. *A single Edge UAV in the swarm is equipped with a camera, whereas the other swarm members are equipped only with scalar sensors.*

Assumption 2. *A UAV with a camera sensor (which can be an RGB, thermal, or a multispectral camera) has a sensing range of $a \times b$, which is much larger than that of a UAV with a scalar sensor with a sensing range of only $a/2 \times b/2$. We consider the scalar sensing range as a single grid location in this work. $\mathcal{L}_i$ represents the i^{th} grid location covered by a UAV.*

Fig. 1(a) shows the sensing and communication range of multiple UAVs in the swarm. In the aerial plane, the central UAV – node 0 – consists of a camera sensor, whereas the other UAVs (nodes 1-10) consist of scalar sensors only. A much broader search area can be covered by either scaling-up the Edge-based swarm or by using multiple such Edge-based swarms. Additionally, we consider that a UAV with the camera can visually search 4 grids at the same time the other scalar sensor armed UAVs take to search a single grid each. For rs_i denoting the sensing range of the i^{th} UAV in the swarm at any instant of time, the camera-armed UAV's sensing range $rs_0 = a \times b$, whereas the scalar sensor fitted UAV's sensing range $rs_{1-7} = a/2 \times b/2$. The communication range between a source (S) and destination (D) UAV node rc_{SD} of each UAV is limited to one hop within its immediate neighborhood, beyond which, the UAV has to communicate via an intermediate UAV in a multi-hop manner.

Assumption 3. *In a k UAV system, the UAVs never search the same grid twice, nor do other UAVs sense the grid locations already covered by a UAV such that $\bigcap_k \left\{ \bigcap_{i=1}^{n^2} \mathcal{L}_i \right\} = \emptyset, \quad \forall 0 < k \leq n^2$*

Assumption 4. *Each UAV in the network is assumed to have two wireless access points – one for receiving the data and the other for sending the data. Once the image is processed in the assigned UAV, it returns the coordinates of the detected object to the central UAV of the swarm.*

Definition 1. *We consider the swarm of Edge UAVs in this work to be heterogeneous due to the presence of a unique sensor type on each UAV. Additionally, the sensors can be scalar, as well as multimedia ones.*

1.1. Heterogeneous Collaborative Sensing

We consider a $n \times n$ observation area consisting of equally divided grids. If a single UAV-based visual coverage/observation/remote sensing of an area takes x units of time in a single grid, the time taken to cover the whole observation area by a UAV becomes n^2x units of time, which results in worst-case time complexity of $T(n) = O(n^2)$. In contrast, having as many UAVs as the number of search grids incurs a time complexity of $O(1)$. However, this approach is infeasible for large deployments. Along similar lines, the use of UAVs fitted with scalar sensors for remote sensing tasks ushers in worst-case time complexity of $T(n) \simeq O(n^3)$ as it needs to sense in a 3-dimensional space due to the insufficient sensing range of these sensors. Despite the low data volume generated from these sensors, the search time of this approach is infeasible for use in time-critical tasks, except in vast numbers, which again makes the proposed approach infeasible.

We, therefore, propose the use of a heterogeneous swarm of UAVs for accomplishing the search task in a relatively time-efficient manner by making use of the benefits of both UAV-based multimedia, as well as scalar sensing. We attribute the heterogeneous nature of each Edge swarm to the presence of different sensors on each swarm member – either multimedia or scalar. Each of these individual members of the swarm performs individual sensing tasks to achieve the more massive collective task of a time-efficient observation of an area or in a collaborative manner. Additionally, the use of multiple UAVs in farmland sensing provides resilience against individual UAV failures. However, this heterogeneity creates some unique issues such as the problem of the ratio of UAVs with multimedia sensors and UAVs with scalar sensors. Additionally, this heterogeneity also results in the problem of unequal data-rate and data-volume from each swarm member, resulting in various degrees of processing under-utilization and over-utilization within the Edge swarm. Considering Δ_l is the data generated from the UAV camera per second for a frame rate of f_{acc}, and a frame size of δ_l, the data load per second from this UAV can be expressed as $\Delta_l = \delta_l \times f_{acc}$. We summarize the whole problem as processing Δ_l in the least time possible within the UAV swarm.

1.2. Distributed Collaborative Processing

To address our problem statement, we propose an intra-swarm distributed processing scheme for mitigating the processing load from the multimedia Edge UAV node. The UAV with camera sensors offloads the majority of its processing onto other swarm members, which as per our implementation scenario, have a relatively lesser processing load on them due to the integration of scalar sensors only. Previously, the distributed processing of computationally intensive tasks has been performed with multicore parallelism and coprocessing on GPUs [7], and division of datasets for simultaneous processing on multicore processor architectures [8] with very promising reports of computation speed-ups and energy conservation. However, these approaches do not consider a highly mobile and resource-constrained environment such as the one in our case, in which processing and even data-offloading become significant factors in deciding the offload targets.

In this work, we distribute the captured video frames to other swarm members for processing. Each of these swarm members has similar processor specifications. As the member UAVs do not have a camera sensor to process their data, each of the member UAVs processes the data offloaded to them for processing, besides their regular and comparatively low-scale processing and scalar sensing tasks. If t_{UAV} is the amount of time required to process Δ_l, then for a k UAV swarm,

$$t_{UAV}(k) = \frac{\Delta_l}{k} + \sum_{i=1}^{k} C_i + \sum_{i=1}^{k-1} \tau_i \tag{1}$$

In equation 1, C_i is a constant representing the internal processing time of the i^{th} UAV, and τ_i is the delay incurred during the transfer of one frame from one UAV to another in a single hop. To maximize processing throughput from each UAV processor by minimizing $\sum_{i=1}^{k} C_i$ we estimate the average processing wait times for the images at each UAV node from their respective queue properties.

Additionally, based on the distribution of the traffic flow in the deployed network, and the resources available at each UAV node, we formulate a joint utility function for the UAV nodes in the swarm. A Nash bargaining solution is applied to the utility function to strategize the distribution of acquired video frames from the multimedia UAV with the camera to the other UAV nodes in the swarm before deployment. This approach allows the setting of an optimum frame rate of video capture, the swarm size, and even the communication architecture of the swarm. Finally, we compare the results obtained to various star and mesh topologies. Our approach shows positive results regarding processing speed-ups, as well as scalability of deployment.

1.3. Contributions

In this work, we establish a viable means of time-critical remote sensing of ground plots and crops in smart farming. We propose the use of heterogeneous Edge UAVs in a swarm formation to remotely sense a given zone – some using camera sensors, while the others using scalar sensors. The unequal data-load generated and subsequently the processing load on the UAVs in the swarm, due to the heterogeneous nature of this swarm, is mitigated by a Nash bargaining game to achieve significant processing speed-ups and enhance the scalability of the system. The main contributions of this work are:

1. A proposition for the use of heterogeneous UAV swarm consisting of mixed UAVs armed with either scalar or multimedia sensors, jointly performing remote sensing over farmlands, is put forward.
2. A distributed multimedia data processing approach for mitigating the processing load of a few swarm members to the whole swarm is proposed to contain the processing within the Edge itself.
3. A Nash bargaining based game is proposed to decide the intra-swarm offload architecture such that for a given number of UAVs, the optimized offload architecture formed aims to minimize processing lag, reduce the offload delay times, and allocates maximum processing resources to the multimedia data offloaded.
4. An evaluation hardware consisting of four UAVs in a swarm is setup. The communication, time, and energy metrics measured from the hardware is used for emulating the behavior of our proposed approach for a large Edge swarm.

2. Related Works

The use of UAVs and UAV swarms has been explored for a multitude of tasks such as tracking [9], path planning, and other communication aspects within [10], and outside the swarm [11]. Concerning the objectives being pursued in this work, we divide the related works into three groups – 1) Heterogeneous Collaborative Sensing, 2) UAV swarms in sensing and tracking, and 3) Distributed processing in highly mobile environments.

2.1. Heterogeneous Collaborative Sensing

Heterogeneous collaborative sensing, although challenging, has been used for achieving resource-efficient results as compared to traditional approaches. Typically, collaborative sensing has been used for spectrum sensing and robotic swarms. Collaborative spectrum sensing has been used for tasks such as radio resource allocation [12], estimating the global spectrum states [13], and others. Further, approaches such as *EasiSee* [14], which is a WSN-based real-time vehicle identification system, report achieving a reduction in overall energy consumption through collaborative sensing using heterogeneous sensors. Collaborative sensing, especially using heterogeneous sensors, are also commonly encountered in the domain of robotics and multi-robot sensor networks. Platforms such as SENORA [15] and other middlewares [16] enable peer-to-peer networking and collaboration amongst mobile robotic entities.

