UAV Virtualization for Enabling Heterogeneous and Persistent UAV-as-a-Service

Nidhi Pathak, *Student Member, IEEE*, Sudip Misra, *Senior Member, IEEE*, Anandarup Mukherjee, *Student Member, IEEE*, Arijit Roy, *Student Member, IEEE*, and Albert Zomaya, *Fellow, IEEE*

***Abstract*—In this paper, we propose an architecture for UAV virtualization with the primary aim of providing virtualized UAV services to multiple users by envisioning the concept of *UAV-as-a-Service*. In contrast to traditional UAVs, which are resource-constraint in nature and exhibit shorter flight times, our proposed UAV virtualization overcomes the limitations of short flight time of traditional UAVs, in turn allowing them to provide persistent and ubiquitous services. We achieve the virtualization of a UAV through multiple collaborating real-life UAVs performing various tasks in tandem. In this work, we focus on the placement and selection of UAVs to achieve virtualization. We use a social welfare-based approach to select suitable UAVs, from the available ones, and map the UAV to a virtual one. This work enables the provision of different UAV services to multiple end-users, without actual procurement of the UAVs by the end-users. We compare the results for optimal placement, normal maximum energy-based UAV selection, and Atkinson-based selection method. We also compare the virtual model and simple UAV-to-task model to physical UAV usage, task completion ratio, and residual energy of the system. Our proposed model outperforms the traditional model with a task completion efficiency of 94.26%. The residual energy of the system also increases with an increase in the number of tasks.**

***Index Terms*—Unmanned Aerial Vehicle, persistent service, virtualization, scheduling, task allocation, social welfare.**

I. Introduction

THE In this paper, we propose an architecture to provide persistent and ubiquitous UAV services to the end-users, unlike the traditional UAV services, which are intermittent and short-lived. To enable the proposed architecture, we introduce the concept of UAV virtualization. Fig. 1 depicts the overall system architecture of the proposed UAV virtualization, along with the involved actors. The architecture not only facilitates the services but also provides monetary benefits to the actors involved in this system. Virtualization allows multiple similar UAVs, that may or may not belong to the same owners, to take up a requested task. Additionally, Virtualization makes it possible to perform long-duration tasks without any physical involvement of the end-user. The proposed architecture consists of three actors– 1) the UAV owners, 2) the service provider, and 3) the end-users. We consider the UAVs are heterogeneous in terms of the number of connected sensors, types of sensors, and battery capacity. The proposed architecture uses the cloud as the backend infrastructure for its implementation. The available UAVs in the range of the tasks

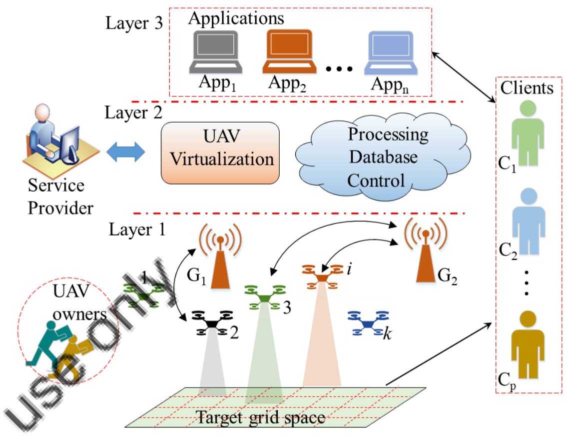


Fig. 1: The proposed system architecture for UAV virtualization.

form a group called the local UAV society (soc^{i}_{local}), specific to that task. These groups are used to select physical UAVs and map it to the virtual UAV. A social welfare-based selection scheme is used to maximize the overall residual energy (E_{res}) of the society. For simplicity, we consider only the energy consumption for UAV traversal and task performance by the sensors.

A. Motivation

UAVs are being used to serve a plethora of applications. However, it is not feasible and economical for an end-user to procure different application-specific UAVs. The operation of multiple UAVs typically requires a team of skilled personnel to deploy, collect, and recharge the UAVs. Additionally, the requirements of prior permission from the government and regulatory bodies before initiating any UAV operations also complicates UAV ownership. The cost of procuring, maintaining, and operating the UAVs are highly prohibitive and cumbersome to manage for a majority of the end-users. In a traditional UAV service, a UAV owner can serve only a single end-user at a time. The flight time served by the UAV is not sufficient enough to perform long-duration tasks in a single stretch. The existing shortcomings motivate us to propose the architecture of UAV virtualization to provide seamless UAV services to the end-users. In the proposed architecture, a UAV can serve multiple end-users based on their preferences.

