A presentation on – “Distributed Edge computing in IoT”

Based on the research paper titled “Distributed aerial processing for IoT-based edge UAV swarms in smart farming” by Anandarup Mukherjeea, Sudip Misraa, Anumandala Sukrutha and Narendra Singh Raghuwanshi.

By:

Mohammed Shahzad

444105788@student.ksu.edu.sa

Contents of this presentation

- Introduction –> Introduction to the research paper
- Problem statement and their solution approach
- DAP - System architecture, Modelling, Algorithms
- Results discussion
- Conclusion

Introduction

<u>Introduction to the research (DAP)</u>

- This work addresses the challenges of a decentralized and heterogeneous Unmanned Aerial Vehicle (UAV) swarm deployment
- They address the resulting problem of sensing and processing resource-intensive data aerially within the Edge swarm in the fastest and most efficient manner possible.
- To address under-utilization of the available computation resources in heterogenous swarms, they propose a Nash bargaining-based weighted intra-Edge processing offload scheme to mitigate the problem of heavy processing in some of the swarm members.

Introduction

Introduction to Agricultural IoT

- 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 and UAVs
- 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.

Fig: Agricultural IoT

Introduction

Introduction to Agricultural IoT

- Various solutions are proposed to address the problems of low computation capability of such UAVs.
- Solutions such as:
  - 1. cloud based data processing offloading from single UAVs ,
  - 2. processing offloading from a UAV to a ground server, etc.
- 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.

Problem statement and solution approach

Problem statement

- 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.
- 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

Problem statement and solution approach

Their approach

- 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, due to relatively lesser processing load on them.
- In this work, we distribute the captured video frames to other swarm members for processing.
- 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.

(b) A swarm of autonomous UAVs in flight

(c) Aerial imagery of agricultural plots

Problem statement and solution approach

Their approach

- 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.

Distributed Aerial Processing (DAP)

<u>The System Architecture</u>

- A one-hop UAV data-offload architecture consists of a central UAV to which m UAVs can connect.
- each UAV connected to a central node puts a certain amount of strain on its resources
- 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 Rc is denoted as Rc = γ1 + γ2 + . . . + γk−1 = k−1 i=1 γi.
- tk = Ra * t0 / (Ra − Rc) where Ra is available resources, Rc is consumed resources due to maintaining connections (fig 2)

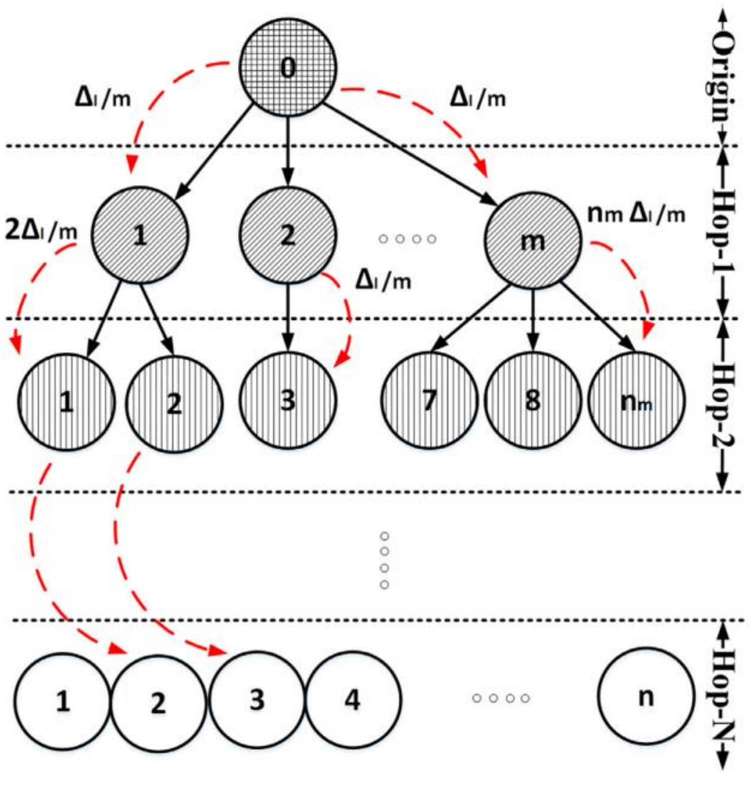


Fig. 2. A representation of the multi-hop offload connection.

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 $R_c = \gamma_1 + \gamma_2 + \ldots + \gamma_{k-1} = \sum_{i=1}^{k-1} \gamma_i$. Similarly, at $t_k|_{t>0}$, for $k-1$ UAVs connected to a central UAV node, we represent t_k as:

$$t_k = \frac{R_a}{R_a - R_c} t_0 \qquad (2)$$

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 $\gamma_1 = \gamma_2 = \ldots = \gamma_{m-1} = \sum_k \gamma$.