2.2. UAV Swarms in Sensing and Tracking

Works on UAV swarm-based tracking of targets on the ground, especially moving targets, present solution approaches to a very challenging problem of target localization, which has huge implications in real-life scenarios such as farming, surveillance, and disaster management. UAV swarm-based searching involves cooperative search and tracking for targets, which may be RF-based sources [9], vehicles, or even humans. These tasks involve precision in path planning and flawless coordination amongst swarm members. Works by Nigam *et al.* [17] and Pitre *et al.* [18] successfully address some of the challenges related to control and path planning for search and track missions respectively.

Nigam *et al.* [17] propose high-level aircraft control strategies, control policies for compensating dynamic aircraft constraints, and health-and-endurance monitoring policies for control of multiple UAVs during persistent surveillance. In contrast, Pitre *et al.* [18] take an information value approach for path planning in UAV-based joint search and track missions. Their work relies on a modified particle swarm optimization approach for optimizing the trajectory of the UAV to maximize the targets searched. Additional tasks directly associated with multiple UAV-based searching involves increasing spatial coverage distribution of sensing [19] as well as addressing connectivity management issues in UAV networks [11].

2.3. Distributed Processing in Highly Mobile Environments

Processing offloading from low-power devices to more powerful ones is one of the widely addressed topics in the domain of distributed computing and processing. However, specific persistent issues arise while addressing the task of processing offloading [20] in mobile environments such as scalability [21] [22], bandwidth management [23], and resource allocation. Various approaches addressing scalability issues of distributed processing in mobile environments include those by Gedik and Liu [21], where they propose a distributed architecture in conjunction with their optimization techniques to address scalable processing challenges of continuously moving location queries. Their approach reports significant server load and messaging cost savings in comparison to traditional central processing approaches.

Similarly, Wu *et al.* [22] propose the use of ADDSEN, a middleware developed by them for urban sensing using adaptive data processing and dissemination in UAV swarms. An online learning approach periodically adjusts the broadcast rate and knowledge loss rate, whereas a strategy function guides the state transitions of link status changes. Other approaches addressing various challenges in distributed processing for highly mobile environments include the use of Markov chain-based pattern prediction, and subsequent passive bandwidth management in QoS optimization for vehicular networks, and maximizing Markovian network utility functions of multi-server systems and networks in which each user may be granted resources by different servers [24].

2.4. Synthesis

Various works in the realm of UAV-based aerial sensing tasks rely mainly on homogenous sensing platforms, which either incur massive delays in sensing (e.g., scalar sensors) or massive delays due to processing (e.g., multimedia sensors), even when they are used in swarms. Typically, heterogeneous and collaborative sensing rely on a central controller or server for coordinating the sensing and collaboration. A huge majority of these approaches do not consider the network or real-time processing requirements of the collaborating members. Additionally, the offload of processing requirements to other members in a swarm or more powerful processors is also biased regarding network bandwidth considerations. A considerable majority of the works related to processing offloading do not even consider the resource-constrained nature of the network or the swarm itself, where it may not always be possible to offload data to remote locations over high-speed networks or have multiple high-speed mobile processors. Our proposed approach of a heterogeneous collaborative Edge UAV swarm-based tasks, aimed mainly at smart farming, makes use of the benefits of both scalar and multimedia sensing. Our approach speeds up the time taken to sense large swathes of farmlands, and the Nash bargaining based distributed processing within the swarm takes care of the high data and processing load generated due to the multimedia sensors in the swarm.

3. System Architecture

A one-hop UAV data-offload architecture consists of a central UAV to which m UAVs can connect. The UAVs can communicate with each other in a star or mesh configuration for achieving distributed

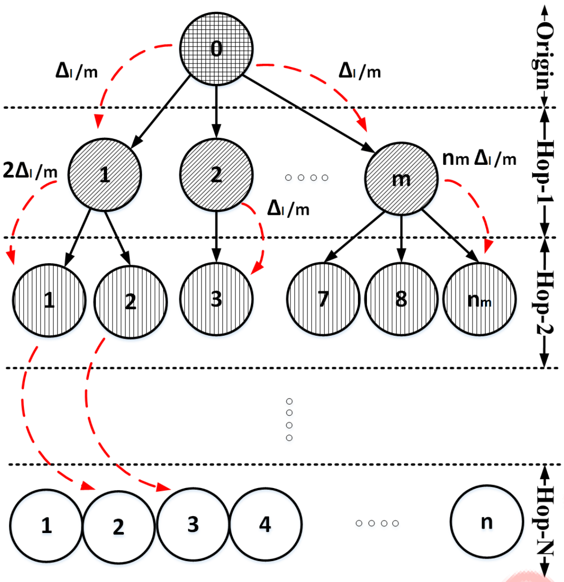


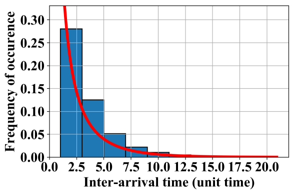
Figure 2: A representation of the multi-hop offload connection.

aerial swarm-based processing. As the connection between the UAVs is established wirelessly, each UAV connected to a central node puts a certain amount of strain on its resources. This connection-based strain on the UAV's resources is attributed to the resources consumed for maintaining the radio connection. If R_a is considered to be the total available resources at the central UAV node (node-0), then initially at $t_0|_{t=0}$ when no UAVs are connected to the central UAV node, we have $t_0 \propto R_a^{-1} \Rightarrow t_0 = \frac{K}{R_a}$ such that K is the constant of proportionality. For a k UAV system, let each UAV connection to the central UAV put a constraint on the central UAV node's resources by a factor of γ_k such that over a period, the resources consumed at the central UAV node R_c is denoted as $R_c = \gamma_1 + \gamma_2 + \cdots + \gamma_{k-1} = \sum_{i=1}^{k-1} \gamma_i$. Similarly, at $t_k|_{t>0}$, for $k-1$ UAVs connected to a central UAV node, we represent t_k as:

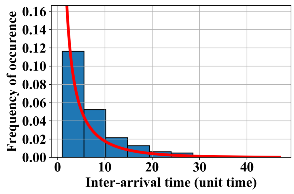
$$t_k = \frac{R_a}{R_a - R_c} t_0 \tag{2}$$

Assumption 5. *The $m-1$ UAVs connecting to a central UAV node in a m UAV system puts identical constraints on the central node's resources such that $\gamma_1 = \gamma_2 = \cdots = \gamma_{m-1} = \sum_k \gamma$.*

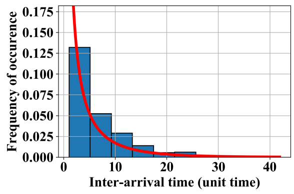
In a one-hop star connected network, n nodes connect to a central node, each contributing a lag Δ/n to the overall lag Δ of the system. The only difference between the star and mesh connected networks during distributed data processing offload is that in a star connection only the central UAV exhausts its resources with an increasing number of connections to it over a period, while in a mesh connection all nodes run out of resources at a point of time. In continuation, each UAV in a hop in a multihop UAV network approach may be connected to a few other UAVs in the next hop, however within a unit-hop distance of each other, as shown in Fig. 2. It is pertinent to mention that Fig. 2 is architecturally similar to the concept of distributed processing denoted in Fig. 1(a). Similar to the one-hop network architecture, every connection to a UAV in the multihop configuration induces a lag in that UAV's processing resources as a result of the operations required to maintain the wireless connection to the connecting UAV.



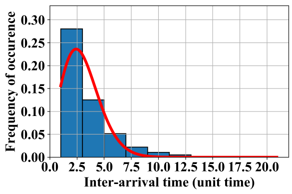
(a) Logarithmic distribution on UAV Node 0



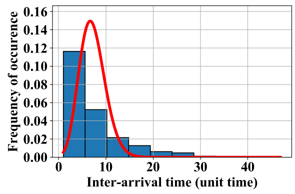
(b) Logarithmic distribution on UAV Node 1



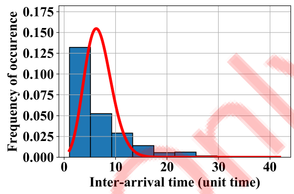
(c) Logarithmic distribution on UAV Node 2



(d) Poisson distribution on UAV Node 0



(e) Poisson distribution on UAV Node 1



(f) Poisson distribution on UAV Node 2

Figure 3: Fitting Poisson and Logarithmic distributions to inter-arrival times at various UAV nodes in the network and their corresponding Chi-squared parameter.