N. Pathak and A. Roy are with the Advanced Technology Development Center, Indian Institute of Technology Kharagpur, India

S. Misra and A. Mukherjee are with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, West Bengal, India

A. Zomaya is with the School of Information Technologies, The University of Sydney, Australia

The architecture enables an end-user to receive the necessary UAV services without procuring any physical UAVs. The UAV owners and service provider receive monetary benefits for their services to the end-users.

B. Contribution

The proposed architecture provides UAV-as-a-service to end-users through the implementation of UAV virtualization. The specific contributions of our work are as follows:

- A generalized UAV virtualization architecture to enable persistent UAV services for long-duration missions and decrease the redundancy in task performance.
- We design an appropriate task-specific UAV selection scheme using social-choice theory, analyze the UAV occupancy and coverage analysis for both homogeneous and heterogeneous UAV types in a defined region.
- We evaluate the performance of the proposed architecture and the selection schemes through rigorous emulation.

II. Related Work

This section highlights the recent developments and research concerning our proposed problem. We identify these works separately and link them at the end of this section to provide a clear insight into the evolution of the presented idea.

A. Sensor Cloud and Virtualization

Sensor cloud has been one of the most demanding areas of research and application since its inception. *Yuriyama* and *Kushida* [1] proposed the idea of making the sensors available and ubiquitous through the sensor cloud infrastructure. Bose *et al*. Misra *et al*. [2] proposed theoretical modeling of the sensor cloud infrastructure with detailed comparative feature analysis.

B. UAV Cloud

Mahmoud *et al*. [3] proposed integrating UAVs with cloud and incorporated its basic advantages of scalability, high computation resource, high storage, and ubiquity. Luo *et al*. [4] proposed offloading data from the UAVs to the cloud, which releases the onboard memory space for data acquisition. Similarly, other approaches include NFV-based UAV-cloud integration [5] and UAV cloudlets [6]. Tang *et al*. [7] proposed an architecture for using UAVs mounted cloudlets for location-based social networks (LBSN) services with highly mobile users. The architecture utilized the UAV-based cloudlets to implement an adaptive recommendation model for LSBN.

C. Persistent UAV services

Increasing the capacity of the UAV flight time to produce persistent coverage is a challenging aspect of UAV operation. Lee *et al*. [8] developed a robotic operating system (ROS)-based tracking system, where multiple UAVs can be connected to a central computer and a system for task allocation and handoff among UAVs during a mission, autonomously. Similarly, Park *et al*. [9] implemented a prototype for providing continuous security presence of a UAV, using multiple UAVs in sequence, for a customer in an outdoor scenario.

D. UAV Selection and Task Allocation

Task allocation in a multi-agent system is a well-studied area and is under continuous exploration for various cyber-physical systems. Various algorithms are used for UAV selection [10] and task allocation, focusing on the different network and flight parameters [11]. Kim *et al*. [12] [13] proposed a social choice theory-based selection process focused on the overall resource consumption of a group of robots or UAVs. Galkin *et al*. [14] proposed UAV positioning strategies for wireless access points. The positions are based on user hotspots, interference due to coexisting UAVs in the network, and other parameters in the urban location scenario.

***Synthesis*:** The works discussed so far are independently simulated and implemented, with most of them being application-specific. Also, the implementation of the works done so far is limited by various factors like ground control station, LoS operation, and autonomous control. A more generic architecture is required with the capability to deal with the issues in a single platform. Towards this aim, we propose a novel scheme in this area targeting the gaps discussed earlier.

III. System Architecture

Fig. 1 shows the proposed architecture, subdivided into three layers. These layers encompass different components and actors. We discuss the three layers in details below.

Layer 1: Fig. 2 shows the basic sub-system and the interconnections in a UAV. A traditional UAV consists of a processor-controller board, rotors, GPS module, and a power source. An additional secondary processor is attached to the traditional UAV to connect the sensors, actuators, and add decision making capability to the UAV. A communication module is connected to the secondary processor to establish the connection between UAV and the cloud server through the Internet. A physical UAV is equipped with a necessary

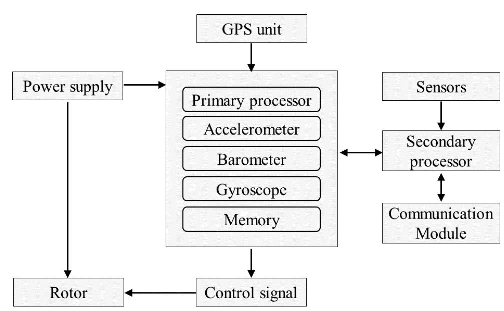


Fig. 2: UAV sub-systems and their interconnections.

communication module that can use both WiFi and cellular networks. The module efficiently switches its mode of connectivity based on the availability and strength of the network throughout its operation. Due to highly dynamic nature of the architecture, the data transfer may suffer considerable delay due to the delay in processing, queue, transmission and propagation, given as $\delta_{total} = \delta_{proc} + \delta_{que} + \delta_{trans} + \delta_{prop}$. The continuous connectivity and communication with the cloud platform also increase energy consumption.