Distributed Aerial Processing (DAP)

<u>The System Architecture</u>

- If tUAV is the amount of time required to process l, then for a k UAV swarm, tUAV (k) (fig 2)
- Ci is a constant representing the internal processing time of the ith UAV, and τ i is the delay incurred during the transfer of one frame from one UAV to another in a single hop.
- Inter Arrival time is denoted as IA = tf(i)− f(i−1)

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.

tasks. If t_{UAV} is the amount of time required to process Δ_l, then for a k UAV swarm,

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

Distributed Aerial Processing (DAP)

UAV node traffic modeling

- Service Time: It is the time for which the ith image frame f(i) resides in a UAV node, and is denoted by ST
- We denote the mean IA rate and the mean ST by ßa and ßs, respectively. ßa and ßs
- Where Nq is average Number of frames in processing queue
- The probability that there are fi frames in a queue is calculated as P(fi).
- For a single image frame fi and a single processing UAV node, we formulate the utility of the UAV node as Us = ßaßs−1

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

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

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:

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

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:

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

Distributed Aerial Processing (DAP)

UAV node traffic modelling

- Traffic modelling

$c_s = \sqrt{variance(ST)\beta_s^{-2}}$. 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 Eq. (13) as:

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

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 Eq. (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, Eq. (16) is rewritten by replacing N_Q with L_Q, which is the average number of image frames in

Distributed Aerial Processing (DAP)

UAV node traffic modeling

- Traffic modelling

- Eq. (16) is rewritten by replacing Nq with Lq, which is the average number of image frames in the queue of a G/G/m UAV node.

the queue of a $G/G/m$ UAV node, and is approximated by Kingman [25] as:

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

such that

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

Eq. (17) is used for calculating the queue at every UAV node in the UAV swarm network.

Distributed Aerial Processing (DAP)

Strategizing a Nash bargaining game

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

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

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

$$\sum_{i=0}^{m} P_i = \sum_{i=0}^{m} P^i_{\text{min}} + \sum_{i=0}^{m} U_i(P_i)(c_i + 1) = 1$$
$$\Rightarrow \sum_{i=0}^{m} U_i(P_i)(c_i + 1) \leq 1 - \sum_{i=0}^{m} P^i_{\text{min}} \tag{24}$$

From Eqs. (23) and (24), the joint utility function S of the UAV swarm is generalized to

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

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.

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:

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

Additionally, another parameter – rank R_i – is assigned to P^i_{min} for each UAV node. R_i for the ith 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 ith UAV node for processing with respect to Eq. (20) and R_i is reformulated as:

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

The utility of the ith 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 ith UAV node denoted by c_i is given by:

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

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 $P_i \geq P^i{}_{\text{min}}$

Distributed Aerial Processing (DAP)

Strategizing a Nash bargaining game

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 $F(P_i, P^i_{\min})$ is derived 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 Eq. (28) using Lagrange Multiplier λ, the function of which is formulated as:

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

We arrive at the solution the optimization function in Eq. (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 ith node as:

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

Distributed Aerial Processing (DAP)

Algorithms for processing and node selection

- 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 is responsible for the distribution of the generated image frames within the swarm members, depending on the network traffic and available processing

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

Distributed Aerial Processing (DAP)

Algorithm for node discovery

- 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.
- Once the network is formed, the average queue length at every UAV node is calculated using Eq. (16).