4. UAV Swarm Network Traffic Analysis

A multi-hop network architecture with randomized connections between the UAVs in the network is simulated, which is subject to the constraint of one-hop communication between the immediately communicating nodes. Video frames captured from the origin UAV node are allotted for processing to each immediately one-hop neighboring UAV based on the number of UAVs it is one-hop connected to and the number of the video frames already waiting to be processed by that UAV. The inter-arrival time IA for video frames arriving at every UAV in this network is calculated. The data traffic being discrete and multi-valued (not binary) is fit using Poisson and Logarithmic distributions to estimate the nature of the traffic in this network. For an event rate of λ in a network following Poisson distribution, the probability mass function (PMF) is given by:

$$f(\lambda, x) = \frac{\lambda^x e^{-\lambda}}{x!}\Big|_{x=0,1,2,\cdots} \quad \forall\ \lambda > 0 \tag{3}$$

Similarly, the PMF of a logarithmic distribution is evaluated as:

$$f(p, x) = \frac{-1}{ln(1-p)}(\frac{p^x}{x})\Big|_{x\geq 1} \quad \forall\ 0 < p < 1 \tag{4}$$

Fig. 3 shows the result of fitting Logarithmic and Poisson distributions on the IA at various UAV nodes. Three goodness of fit (GoF) metrics – Chi-squared GoF, Akaike Information Criteria (AIC), and Pearson correlation coefficient – are calculated to determine the most appropriate distribution for the traffic in our network. The IA data is divided into x bins to calculate the Chi-square GoF, and is represented as:

$$\chi^2 = \sum_{i=1}^{x} \frac{(O_i - E_i)^2}{E_i}, \tag{5}$$

where, O_i is the observed frequency in the bin, and E_i is the expected frequency of IA in the bin. Again, the likelihood $\mathcal{L}$, which denotes the probability of the data given a model, and F free parameters in the distribution, the AIC is calculated as,

$$\text{AIC} = -2(\log(\mathcal{L})) + 2\text{F} \tag{6}$$

Finally, for N number of IA samples with expected value x, observed value y, and mean of x and y denoted by $\bar{x}$ and $\bar{y}$, respectively, the Pearson correlation coefficient is represented as:

$$pearson = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{7}$$

From the above metrics, calculated on the inter-arrival times at every node, and the corresponding results tabulated in Table 1, it is inferred that the inter-arrival times at each node in the proposed multi-hop UAV network follows a Logarithmic distribution.

Definition 2. ***Effective UAV Bandwidth****: For m UAVs, each occupying a bandwidth of b, which is connected to a UAV with a total bandwidth of B, then $\sum_{i=1}^{m} b_i = mb \leq B$ and $m \leq B/b$. We term B as the Effective UAV Bandwidth, which is responsible for limiting the number of UAVs connecting to a single UAV.*

Definition 3. ***Swarm Node Depth****: It is the maximum depth (i.e., the maximum number of hops to be undertaken by an image frame) before processing. The depth of the node is limited by the Swarm Node Depth D_M such that $d \leq D_M$.*

Definition 4. ***Inter Arrival Time****: It is the time elapsed between the reception of two consecutive image frames by a UAV node. For time taken to transfer the i^{th} image frame $f(i)$ between UAVs denoted by $t_{f(i)}$, the Inter Arrival time is denoted as $IA = t_{f(i)-f(i-1)}, \quad \forall i > 1$. Additionally, with respect to equation 1, it can be stated that $IA \simeq \tau_i$.*

Definition 5. ***Service Time****: It is the time for which the i^{th} image frame $f(i)$ resides in a UAV node, and is denoted by ST. With respect to equation 1 it can be inferred that $ST\{f(i)\} \propto C_i$.*

Node	Chi-square		AIC		Pearson-coefficient	
	Poisson	Log	Poisson	Log	Poisson	Log
0	0.016	0.018	20.402	20.404	0.925	0.996
1	0.006	0.001	20.388	20.381	0.737	0.994
2	0.004	0.002	20.385	20.381	0.856	0.982
3	0.009	0.001	20.391	20.380	0.547	0.962
4	0.005	0.002	20.387	20.382	0.810	0.997

Table 1: Node wise Chi-squared GoF, Pearson correlation coefficient and AIC values for Poisson and Logrithmic distribution on IA of the network.

5. UAV Node Traffic Modeling

The IA, in our architecture, follows a Logarithmic distribution. We calculate the average waiting list of image frames at every UAV node using Queuing theory. We group the various UAV nodes in our architecture into three categories – 1) central node, 2) leaf node, and 3) intermediate node.

The central UAV node does not process any frames and is responsible only for video capture and frame generation, which is subsequently offloaded for processing to its immediate one-hop neighbors. Each swarm has a single central UAV node. In contrast, a UAV in the swarm with no further UAVs to offload their data to (i.e., no further children nodes present) is considered a leaf node. A leaf UAV node has to process whatever image frames get offloaded to it. Finally, any other UAV node in the network besides the central and leaf UAV nodes has two options – either process the frame by itself or assign the frame to one of its children. The data offload to other UAVs is decided from a Nash bargaining strategy-based pre-allocation of weights according to the swarm communication architecture.

In our work, we consider that the central UAV node is responsible for video capture in a swarm, whereas the other swarm members are responsible for sensing using only scalar sensors. This arrangement implies that the central UAV node is responsible for highly processing-intensive tasks, whereas the other member UAVs in the swarm have under-utilized processing resources. In our proposed multi-hop data offload scheme, considering that the tasks performed by all the UAV nodes, except the central UAV node, are not processing-intensive, the average service time ST for the processing of image frames for a fixed frame size is constant. A node's ST is only affected by the number of wireless connections to other UAV nodes maintained by it. An increase in the number of connections to a UAV node results in increased resource consumption at that node, which slows down the processing of that node leading to an increase in the time taken to process an image frame. Considering ϱ to be the percentage increase in ST for every wireless connection the UAV node is maintaining such that $\varrho \propto \gamma$, the expected ST of the node with m connections to it is formulated as:

$$E[ST] = ST_o(1+\varrho)^m \tag{8}$$

As each connection to a UAV node increases, it slows down the concerned UAV's processing by γ, increasing the original service time of that ST_o by $1+\varrho$. For m connections to a UAV, ST_o increases by $(1+\varrho)^m$. We denote the mean IA rate and the mean ST by β_a and β_s, respectively. β_a and β_s can be represented as $\beta_a = E[IA]^{-1}, \quad \beta_s = E[ST]^{-1}$ such that $E[.]$ represents the expectation of a random variable. The data offload mechanism in our proposed approach is similar to a $G/G/m$ queue such that the queue has m servers (UAV nodes) in which both service and the inter-arrival time have any given distribution. The IA, in our case, is distributed logarithmically (as established in Section 4), whereas the ST follows a polynomial distribution (from equation 8). For a single image frame f_i and a single processing UAV node, we formulate the utility of the UAV node as $U_s = \beta_a\beta_s^{-1}$. Along the same lines, for f_i with the choice of selecting any processing node from m UAV processing nodes, the utility of each UAV node is formulated as $U_s = \beta_a(m\beta_s)^{-1}$. Subsequently, the U_s is normalized to the maximum U_s of the system. For the sake of simplicity in calculations, we start the queue analysis of a $M/M/m$ queue, and eventually approximate it to a $G/G/m$ queue [25] when required. A $M/M/m$ queue is one in which there are m UAV nodes, and both the inter-arrival time and service time are exponentially distributed. The balance equations for a $M/M/m$ queue are formulated as:

$$\beta_a P(f_i - 1) = \begin{cases} f_i\beta_s P(f_i), & f_i \leqslant m \\ m\beta_s P(f_i), & \text{otherwise} \end{cases} \tag{9}$$

Using equation 9, the probability that there are f_i frames in a queue is calculated as:

$$P(f_i) = \begin{cases} P(0)\frac{(mU_s)^{f_i}}{f_i!}, & f_i \leqslant m \\ P(0)\frac{m^m {U_s}^{f_i}}{m!}, & \text{otherwise} \end{cases}, U_s \leqslant 1 \tag{10}$$

From equation 10 and the fact that $\sum_{f_i=0}^{\infty} P(f_i) = 1$, we calculate the probability of zero frames in a node's queue $P(0)$ as:

$$P(0) = \left[\sum_{f_i=0}^{m-1} \frac{(mU_s)_i^f}{f_i!} + \frac{(mU_s)^m}{m!\,(1-U_s)}\right]^{-1} \tag{11}$$

Subsequently, the average number of frames in the queue of a $M/M/m$ UAV node is calculated as:

$$N_Q = \sum_{f_i=0}^{\infty} f_i P(f_i + m) = P_Q(\frac{U_s}{1-U_s}),\ s.t.\ P_Q = \sum_{f_i=m}^{\infty} P(f_i) \tag{12}$$