Assumption 1. *A UAV is assumed to be always within the coverage range of a wide-area wireless network (such as WiFi or cellular), which enables the UAV to be connected to the cloud.*

Layer 2: The paper proposes a cloud-based architecture for providing UAV-as-a-service by incorporating the concept of virtualization. The end-users and the UAV owners register to the platform to avail of the platform services. Layer 2 comprises of the cloud infrastructure which handles the database, hosts the platform server, manages the virtual UAV provisioning, and other operations during any UAV flight. This layer mainly handles the dynamic processes of the architecture and maintains the abstraction.

Layer 3: Layer 3 comprises of the actors involved in the proposed architecture – UAV owners, end-users, and service providers. The end-user registers to the web application, hosted in layer 2, with personal and task details. The UAV owner registers with detailed information about the UAV. Each end-user application is assigned a virtual UAV to complete the requested task. A service provider provides the necessary infrastructure to implement the platform and enable UAV services. A service provider monitors the administration and coordination among various components of the proposed architecture, such as database management, UAV maintenance, and virtual UAV provisioning. An end-user or a UAV owner may anytime withdraw their requests and services from the platform.

Technologies such as SDN and NFV can be used to further implement the proposed architecture [15] [16]. We further discuss two different aspects of the virtualization problem. First, we discuss the optimal placements of UAVs in our defined region of interest. Second, the virtualization architecture for a UAV system is defined, supported by a social-welfare-based UAV selection scheme.

IV. Physical UAV Placement

The target region for UAV services is considered to be a $n \times n$ grid space. The placement and operation of the UAVs in a defined grid space are ruled by the conditions and constraints of the architecture. A grid can be populated by only one UAV.

Definition 1. *A n-hop is defined as the distance between a grid and its n^{th} adjacent grid.*

The movement of a UAV is restricted to $1-hop$, where $1-hop$ is the distance between two adjacent grids. Each of the UAVs has a threshold value of energy-level, E_{thr}, below which the UAVs are unable to serve any application.

Let a UAV travel the distance, L_{max}, with a constant velocity v. Also, the maximum energy capacity of the UAV is E_{cap}. To determine the energy consumption during UAV's traversal from one point to another in a mission, we define a metric as follows:

Definition 2. *The per unit energy consumption for the distance traveled by a UAV with its full energy capacity is defined as a metric called energy index, λ_{energy}, of the UAV such that $\lambda_{energy} = E_{cap} L_{max}^{-1}$.*

The total time taken by a UAV during a mission is divided into two parts- t_{travel} be the time required to travel to the target location and t_{hover} be the hovering time utilized for performing the assigned task. The total time required and energy consumed for traversal during a mission by a UAV is given by $T_{total} = t_{hover} + t_{travel}$ and $E_{travel} = v \times T_{total} \times \lambda_{energy}$, respectively. Let, energy consumed for performing a task be E_{task} and the total, E_{tot}, consumed by a UAV during a mission is given by $E_{tot} = E_{travel} + E_{task}$.

Our objective is to minimize the number of UAVs being used for completion of the overall tasks, reduce the overall energy consumption for the tasks generated, and maximize the number of tasks completed subject to certain constraints.

$$min \sum_{i=1}^{Tasks} u_i, max \sum_{i}^{grid} p_{comp}, min \sum E_{travel} + E_{task} \quad (1)$$

subject to

$$1 - hop \leqslant \sqrt{2} L \quad (2)$$

$$x_j.u_{ij} = \begin{cases} u_i, & E_{res}(u_i) > E_{thr} \\ 0, & \text{Otherwise,} \end{cases} \quad (3)$$

$$\sum_{i=1}^{grid} u_i \leqslant 1 \quad (4)$$

UAVs in a group with the same service providing capabilities are termed as homogeneous UAVs, while the UAVs with different service providing capabilities are referred to as heterogeneous UAVs. The following subsections discuss the UAV placements of the two categories in a $n \times n$ grid-space where each UAV can travel a distance of $p - hops$.

Definition 3. *A UAV coverage unit is the number of grids that is within its allowed hop distance and can be served for any task.*

A. Homogeneous UAV

First, we define the area covering capacity for a single UAV. Any grid-space can be divided into multiple UAV coverage units.