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
```

Distributed Aerial Processing (DAP)

Algorithm for optimal node selection

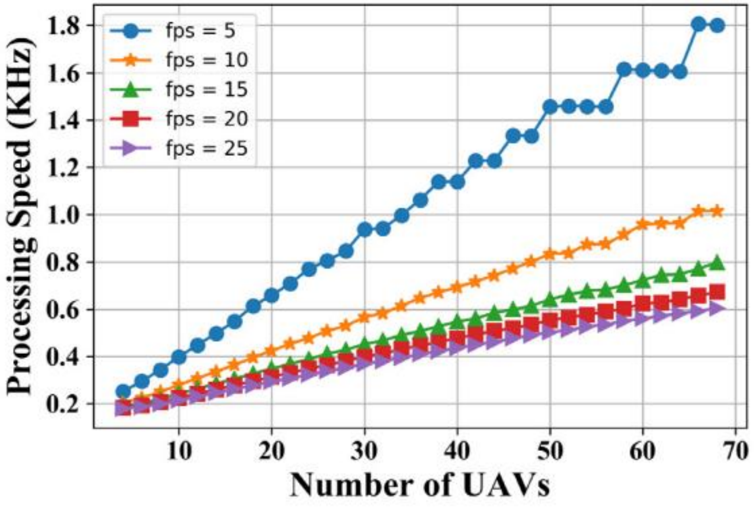
- The image frames captured at the central UAV node are assigned to available UAV nodes for processing using Algorithm 3.
- This algorithm first checks whether the current node is the root node and whether it has children nodes (child_).
- If the current node has only one level of children nodes, it 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.

Algorithm 3 Optimal node selection algorithm.

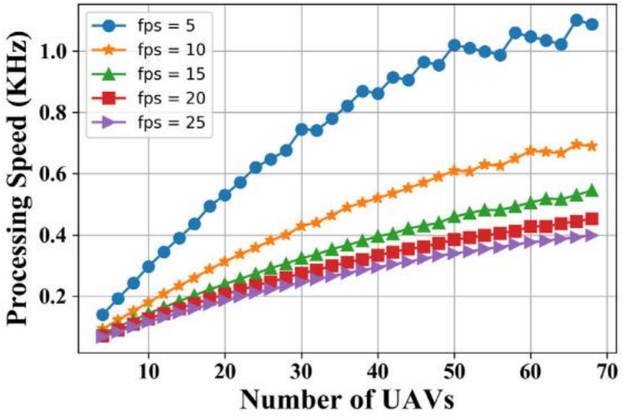
```
Inputs:(Network, Weights)
Outputs:(Target)
Initialize:
count = 1
Node = Network → root
while True and (Node != NULL) do
    child_ = Node → child
    if count = 1 then
        Target = randomly select a Node among the child_ with the probabilities of them being servers.
    else
        Target = 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
```

Results obtained using DAP

- Results analysis
- Collective network processing speed-
- Fig. 8 shows the available collective processing speed of the network in kHz. In Fig. 8(a) and (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.



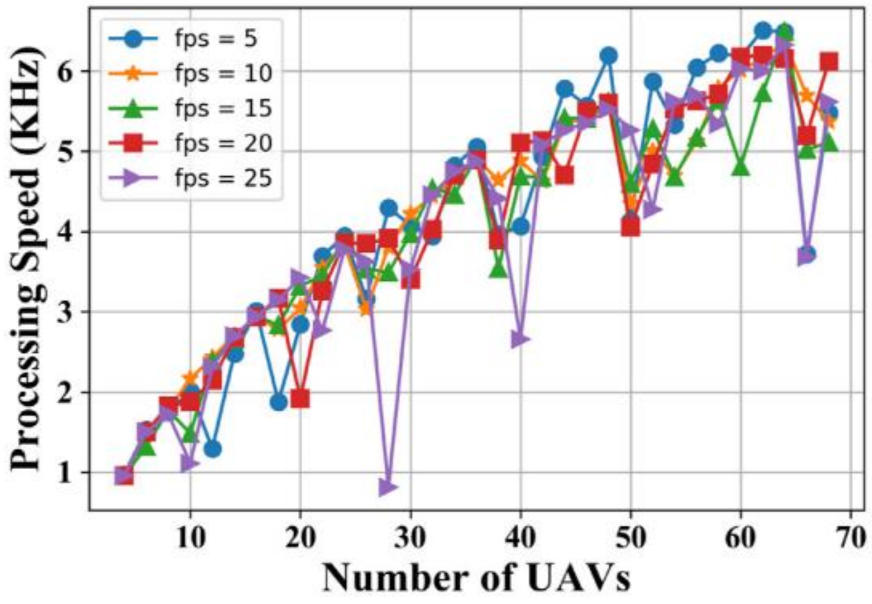
(a) Star Connection



(b) Star connected to Server

Results obtained using DAP

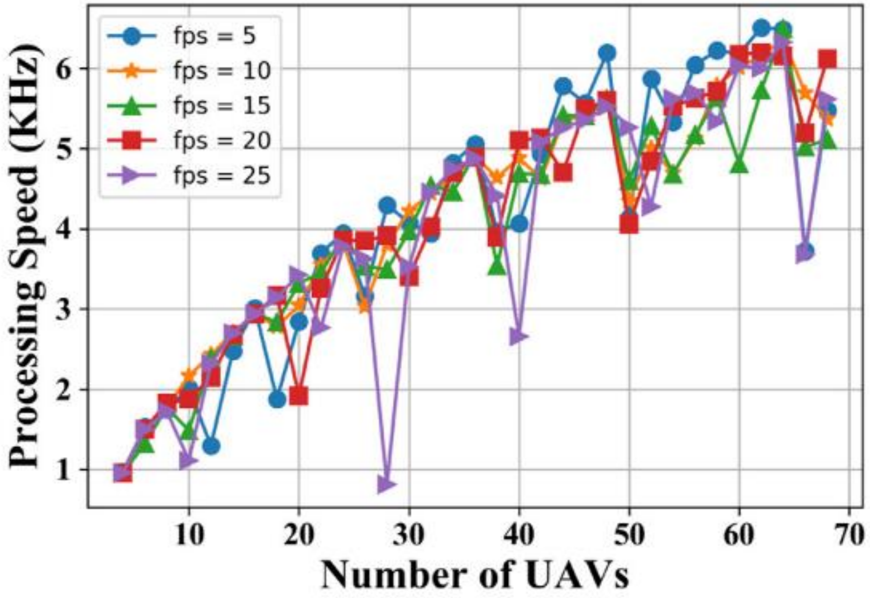
- <u>Results analysis</u>
- The poor performance of mesh topology is attributed to the resources spent in establishing peer connections in the network, which leaves very little for the processing of image frames.
- Eventually, it is seen that DAP outperforms all the topologies regarding the collective network processing speed.



(e) Distributed Aerial Processing

Results obtained using DAP

- <u>Results analysis</u>
- In Fig. 8(e) and (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.



(f) Distributed Aerial Processing and Decision Return

Conclusion

- 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 intraswarm 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.

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
- 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.

Thank You!