From Little's theorem [26], the average waiting time W_M of a frame in a given UAV node for a $M/M/m$ queue is calculated as $W_M = \frac{N_Q}{\beta_a}$. The waiting time W_G of a frame for a $G/G/m$ queue at a UAV node can be approximated [27] as:

$$W_G \simeq W_M\Big(\frac{c_a^2 + c_s^2}{2}\Big) \tag{13}$$

where, c_a and c_s represent the coefficient of variation of IA and ST, respectively, and are calculated as $c_a = \sqrt{variance(IA)\beta_a^{-2}}$ and $c_s = \sqrt{variance(ST)\beta_s^{-2}}$. Similarly, the total time spent by a frame in a UAV node T_M for a $M/M/m$ queue is calculated as the sum of waiting time W_M and processing (servicing) time β_s^{-1}, and is represented as:

$$T_M = W_M + \frac{1}{\beta_s} = \frac{N_Q}{\beta_a} + \frac{1}{\beta_s} \tag{14}$$

whereas, for a $G/G/m$ queue, the total time spent by a frame in a UAV node T_G is formulated with respect to the relation in equation 13 as:

$$T_G = \Big(\Big(\frac{c_a^2 + c_s^2}{2}\Big)W_M\Big) + \frac{1}{\beta_s} \tag{15}$$

Further, applying Little's theorem, the average number of frames N at a UAV node is given by $N = \beta_a T$, which for a $M/M/m$ queue is calculated by incorporating equation 14 as:

$$N = \frac{\beta_a}{\beta_s} + N_Q \tag{16}$$

In case of our implementation, as we have previously established our system to be a $G/G/m$ one, equation 16 is rewritten by replacing N_Q with L_Q, which is the average number of image frames in the queue of a $G/G/m$ UAV node, and is approximated by Kingman [25] as:

$$L_Q = \frac{P_{Q0} U_s}{m!\,(1-U_s)^2} \frac{\beta_a}{\beta_s} \tag{17}$$

such that

$$P_{Q0} = \left(\sum_{k=0}^{m-1} \frac{(mU_s)^k}{k!} + \frac{(mU_s)^k}{k!\,(1-U_s)}\right)^{-1} \tag{18}$$

Equation 17 is used for calculating the queue at every UAV node in the UAV swarm network.

6. Strategizing a Nash Bargaining Game

Two cases which are encountered during processing offload from a UAV to its 1-hop neighbors are – 1) the offloading UAV node has more than one neighbor/child node and are mainly found in the intermediate levels of the offload architecture, and 2) the offloading UAV node has a single child node, which is a leaf node. Considering the case of an intermediate node, the queue at any node i is denoted by q_i. The 0^{th} node has the choice to either process the image frames itself or distribute it among its m children. The reduction of processor load at the 0^{th} node is made by distributing the processing of individual frames amongst the $m+1$ UAV nodes such that the node and its children share the frame-wise processing to mitigate the load on the 0^{th} node itself. We assign a penalty Q_i to a UAV node for offloading its processing to other UAV nodes. The penalty for assigning a frame to a child node is taken to be the frame transfer time t_{ld} between these nodes, whereas the penalty of processing the frame within the UAV node is t_{lc}, which is attributed to the increase in processing time of the node as a result of the connections maintained by the 0^{th} node. Another metric – strength of a node S_i – is considered for use in the penalty function Q_i such that for a UAV node i, its corresponding s_i denotes the number of child nodes under it such that $1 < s_i \leq m$. To embed these penalties Q_i is defined such that,

$$Q_i = \begin{cases} (q_i s_i)/t_{lc}, & i = 0 \\ (q_i s_i)/t_{ld}, & \text{otherwise} \end{cases} \tag{19}$$

The minimum probability with which a frame is assigned to a UAV node for processing is formulated as:

$$P^i{}_{min} = \frac{Q_i}{\sum_{j=0}^{m} Q_j} \tag{20}$$

Additionally, another parameter – rank R_i – is assigned to P^i_{min} for each UAV node. R_i for the i^{th} UAV node is formulated based on its depth d_i in the network with respect to the total depth of the network D_i, and is represented as $R_i = 1/(D_i - d_i)$ such that $D_i \geq 1$ and $d_i \geq (D_i - 1)$. Subsequently, the minimum probability of assigning a frame to the i^{th} UAV node for processing with respect to equation 20 and R_i is reformulated as:

$$P^i{}_{min} = \frac{Q_i R_i}{\sum_{j=0}^{m} Q_j} \quad \forall\, 0 \leqslant i \leqslant m \tag{21}$$

The utility of the i^{th} UAV node for processing offloading is formulated in terms of P^i_{min}, the probability of assigning an image frame to node i denoted by P_i, and child nodes under the i^{th} UAV node denoted by c_i is given by:

$$U_i(P_i) = \frac{P_i - P^i{}_{min}}{c_i + 1} \tag{22}$$

P_i for each UAV node, for a given UAV swarm architecture, is calculated prior to operation of the swarm using Nash bargaining (discussed later in this section), subject to the constraints $P_i \geq P^i{}_{min}$ and $\sum_{i=0}^{m} P_i = 1$. A set S denoting the joint utility function of all UAV nodes in the swarm is defined for this work such that

$$S = \{U_0(P_0), U_1(P_1), U_2(P_2), \cdots, U_m(P_m)\} \tag{23}$$

Equation 22 with respect to its constraints can be rewritten and represented for all the UAV nodes in the swarm as:

$$\begin{aligned} \sum_{i=0}^{m} P_i = \sum_{i=0}^{m} P_{min}^i + \sum_{i=0}^{m} U_i(P_i)(c_i+1) = 1 \\ \Rightarrow \sum_{i=0}^{m} U_i(P_i)(c_i+1) \leq 1 - \sum_{i=0}^{m} P_{min}^i \end{aligned} \tag{24}$$

From equations 23 and 24, the joint utility function S of the UAV swarm is generalized to

$$S = \left\{ U_i(P_i) \quad \middle| \sum_{i=0}^{m} U_i(P_i)(c_i+1) \leq 1 - \sum_{i=0}^{m} P_{min}^i \right\} \tag{25}$$

To establish the existance of the formulated utility function $U_i(P_i)$, the joint utility function S of the UAV nodes within the domain of the network proposed $i \in [0, m]$ has to be convex.

Theorem 1. *The joint utility function S of all the UAV nodes in the swarm is convex such that $f : U_i(P_i) \; | \sum_{i=0}^{m} U_i(P_i)(c_i+1) \leq 1 - \sum_{i=0}^{m} P_{min}^i$, $\forall 0 \leq i \leq m$.*

For a function $F : (P, P_{min}) \rightarrow \mathbb{R}^{+(m+1)} \; \forall \; 0 \leq m$ representing the solution for the weight allocation to the UAV nodes using the proposed Nash bargaining strategy, we consider the case of only one child UAV node connected to an offloading UAV. The optimization function is formulated as $F(P, P_{min}) = \arg\max_{P_1,P_2} U_1(P_1)U_2(P_2)$, which is rewritten as –

$$F(P, P_{min}) = \underset{P_1,P_2}{\arg\max} \frac{(P_1 - P^1{}_{min})(P_2 - P^2{}_{min})}{(c_1+1)(c_2+1)} \tag{26}$$

A Nash bargaining strategy can hold *iff* $F(P, P_{min})$ satisfies the criteria of Pareto efficiency, symmetry, invariance to linear transformation, and is independent of irrelevant alternatives. These four conditions validate the consideration of a utility function in a bargaining problem such that it provides a proportionally fair solution.

Lemma 1. *The proposed solution for the allocation of weights to the UAV nodes $F(P, P_{min})$ is Pareto-optimal, symmetric, invariant to linear transform, and independent of irrelevant alternatives.*

Theorem 2. *There exists a unique solution for the weight allocation among the UAV nodes, which satisfy the four Nash axioms, and this solution to the optimization problem is the pair $(P_1, P_2) \in P$ such that $(P_1, P_2) \geq (P^1{}_{min}, P^2{}_{min})$ that solves $F(P, P_{min}) = \arg\max_{P_1,P_2} U_1(P_1)U_2(P_2)$, which can also be rewritten as:*

$$F(P, P_{min}) = \underset{P_1,P_2}{\arg\max} \frac{(P_1 - P^1{}_{min})(P_2 - P^2{}_{min})}{(c_1+1)(c_2+1)} \tag{27}$$

Here, $(P_1 - P^1{}_{min})(P_2 - P^2{}_{min})$ is termed as the Nash product.