Theorem 1. *For a p-hop coverage, number of UAVs, U to cover a $n \times n$ grid can be given as $U = \left(\left\lceil \frac{n}{2p+1} \right\rceil \right)^2$*

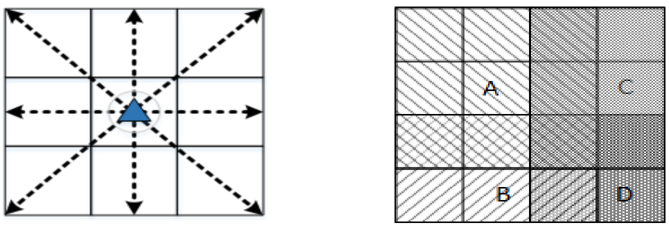
Proof. For a p-hop coverage, the UAV coverage unit consists of $(2p+1)^2$ grids. An $n \times n$ grid space can be seen as a combination of multiple UAV coverage units represented by a linear combination as:

$$n = (2p+1)k + m, \quad k \in \mathbb{R}_+ \quad and \quad m \in \mod (2p+1) \quad (5)$$

To cover any $n \times n$ grid space, it is represented as a multiple of the UAV coverage unit. The UAV coverage unit-based grid space represents the minimum number of UAVs required to cover the grid space. We use the ceil value of $2p+1$ to include the case where n is not a multiple of $2p+1$. Let $p = 1$, then $U = \left(\left\lceil \frac{n}{3} \right\rceil \right)^2$. Fig. 3a shows the UAV coverage unit $p = 1$. For $n = 4$ the minimum number of UAVs is 4 as shown in Fig. 3b. $\square$

TABLE I: Comparison of the proposed architecture with similar technologies.

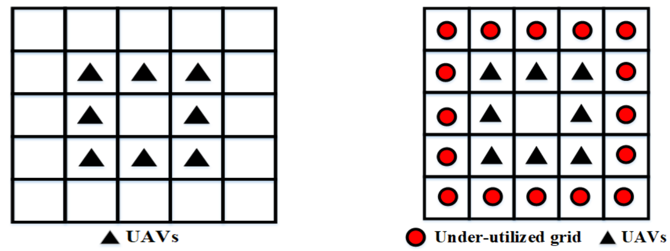
Parameters	WSN	MWSN	Virtual Sensor Cloud	Traditional UAV	Virtual UAV Service
Field of Operation	Ground, 2D	Ground, 2D	Ground, 2D	Aerial, 3D	Aerial, 3D
Operation Range	Local	Short Range	Global	LoS, Short Range	Global
Energy Source	Cell/Li-based Battery	Automotive Battery	Rechargeable cell/Battery	Lipo Battery	Lipo Battery
Lifetime	Limited	Limited	Limited	Limited	Unlimited
Metadata modelling	SML	SML	SML	MAVLink	MAVLink
Mobility	Static	Mobile	Static	Highly mobile	Highly mobile
Deployment Range	Small	Limited	Relatively larger	Limited	Relatively larger
Operating Frequency	2.4/5/	-	-	2.4GHz/5.8GHz	2.4GHz/5.8GHz
Ad-hoc network	✓	-	-	-	-
Hybrid network support	×	✓	✓	✓	✓



(a) UAV coverage unit for 1-*hop*.

(b) 4×4 grid space coverage using minimum number of UAVs.

Fig. 3: Grid coverage using UAV coverage unit.



(a) Possible placement of UAVs to ensure no wastage of grid coverage.

(b) Under-utilized UAV coverage for one-hop UAV.

Fig. 4: Placement of UAVs in a grid-space with full utilization and partial utilization by UAV coverage.

B. Heterogeneous UAV

In the case of heterogeneous UAVs, there is a trade-off between the number of UAVs and types of UAVs that can be placed in a grid space. However, the idea is to provide each type of service to all the grids in the grid space.

Theorem 2. *For a p-hop UAV, maximum number of heterogeneous UAVs that can be placed in a $n\times n$ grid space with no wastage of coverage area is* $\left(n-(2p+1)+1\right)^2$.

Proof. A UAV at the boundary of a grid space or a distance less than its hop distance from the boundary wastes some of its coverage grid. We find all the grids that are at a minimum distance of p-hop from the boundary of the grid space. This quantity can be expressed in terms of the hop distance p and grid size n as $U_{max}=(n-(2p+1)+1)^2$. The shaded grids in Fig. 4a represents the possible UAV placements for different values of n with p=1. Emphasizing on the type of UAVs, each of these grids can be occupied by a different type of UAV. □

Lemma 1. *Number of under-utilized grids if all the grids are covered is given by* $W=(n+2p)^2-n^2$.

Proof. Let all the grids at the edges in Fig. 4b be occupied by UAVs. The UAVs at the edges of the grid space cover only the area inside the grid space while the other half is under-utilized, i.e., the grids are under the coverage of a UAV but are not being served. Fig. 4b represents the total under-utilized grids if the UAVs are placed at the specified locations in a 3×3 grid-space. □

Lemma 2. *Maximum number of type of UAVs in a $n\times n$ grid space, if no same type of UAV are allowed to overlap their coverage is given as* $(2p+1)^2$, $\forall\ n\geq(2p+1)$.