6.1. Solution to the Nash Bargaining Problem

The optimization function, which allocates weights to the various UAV nodes for a weighted distributed processing offloading within the m UAV nodes in the aerial swarm follows the four conditions or Nash axioms. A unique solution to the optimization function $F(P_i, P_{min}^i)$ is derived

Algorithm 1 Swarm Frame Distribution Algorithm

```
Inputs:(Camera_ID, Camera_fps)
Outputs:(Tracked_coordinates)
Initialize:
Add Camera_ID to Network
Network = Discover_nodes(Camera_ID,Network)
Queue = cal_queue(Network,Camera_fps)
Weights = cal_weights(Network,Queue)
flag,frame = capture(Camera_ID)
while flag do
    Target = get_Optimal_node(Network,Weights)
    Tracked_coordinates = Process(frame,Target)
end while
```

using the Lagrange Multiplier method. Now considering the weight allocation among the UAV nodes in the swarm, the optimization function subject to $\sum_{i=0}^{m} P_i = 1$, $P_i \geq {P^i}_{min}$ is $F(P, P_{min}) = \arg\max_P \prod_{i=0}^{m} U_i(P_i)$, and is simplified as:

$$F(P, P_{min}) = \arg\max_{P} \sum_{i=0}^{m} log\left(\frac{P_i - {P^i}_{min}}{c_i + 1}\right) \tag{28}$$

We solve equation 28 using Lagrange Multiplier λ, the function of which is formulated as:

$$L = \sum_{i=0}^{m} log\Big(\frac{P_i - {P^i}_{min}}{c_i + 1}\Big) - \lambda\Big(\sum_{i=0}^{m} P_i - 1\Big) \tag{29}$$

We arrive at the solution the optimization function in equation 28 considering $\frac{\partial L}{\partial P_i} = 0$ and $\frac{\partial L}{\partial \lambda} = 0$. This also ensures that the solution maximizes the optimization problem. A total of $(m + 1) + 1$ equations are obtained, the solutions to which can be generalized to obtain the weight assigned to i^{th} node as:

$$P_i = {P^i}_{min} + \frac{(1 - \sum_{i=0}^{m} {P^i}_{min})}{m + 2} \tag{30}$$

6.2. Weight Allocation to UAV Nodes in the Swarm

All the UAV nodes other than central and leaf UAV nodes have two probabilities – one with which its parent UAV node assigns it a frame, and the other with which it processes the frame by itself without passing it to its child node. The central UAV node does not process any image frames and acts as a client in a client-server communication analogy. Post-assignment of an image frame for processing, a leaf UAV node does not have the option of offloading their processing to other UAV nodes and act only as servers. The intermediate nodes act as both clients as well as servers. Algorithm 1 outlines the image frame distribution scheme for processing mitigation to member UAV nodes in a heterogenous UAV swarm. Algorithm 1 is responsible for the distribution of the generated image frames within the swarm members, depending on the network traffic and available processing. Initially, given the *ID* of the central UAV node with the attached camera sensor, and information of the camera's capture rate in frames per second (fps), a network is formed by the central UAV node by polling for UAVs in its vicinity and within its swarm using Algorithm 2.

Algorithm 2 on a UAV node first checks whether the node is a child node or not. IF at first pass, the node does not find any parent nodes, it becomes the parent node (root node). Further, if it is a child node, it establishes a connection with the parent node upon satisfying the bandwidth requirements for data offloading. Similarly, the node checks for the presence of child nodes under it, the detection of which results in running Algorithm 2 in these child nodes. This process keeps on repeating until there are no child nodes left to discover (all the current nodes are leaf nodes).

Algorithm 2 UAV Node Discovery Algorithm

Inputs:($Node$, $Network$)
Outputs:($Network$)
Initialize (Discover_nodes):
$child$ = check($Parent$)
for each $Node$ in $child$ **do**
    Establish connection between $Parent$ and $node$ in $Network$ if the Bandwidth constraint is satisfied
    $child\_child$ = check($Node$)
    for each $Node$ in $child\_child$ **do**
        $Network$ = Discover_nodes($Node$,$Network$)
    end for
end for

Once the network is formed, the average queue length at every UAV node is calculated using equation 16. The information of the estimated queue lengths at each UAV node enables the assignment of weights to each of these nodes. The image frames captured at the central UAV node are assigned to available UAV nodes for processing using Algorithm 3. This algorithm first checks whether the current node is the root node and whether it has children nodes ($child\_$). If the current onde has only one level of children nodes (which will be leaf nodes of the generated graph), it randomly selects any one of the children nodes for acting as servers during the distributed processing. Otherwise, the child node can act as a data generator (consumer) as well as a data processor (server). This is repeated until the leaf nodes are reached. Algorithm 3 thus decides its target nodes. The list of these selected nodes is returned to Algorithm 1. The selected nodes process the offloaded images using a pre-trained visual tracking algorithm and return the coordinates ($Tracked_{coordinates}$) of tracked humans to the central UAV node.

7. Performance Evaluation

This section is divided into two parts – 1) Evaluation hardware setup and 2) Simulation. The UAV network architectures used for comparison are recreated using four real-life UAVs with externally mounted Raspberry Pi processors to obtain network metrics from these implemented UAVs, as shown in Fig. 4. Large-scale simulation of the network is performed based on the real-life network metrics obtained and tuned into our custom-made simulator developed in Python. The network traffic and performance metrics from the real-life, small-scale UAV network is used for realistically guiding the behavior of the large-scale network of UAVs formed, which holds even for different network configurations using the same radio protocol (in our case, WiFi).

7.1. Evaluation Hardware Setup

A pilot-scale implementation of an aerial swarm using 4 UAVs is implemented, as shown in Fig. 4. Every member of the swarm is armed with unique sensors – scalar, as well as multimedia. For

Algorithm 3 Optimal Node Selection Algorithm

Inputs:$(Network, Weights)$
Outputs:$(Target)$
Initialize:
count $= 1$
$Node = Network \rightarrow root$
while True and ($Node$!= NULL) **do**
  $child\_ = Node \rightarrow child$
  if count $= 1$ **then**
    $Target$ = randomly select a $Node$ among the $child\_$ with the probabilities of them being servers.
  else
    $Target$ = randomly select a $Node$ among the $child\_$ and the $Node$ itself with the probabilities of them being servers and consumer respectively.
  end if
  if $Target == Node$ **then**
    return($Target$)
  else
    $Node = Target$
  end if
end while

(a) Circular formation (b) Linear formation

Figure 4: A pilot-scale UAV swarm implementation.

our work, we use a single camera-armed UAV. The other three UAVs in the network are armed with just scalar sensors. Initially, we use a standard video to test a Faster RCNN-based approach [28] for tracking ground targets (in our case, humans) in successive video frames. This model can be easily trained for use with UAV-based aerial videos of humans on the ground. However, as a part of this work addresses UAV-based visual tracking of human targets on the ground, the Faster RCNN module has been implemented on the low-power processors on the UAV. This implementation results in massive delays in computation and video frame-wise tracking. Additionally, the substantial power requirements of GPUs acts as a deterrent for its use on the small-scale UAVs, especially quadrotors. Typically in our case, a GPU takes 0.2 seconds, a CPU server takes 7 seconds, and the processor on the UAV (a Raspberry Pi module) takes 90 seconds to process a single video frame. A single UAV tasked with executing the tracking task on its own would severely deteriorate the efficiency of the said UAV's processing system and would be too slow to be of any use in real-time tracking of humans/targets on the ground. Further, ST_0 is calculated by allowing a single UAV with no connections to implement the Transfer-learning (Faster RCNN) based visual object detection on a single video frame. Similarly, τ is estimated according to the time taken to process the single frame by the UAV with a subsequently increasing number of connections to it. Finally, the transfer time of an image frame between UAVs is calculated by transmitting and receiving an image frame between two UAVs over a Wi-Fi link between the UAVs. The actual values of ST_0, τ, and T_f obtained from one of our UAVs in real-time are 90 seconds, 5%, and 0.005 seconds, respectively for a video frame size of $1KB$. Fig. 5 shows the results of the large-scale implementation of our proposed approach, and its comparison against the benchmark architectures for an incoming video frame rate of $25fps$ from the origin UAV.

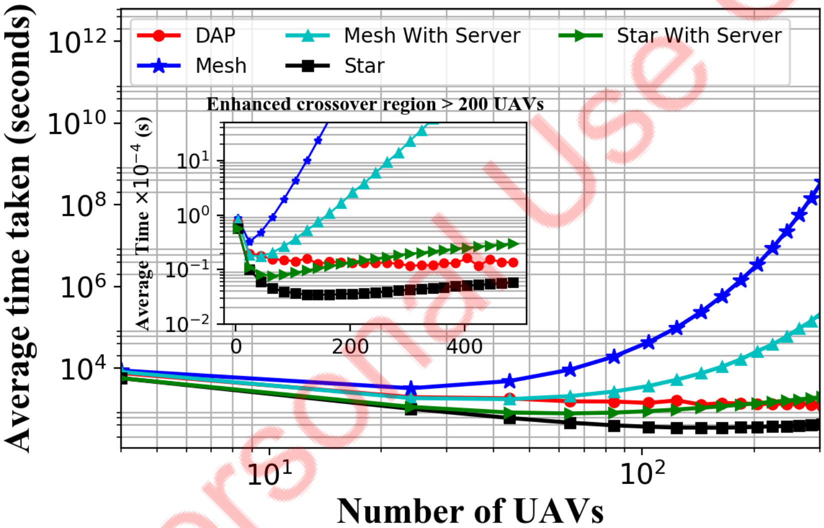
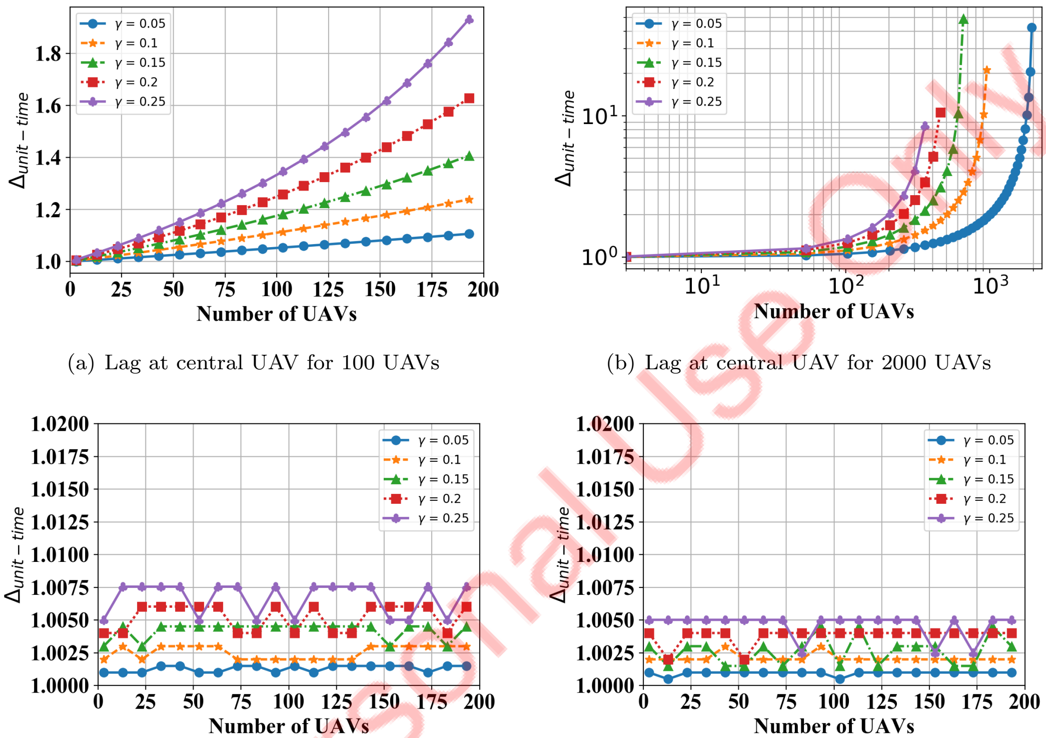


Figure 5: Comparison of the average processing time in the network taken for a frame-rate of $25fps$ among various architectures. Both x and y axes are on the log-scale. Only the y axis of the inset plot is on the log-scale.

7.2. Simulation

Simulation is performed to emulate UAV swarm networks of varying architectures, given the number of UAVs, processing time of single image frame on a UAV node with zero connections (ST_0), percentage increase in the processing time for every maintained UAV connection ($\tau \propto \gamma$), and wireless transfer time of data between two UAVs (T_f). Three broad classes of network connections are considered during our simulation – 1) the proposed multi-hop network architecture, 2) star connected network architecture, and 3) mesh connected network architecture. The simulation for the multi-hop architecture, which is the solution provided in our work, is based on Algorithm 1. This algorithm estimates the length of the queue at each UAV node from the inter-arrival times and then assigns appropriate weights to those UAV nodes, which helps in uniform processing resource utilization across the whole network, without unduly burdening a select few UAV nodes. In contrast, in the architectures based on one-hop communication, e.g., a connected star network, the image frames are equally distributed among the UAVs as all of them are equidistant from the central UAV and process similar resources. Finally, in the mesh connected network architecture, the current waiting list of image frames at each UAV node is considered before assigning that UAV node with an image frame to process.



(a) Lag at central UAV for 100 UAVs

(b) Lag at central UAV for 2000 UAVs

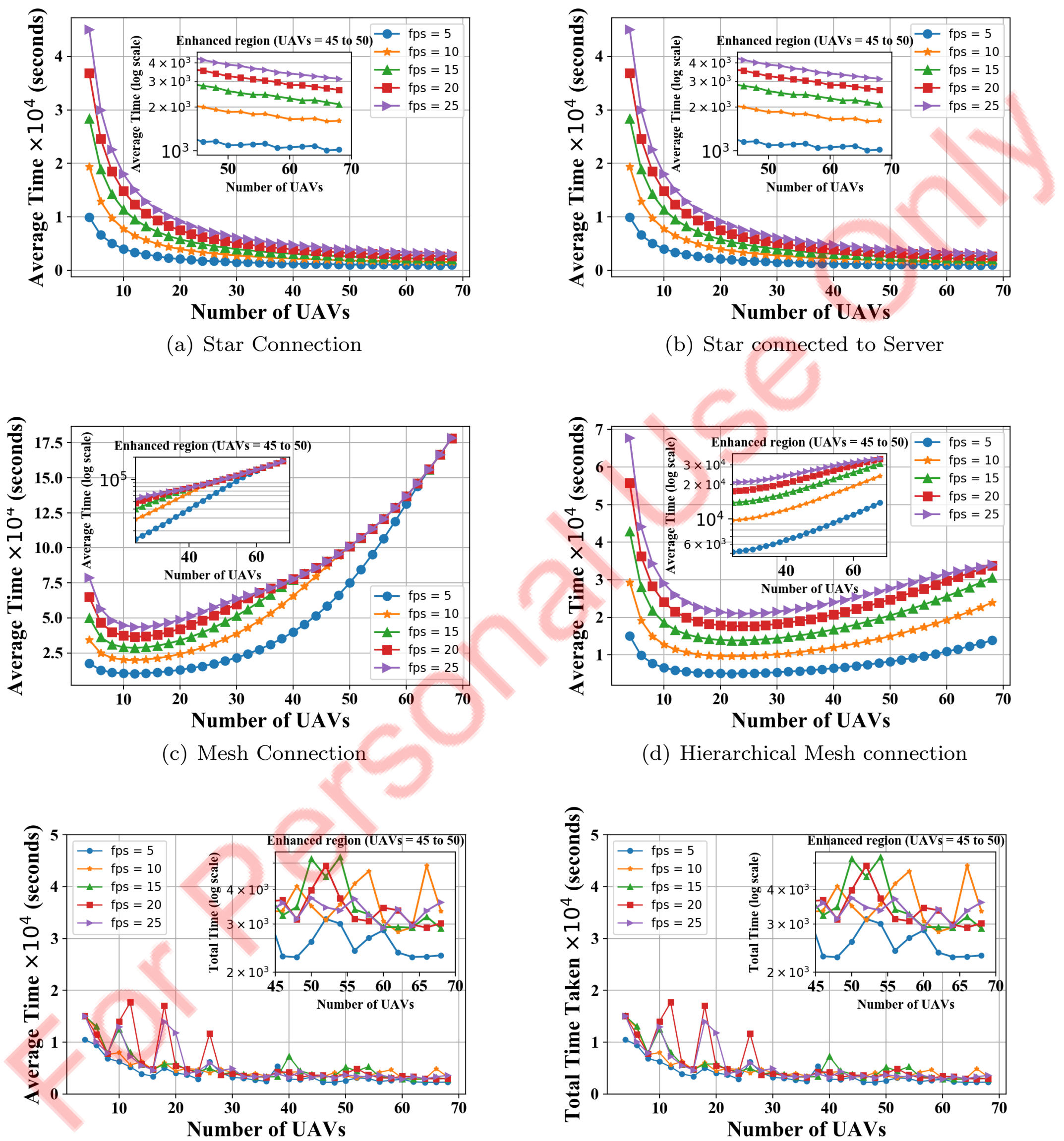
(c) Lag at Origin UAV

(d) Lag at Penultimate UAV

Figure 6: Calculated metrics (lag) for UAV network connection – one-hop star and DAP – architectures.

The performance of the proposed distributed aerial processing (DAP) is compared against the following regular UAV network architectures – 1) Star, 2) Star with a ground server, 3) Mesh, and 4) Hierarchical mesh. The UAVs in a star network communicate through a central UAV, which is connected through a one-hop link only. The number of UAVs that can simultaneously connect to the central UAV is limited due to γ of the central UAV, which results in limited scalability of the network. It is similar to the architecture explored in [29] [30]. In continuation, the UAVs in a star with a ground server network communicate through a central server on the ground, which is connected to the UAVs through a one-hop link only. The number of UAVs that can simultaneously connect to the central UAV is limited due to γ of the central UAV, which results in limited scalability of the network. It is similar to the architecture explored in [31].