Proof. Considering the smallest unit, i.e., the UAV coverage unit, the maximum number of heterogeneous UAVs that are placed without any similar overlapping grids is the total number of grids in the coverage unit. For a 3×3 grid space, the total number of different types of UAV is nine since there are nine grids in the unit. As discussed earlier, any grid space can be represented as a combination of multiple UAV coverage units. Therefore, the maximum number of heterogeneous UAVs for any grid space is always equal to the number of grids in its UAV coverage unit. □

V. VIRTUALIZATION

A UAV owner is denoted as $o_i\in O$, where O is a set of all the UAV owners. UAV owners lease their respective UAVs to the service providing platform. These UAVs are used in the composition of the virtual UAV for the end-users. Each UAV, registered with the UAV service providing platform is assigned a unique $uid_i\in UID$. A UAV contains multiple homogeneous or heterogeneous sensors. A set of sensor types is defined as $ST=\{s_1^t,s_2^t,s_3^t\ldots,s_n^t\}$. A set of sensor is defined as $S=\{s_1,s_2,s_3\ldots\ s_k\}$ where each sensor s is a 3-tuple represented as $s=\langle id,s_t,\mathcal{S}\rangle, s_t\in ST$. The tuple- id is a unique identifier allocated by the system to the sensor. Any location is represented as a 2-tuple $loc=\langle lat,lon\rangle$, where the

lat and lon represent the latitude and longitude values of the location, respectively. The availability of a UAV for certain application depends on its state, $\mathcal{U}$ holding values 0, 1, and -1 for unavailable, available and in-flight status, respectively. Based on the different UAV-related attributes, a UAV and an application are defined as:

Definition 4. *A UAV is defined as a 5-tuple and represented as* $uav_i = \langle uid, S_i, Loc, E_{res}, \mathcal{U}_i, Loc_h \rangle$*, where* Loc_h *is the home location of the UAV. A UAV can have multiple sensors, represented as a set* $S_{uav} = s_i$*,* $S_{uav} \subset S$*. Similarly, an application is represented as a 3-tuple* $App = \langle A_{id}, A_{type}, A_{loc} \rangle$*.*

A_{id} is the application ID provided by the system at the time of registration, A_{type} is the type of application depending on the task requested by the application. It may be any kind of sensing or actuation, capturing image, atmospheric sensing, mainly related to the type of sensors to be used for the application. A_{loc} is the location of the task requested by the end-user. The set of physical UAVs and virtual UAVs are denoted as $pUAV$ and $vUAV$, respectively.

A. UAV Selection and Allocation

The inputs from the end-user in their application, along with the system data about the available UAVs are used for the selection and allocation of UAVs to an application. Fig.

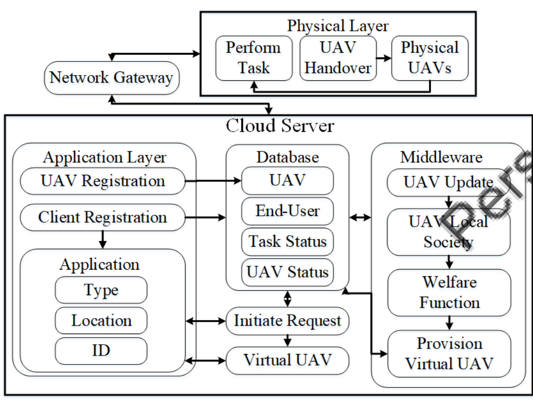


Fig. 5: UAV selection and task completion process.

5 represents the flow for the selection process at different components level. First, the end-user initiates a request for a UAV service through its application. The system uses the input from the end-user and determines the type of sensors required for the application. The system goes through several transitions, selects the most suitable UAV for the task through the selection algorithms, and maps it to the virtual UAV in the application.

We use a function f for selecting suitable sensor nodes for the application, requested by the end-user. Function f is represented in EquationV-A, where A_{type} is the application type ST_{app} is a subset of ST,

$$f(A_{type}) = \{ST_j \mid ST_j \in ST = ST_{app}\} \tag{6}$$

After carefully selecting the type of sensors for an application, a set of UAVs is selected from the available UAVs with the service provider. The function g_1 finds all the UAVs with the type of sensors required and the home location in the range of the location of the application. It also checks for the state of the UAV to avoid selecting the unavailable UAVs.

$$\begin{aligned} g_1(ST_{app}, A_{span}) = \{uav_i \mid uav_i.s.st \subset ST_{app}, uav_i \in UAV \\ uav.loc \in range(A_{loc}), uav.us \neq -1\} \\ = UAV_{app} \end{aligned} \tag{7}$$

The output of function g_1 is a set of UAVs, UAV_{app}, which is the set of all possible UAVs that are eligible for use in the application, also called the local UAV society.