The UAVs in a mesh network can all communicate with each other employing multiple hops via intermediate UAVs and is similar to the architecture in [32]. However, during processing offloading, the processing distribution on all UAVs is not symmetrical, resulting in UAVs with unequal load distribution in addition to the extra time taken to offload the data within the network nodes. Further, the hierarchical mesh network of UAVs is divided into two halves [33]. In each of the halves, all the UAVs are connected in a mesh. The communication between the meshes is through a ground server, which results in bottlenecks during processing and data offload.



(a) Star Connection

(b) Star connected to Server

(c) Mesh Connection

(d) Hierarchical Mesh connection

(e) Distributed Aerial Processing

(f) Distributed Aerial Processing and Decision Return

Figure 7: Comparison of the average processing time to increasing UAV swarm size and changes in the incoming video frame-rate fps for all benchmark architectures. The y axes of all the inset plots from (a) to (f) are on log-scale.

8. Results

This section is divided into four sub-sections to analyze the real-life, hardware metric tuned simulation of large-scale UAV network topologies – 1) inter-topology performance, 2) network scalability, 3) inter-topology processing time performance, and 4) collective network processing speed.

8.1. Inter Topology Performance

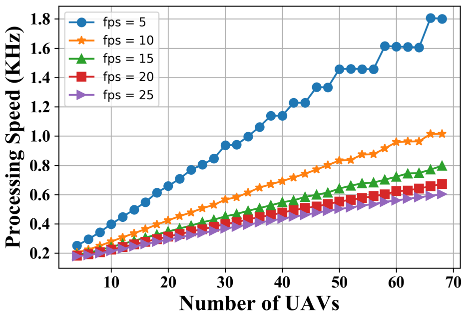
Fig.5 shows the average time taken to process video frames at $25fps$ for various architectures. It is seen that the overall processing time taken for the mesh and hierarchical mesh architectures increase with the increase in the number of UAV nodes in the network. The average time taken to process frames gradually reduces till a saturation point for each configuration is reached. For each of these saturation points, the corresponding network topology can no longer support collaborative processing and offload, which manifests itself in the form of an exponential increase in the average processing times. This is attributed mostly to the transfer time incurred during data offloading between the UAV nodes. The mesh configurations are the first ones to saturate as this topology itself involves data duplication between the network links to ensure network transfer reliability. In contrast, the star topologies ensure better data accommodation through the network links owing to the central controller. Further, the average processing time taken for the star architectures are comparatively lesser, which is attributed to the one-hop-only data offload restrictions. It is additionally seen that DAP initially behaves similar to a mesh network (performs better than mesh but poorer than star topologies), but gradually, for 200 UAV nodes, DAP surpasses the performance of start topology with a ground server (refer Fig. 5). As DAP maintains symmetrical distribution processing time among all the UAV nodes in the network, a more balanced and enhanced performance is projected for an increasing number of UAV nodes in the network.

8.2. Network Scalability

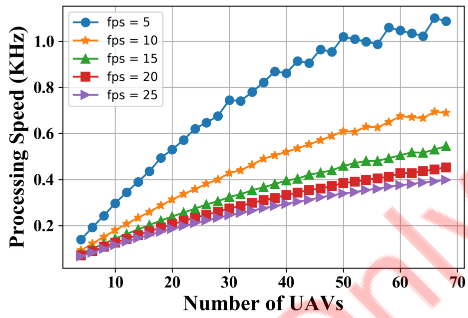
Fig. 6 shows the comparative performance of the star topology and our proposed DAP, regarding the scalability of the architecture. In Figs. 6(a) and 6(b), it is seen that for increasing γ, and increasing UAVs in the network, the data processing lag Δ increases and eventually saturates for larger number of UAVs (Fig. 6(b)). In contrast, the multihop topology followed by DAP results in constant lag for an increasing number of UAVs, as seen in Figs. 6(c) and 6(d). Unlike star topology, the processing load in DAP is evenly distributed across the network members. It is seen in Fig. 6(c) that Δ at the root or origin node is comparable to the one at the intermediate nodes (as shown in Fig. 6(d)). Summarizing the scalability, we see that star configuration has limited scalability and saturates beyond a point, which manifests itself in the form of an unrealistic increase in processing time (as shown in Fig. 6(b)). In contrast, the proposed DAP approach takes a balanced approach of uniform scalability and proportional distribution processing time among all the UAV nodes in the network.

8.3. Inter Topology Processing Times

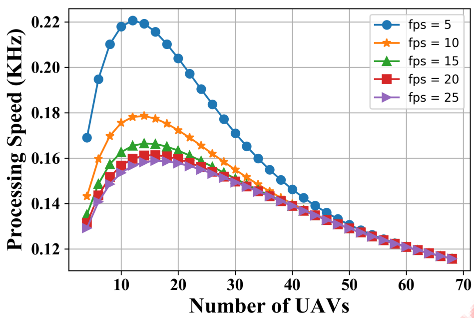
Fig. 7 shows the average time taken to distribute and process video frames for various architectures with varying video frame rates (in fps). The star (Fig. 7(a)) and the server connected star (Fig. 7(b)) networks show a drop in average processing time with an increase in the number of UAV nodes. Additionally, as the frame rate of the video being offloaded increases, the processing time goes up. In contrast, for the mesh (Fig. 7(c)) and hierarchical mesh (Fig. 7(d)) UAV networks, the average processing time increases with an increase in the number of UAV nodes. In mesh networks, it is seen that using the constraints outlined in the previous section, the average processing time for all frame rates converges, which is attributed to the processing overloading of the UAV nodes in that



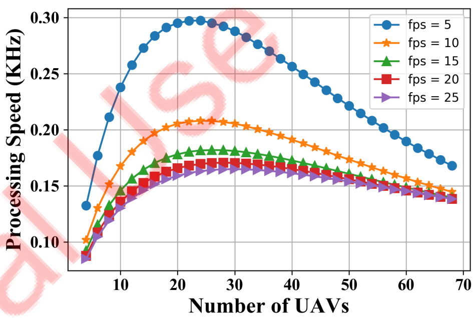
(a) Star Connection



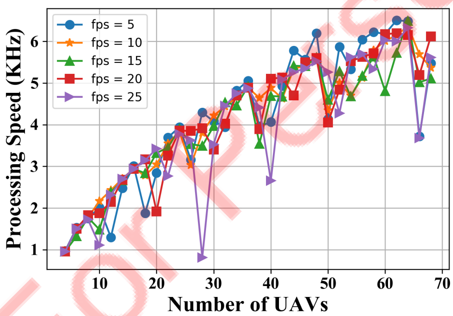
(b) Star connected to Server



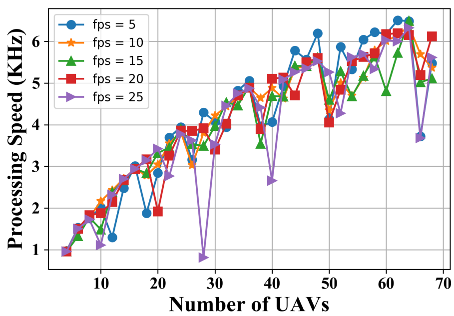
(c) Mesh Connection



(d) Hierarchical Mesh connection



(e) Distributed Aerial Processing



(f) Distributed Aerial Processing and Decision Return

Figure 8: Comparison of the average collective network processing speed available with respect to increasing UAV swarm size and changes in the incoming video frame-rate fps for all benchmark architectures. The y axes of all the inset plots from (a) to (f) are on log-scale.

network. For hierarchical mesh, this convergence occurs much later on. Finally, in the proposed DAP scheme in Figs. 7(e) and 7(f), the initial average processing time is much lower than that of the other configurations, and starts decreasing with an increasing number of UAV nodes. The sudden peaks obtained in the plots are attributed to the random arrangement of the UAV nodes in the architecture, wherein some nodes may not always have a child node to offload its processing. Fig. 7(f) depicts the total time taken to process the image frames and return the detected object's coordinates to the origin UAV. As the detected coordinates of the bounding box incur very low data load, this return operation takes negligible time.