B. Social Welfare Function

The social-welfare function, borrowed from the social-choice theory of Economics, is widely used in different applications. In a social welfare function, a group of agents votes for their preferable options, among the available ones. This function aims to make the voting and selection process unbiased and equally distributed towards the holistic welfare of the society. We use the Atkinson index-based social welfare function [17] to analyze the resource utilization of the UAVs. Atkinson index-based welfare function has been used for multi-robot task allocation problems [12] [13]. Atkinson welfare function model offers the flexibility to vary the magnitude of the penalty for maintaining equal resource utilization in a society. Based on the Atkinson index [13], we derive our welfare function as:

$$w^s = \frac{1}{n_u} \sum (r_i^u)^{(1-a_k)} \tag{8}$$

where n_u, r^u and, a_k are the number of UAVs in the local society, resource value for each UAV in the society and the Atkinson inequality aversion parameter, respectively. When a_k=1, the welfare function is represented as:

$$w^s = exp\left(\frac{1}{n_u} \sum (r_i^u)\right) \tag{9}$$

The eligible set of UAVs is further refined by selecting a UAV in any other application with similar task assignment and location. If any such UAV is found, it can be assigned to multiple applications. Finally, the allocation function f_{alloc} selects the UAV based on the social welfare function and allocates it to the application.

$$\begin{aligned} f_{alloc}(App) = f(g_1((UAV_{app})) \\ = \{uav_i \mid uav_i \in UAV_{app}, uav_i, \\ E_{res} > \ E_{hop} + E_{th}, d(uav_i, A_{loc}) = d_{min}\} \\ = UAV_{vir} \end{aligned} \tag{10}$$

The eligible set of UAVs and the finally selected UAV are recorded in the application for any future requests. The application is updated periodically or on-demand depending on the frequency of application usage.

During each cycle of request triggered by the end-user through the application, a UAV from the set of physical UAVs is mapped to the virtual UAV created for that application, UAV $\rightarrow$ vUAV.

$$f_{virtual}(UAV_{vir}) = \{vuav_{app} \mid UAV \rightarrow vUAV\} \tag{11}$$

In case of long duration mission, the mapping of virtual UAV is repeated when a UAV is worn off its energy and needs another UAV to take over. This is done without notifying the end-user.

As depicted in Fig. 5, the selection process is followed by the completion of a task, requested by an application. After the successful mapping of physical to virtual UAV, the UAV flies to the target location and performs the task until it is complete. The platform keeps a check on the energy level of the UAV to prevent the UAV from depleting its battery below the threshold level. When single UAV is unable to complete the task, it is replaced by another physical UAV. The cloud platform manages the handover of UAVs, autonomously, by analyzing the data from the UAV and selecting a replacement UAV, without any discontinuity in the service.

C. Simulation

We simulate the proposed architecture in our developed python environment. We generate tasks in an application area split into $n \times n$ grid-space.

Algorithm 1 UAV Selection

INPUT: n : Specified $n \times n$ grid space, p : Number of tasks, u : Number of UAVs ,
OUTPUT: Tasks completed by UAVs

```
for Each task do
    Find the local UAV society
end for
for Each local UAV society do
    if Local UAV society is not empty then
        for For each UAV in local UAV society do
            main UAV = UAV
            Calculate the residual energy of the society
            if e_res > e_max then
                e_max = e_res
            end if
            Task completed
        end for
    end if
end for
```

Algorithm 2 Virtualization

INPUT: n : Specified $n \times n$ grid space, p : Number of tasks, u : Number of UAVs,
OUTPUT: Tasks completed by Virtual UAVs

```
for i = 1 to p do
    Assign virtual UAV to each task
end for
for Each task do
    Find the local UAV society
    Assign list of UAV society to Virtual UAVs
end for
for For each task do
    if Task location not served already then
        Find physical UAV and map to virtual UAV
        Check energy
        while Task not complete do
            Find physical UAV and map                    ▷ Persistent service
            Check energy level and perform remaining task
        end while
    end if
end for
```

For each task, a local UAV society is generated. The UAV selection algorithm 1 uses the Atkinson index-based welfare function to calculate the overall welfare value for each UAV in the society. The UAV with maximum welfare value is selected for the task.

VI. Performance Evaluation

We evaluate the performance of our selection algorithm for the four parameters- number of UAVs, number of tasks, size of the grid-space, and the Atkinson aversion parameter (a_k). For different values of the aversion parameter, the penalty imposed upon the society for equal distribution of resource increases. From the plots in Fig. 6, we observe that an aversion value between 1.75 and 2.25 yields the maximum number of completed tasks for our architecture with virtual UAVs.

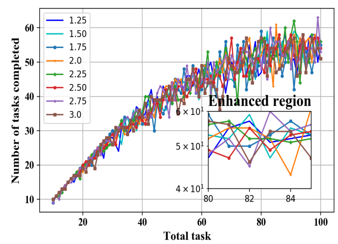


Fig. 6: Tasks completed for varying aversion parameter.

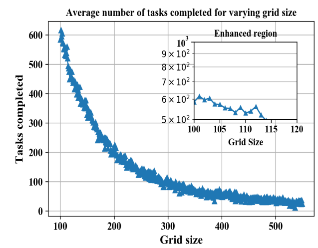


Fig. 7: Tasks completed for varying grid size.

A. Effect of Grid size

We evaluate the trend of task completion for varying grid-size, with a fixed number of 1000 UAVs and 1000 tasks. We observe a gradual decrease in the number of completed tasks with increasing grid-size in Fig. 7. We attribute this behavior to the distribution of tasks and UAVs placed over the grids. Smaller grid-size allows dense placement of UAVs and locations of tasks. Due to the compact placement of UAVs, local UAV societies have more UAVs available to serve the tasks. In the case of large grid-size, the distribution of tasks and UAVs becomes sparse. The sparse distribution of UAVs results in smaller local UAV societies, often resulting in no UAV available to complete a task. Therefore, we see a gradual decrease in the number of completed tasks in Fig. 7. However, it is possible to have the UAVs concentrated in a smaller area of the total grid-size and result in an increased number of completed tasks despite the large grid-size.

B. Effect of Task Count

We analyze the task completion ratio for varying numbers of task count in Fig. 8, keeping the number of UAVs and grid-size fixed. The plot follows an increasing trend with increasing number of tasks. The cause of this phenomenon is explained by the increasing number of tasks available within the range of UAVs. For fixed grid-size, increasing the number of tasks results in dense task location distribution. However, the task completion ratio goes down as the number of tasks increases. This is because of the restricted or fixed availability of UAVs. Hence we evaluate the trend for three different values of UAV count. For UAV count =n, the number of completed tasks is the highest. It is because more number of UAVs are available to complete the tasks. The distribution of UAVs in the grid-space plays a major role in deciding the task completion ratio. Hence, real-life scenarios can be represented as multiple subsets of the proposed architecture.

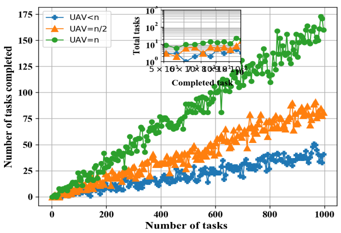


Fig. 8: Tasks completed for varying task count.

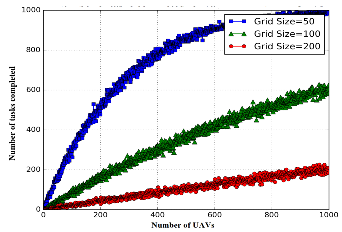
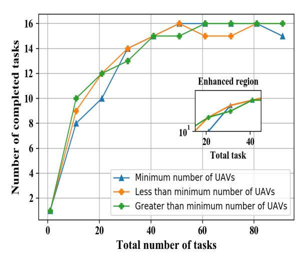


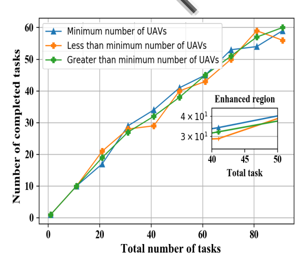
Fig. 9: Task completed for varying UAV count.

C. Effect of UAV Count

We study the effect of varying UAV count on task completion in Fig. 9. The phenomenon follows an almost linear trend as the number of UAVs increases, keeping the number of tasks and grid-size fixed. We further analyze the variations for three different grid-sizes. The smallest grid-size of 50×50 shows the maximum task completion. As the number of UAVs increases, the probability of a UAV being part of a local UAV society increases. As n_u for a task increases, more UAVs are available to complete the task associated with the UAV society. With small grid-size, the location of tasks and UAVs are closer to each other, most of them being at $1-hop$ distance. As a result, the number of completed tasks is more for small grid-size. While increasing the number of UAVs will always result in an increased number of completed tasks, UAV and its resources are limited and constrained in nature. Hence, the number of UAVs has to be decided to maximize the UAV resource utilization and serve maximum tasks efficiently. In Fig. 10 we analyze the trend of task completion with varying numbers of UAVs with fixed grid-space of 50×50 and 1000 tasks, for both with and without virtualization. The proposed



(a) Task completed without Virtualization



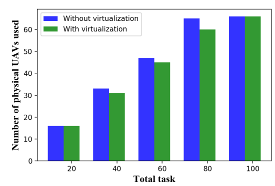
(b) Task completed with Virtualization

Fig. 10: Tasks completed for varying number of UAVs.