8.4. Collective Network Processing Speed

Fig. 8 shows the available collective processing speed of the network in kHz. In Figs. 8(a) and 8(b), it is seen that as the network size goes up, the collective processing speed of the network for various values of γ increases. However, for the available real-life hardware metrics, it is observed that for approximately 200 UAVs in the star and its associated network, the collective network processing speed reaches 3 kHz, saturates, and eventually drops to 1 kHz. This sudden drop is attributed to the exhaustion of all processing resources at the offloading central UAV of the star topology. In contrast, for the mesh and hierarchical mesh topologies (as shown in Figs. 8(c) and 8(d)), reduction in the available processing speed of the topology starts at approximately 15 UAVs for regular mesh and 20 UAVs for the hierarchical mesh. The maximum collective network speed achieved is in the range of 0.3 kHz, which is much lesser than that of the star topology. The poor performance of mesh topology is attributed to the resources spent in establishing peer connections in the network, which leaves very little for the processing of image frames. Eventually, it is seen that DAP outperforms all the topologies regarding the collective network processing speed. In Figs. 8(e) and 8(f), we see that although some UAVs show a fall in their individual available processing speeds, the collective processing speed of the network increases with increase in the number of UAVs in the network. For the available hardware metrics, DAP achieves a collective network speed of approximately 6 kHz, which is double that of star topology for a fraction of UAVs.

9. Conclusion

This work proposes an intra-UAV swarm processing offloading scheme to mitigate the problem of increased processing delays due to processing-intensive tasks such as visual identification of farmlands, crop health monitoring, and crop growth tracking. Our proposed weighted offloading is governed by the use of a Nash bargaining game between the probability of a node processing the data itself or offloading it to a child node by a queueing theory-based analysis of the network traffic in the said swarm. Real-life hardware metrics calculated from our actual 4 UAV system are used for tuning simulations of a large number of UAVs following various network topologies. The results show that unlike star networks, our proposed DAP scheme is highly scalable, and for a larger number of UAVs, performs faster than star networks, as shown in Fig. 5. DAP always outperforms the mesh topology regarding average processing times. Interestingly, our approach outperforms both the star and mesh topologies regarding collective network processing speeds available such that even for a fraction of the UAVs in star and mesh topologies, DAP achieves double the collective speeds up of a star topology. The average processing times, although very high for our tuned hardware metrics due to restrictions of the hardware used (Raspberry Pi), establishes the immense usability and benefits of our approach in comparison to other topologies.

In the future, we plan to study our DAP approach by incorporating resource-constrained and low-footprint visual identification and tracking algorithms.

Acknowledgement

This work is supported by Information Technology Research Academy (ITRA), Government of India under ITRA-Water Grant ITRA/15(69)/WATER/M2M/01.

References

[1] Akyildiz IF, Kak A. The Internet of Space Things/CubeSats: A ubiquitous cyber-physical system for the connected world. Computer Networks 2019;150:134–49.

[2] Erdelj M, Król M, Natalizio E. Wireless sensor networks and multi-UAV systems for natural disaster management. Computer Networks 2017;124:72–86.

[3] Sun P, Boukerche A. Performance modeling and analysis of a UAV path planning and target detection in a UAV-based wireless sensor network. Computer Networks 2018;146:217–31.

[4] Shigekuni T, Ushio T, Azumi T. Cloud-assisted sensing and supervision of multiple unmanned aerial vehicles by a single operator. In: Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems. ICCPS '15; New York, NY, USA: ACM. ISBN 978-1-4503-3455-6; 2015, p. 254–.

[5] Gupta L, Jain R, Vaszkun G. Survey of important issues in uav communication networks. IEEE Communications Surveys & Tutorials 2016;18(2):1123–52.

[6] Sharafeddine S, Islambouli R. On-demand deployment of multiple aerial base stations for traffic offloading and network recovery. Computer Networks 2019;.

[7] Wachowiak MP, Timson MC, DuVal DJ. Adaptive Particle Swarm Optimization with Heterogeneous Multicore Parallelism and GPU Acceleration. IEEE Transactions on Parallel and Distributed Systems 2017;28(10):2784–93.

[8] Karimipour H, Dinavahi V. Parallel Domain-Decomposition-Based Distributed State Estimation for Large-Scale Power Systems. IEEE Transactions on Industry Applications 2016;52(2):1265–9.

[9] Koohifar F, Kumbhar A, Guvenc I. Receding Horizon Multi-UAV Cooperative Tracking of Moving RF Source. IEEE Communications Letters 2017;21(6):1433–6.

[10] Rao S, Ghose D. Sliding Mode Control-Based Autopilots for Leaderless Consensus of Unmanned Aerial Vehicles. IEEE Transactions on Control Systems Technology 2014;22(5):1964–72.

[11] Goddemeier N, Daniel K, Wietfeld C. Role-Based Connectivity Management with Realistic Air-to-Ground Channels for Cooperative UAVs. IEEE Journal on Selected Areas in Communications 2012;30(5):951–63.

[12] Sardellitti S, Barbarossa S. Joint optimization of collaborative sensing and radio resource allocation in small-cell networks. IEEE Transactions on Signal Processing 2013;61(18):4506–20.

[13] Xu Y, Cheng P, Chen Z, Li Y, Vucetic B. Mobile Collaborative Spectrum Sensing for Heterogeneous Networks: A Bayesian Machine Learning Approach. IEEE Transactions on Signal Processing 2018;66(21):5634–47.

[14] Wang R, Zhang L, Xiao K, Sun R, Cui L. EasiSee: Real-time vehicle classification and counting via low-cost collaborative sensing. IEEE Transactions on Intelligent Transportation Systems 2013;15(1):414–24.

[15] Rahman MA, Miah MS, Gueaieb W, El Saddik A. SENORA: A P2P service-oriented framework for collaborative multirobot sensor networks. IEEE Sensors Journal 2007;7(5):658–66.

[16] Gerasimou S, Matragkas N, Calinescu R. Towards systematic engineering of collaborative heterogeneous robotic systems. In: Proceedings of the 2^{nd} International Workshop on Robotics Software Engineering. IEEE Press; 2019, p. 25–8.

[17] Nigam N, Bieniawski S, Kroo I, Vian J. Control of Multiple UAVs for Persistent Surveillance: Algorithm and Flight Test Results. IEEE Transactions on Control Systems Technology 2012;20(5):1236–51.

[18] Pitre RR, Li XR, Delbalzo R. UAV Route Planning for Joint Search and Track Missions – An Information-Value Approach. IEEE Transactions on Aerospace and Electronic Systems 2012;48(3):2551–65.

[19] Daniel K, Rohde S, Goddemeier N, Wietfeld C. Cognitive Agent Mobility for Aerial Sensor Networks. IEEE Sensors Journal 2011;11(11):2671–82.

[20] Li K. A Game Theoretic Approach to Computation Offloading Strategy Optimization for Non-cooperative Users in Mobile Edge Computing. IEEE Transactions on Sustainable Computing 2018;.

[21] Gedik B, Liu L. MobiEyes: A Distributed Location Monitoring Service Using Moving Location Queries. IEEE Transactions on Mobile Computing 2006;5(10):1384–402.

[22] Wu D, Arkhipov DI, Kim M, Talcott CL, Regan AC, McCann JA, et al. Addsen: Adaptive data processing and dissemination for drone swarms in urban sensing. IEEE Transactions on Computers 2017;66(2):183–98.

[23] Fazio P, Tropea M, Rango FD, Voznak M. Pattern Prediction and Passive Bandwidth Management for Hand-over Optimization in QoS Cellular Networks with Vehicular Mobility. IEEE Transactions on Mobile Computing 2016;15(11):2809–24.

[24] Teymoori P, Sohraby K, Kim K. A Fair and Efficient Resource Allocation Scheme for Multi-Server Distributed Systems and Networks. IEEE Transactions on Mobile Computing 2016;15(9):2137–50.

[25] Kingman J, Atiyah M. The single server queue in heavy traffic. Operations management: critical perspectives on business and management 2003;57:40.

[26] Little JD. A proof for the queuing formula: L= λ w. Operations research 1961;9(3):383–7.

[27] Bertsimas DJ, Van Ryzin G. Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles. Operations Research 1993;41(1):60–76.

[28] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017;39(6):1137–49.

[29] Saad W, Han Z, Basar T, Debbah M, Hjorungnes A. A selfish approach to coalition formation among unmanned air vehicles in wireless networks. In: 2009 International Conference on Game Theory for Networks. 2009, p. 259–67.

[30] Erdelj M, Natalizio E, Chowdhury KR, Akyildiz IF. Help from the sky: Leveraging uavs for disaster management. IEEE Pervasive Computing 2017;16(1):24–32.

[31] Sivakumar A, Phang TS, Tan CKY, Seah WKG. Robust airborne wireless backbone using low-cost UAVs and commodity WiFi technology. In: 2008 8^{th} International Conference on ITS Telecommunications. 2008, p. 373–8.

[32] Sbeiti M, Goddemeier N, Behnke D, Wietfeld C. PASER: Secure and Efficient Routing Approach for Airborne Mesh Networks. IEEE Transactions on Wireless Communications 2016;15(3):1950–64.

[33] Bok PB, Tuchelmann Y. Context-aware qos control for wireless mesh networks of uavs. In: 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN). 2011, p. 1–6.