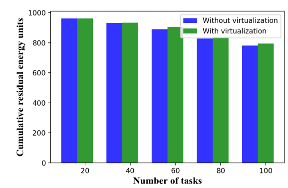
architecture achieves a higher task completion ratio in Fig10b. The general increasing trend in both the architecture is due to more number of available physical UAVs that can accomplish the tasks. However, we attribute the difference between the virtualization architecture and non-virtualization architecture to the fact that virtualization allows similar tasks with common interests such as location and sensor type to be completed as one mission, physically. More UAVs can serve more tasks, thereby increasing the total number of completed tasks. Since a UAV can serve a task only within its $1-hop$ range in the proposed architecture, the location of UAVs also affects the task completion. As an example, there can be UAVs available with desired sensors and battery level at the unreachable location.

D. Residual Energy Comparison and Analysis

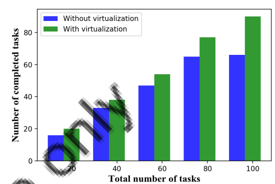
We compare the performance of simple non-virtualization architecture with the proposed virtualization architecture. In the simple non-virtualization architecture, we assume that only a single UAV can complete a task, i.e., a UAV with energy greater than or equal to the required energy can take up a task without distributing it to multiple UAVs. The physical



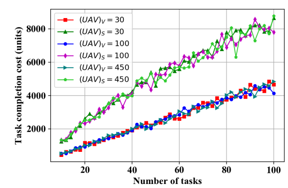
(a) Number of physical UAVs used



(b) Residual energy of the system



(c) Tasks completed



(d) Total cost analysis

Fig. 11: Comparison of virtual architecture with simple non-virtualization architecture.

UAVs are selected and assigned to the virtual UAV based on Equation (10). With an increasing number of tasks, multiple tasks are generated on the same grid location. In virtualization architecture, the same physical UAV is mapped to virtual UAVs of different applications with the same requests. As a result, the number of physical UAVs used for task completion decreases in Fig. 11a. As a result, the overall residual energy of the system increases, as evident in Fig. 11b. Fig. 11c compares the number of tasks completed for the two architectures. The essence of the proposed architecture holds for scenarios with multiple tasks in a grid location. For a condition with one task per grid location, the number of physical UAVs for both the architectures is similar. We speculate this as the cause for the behavior of our plots in Fig. 11a and Fig. 11b, where values are the same for both the architectures with task count of 20 and 100. However, the number of tasks completed in virtualization architecture surpasses the same in the other architecture because of its ability to perform a partial task by a UAV. Thus, the proposed virtualization architecture allows UAVs to perform tasks that require multiple UAVs. The proposed architecture uses multiple UAVs for a task involving heterogeneous sensors or a task that requires longer UAV flight time.

E. Cost Comparison and Analysis

Next, we analyze the total cost involved in performing the assigned task for the two architectures. We compare the cost for varying number of tasks in the range of 10 to 100 and UAVs ranging from 30 to 450 placed inside a grid-space of 30×30. Fig. 11d shows the difference in cost where UAV_V and UAV_S are the number of physical UAVs in the grid for virtualization and simple non-virtualization architecture, respectively. Virtualization allows multiple users to be served simultaneously using the same resources, whereas, for the simple non-virtualization architecture, individual UAV performs individual tasks. Additionally, in the case of virtualization architecture, the cost of a task is distributed among multiple end-users. In the case of simple non-virtualization architecture, each individual end-user bears the total cost of the task. As a result, we see that the cost procured for task completion in the proposed virtualization architecture reduces by approximately 46.5% of the cost incurred by the simple non-virtualization architecture. With an increasing number of tasks, we speculate further reduction in task completion cost, giving more margin for monetary benefits to all the actors involved.

VII. Conclusion

In this paper, we proposed a novel scheme for persistent and ubiquitous UAV services through the virtualization of physical UAVs. Our work enables the provision of UAV-as-a-service, overcoming the barrier of short flight time of a single UAV. Virtualization also increases the utilization of resources, avoiding multiple UAVs for redundant services. We incorporate a social welfare-based selection process for UAV selection. In the future, we plan to extend the work considering heterogeneous types of UAVs with different sensors onboard. This will challenge the selection algorithms and find the best possible UAVs within the target range. SDN-IoT and NFV can be explored as the enabling technologies for the proposed UAV virtualization architecture. A detailed energy consumption model and analysis should be carried out. It is important to note that UAV operations with virtualization may raise other concerns and issues that need to be addressed. Service provisioning related parameters can be evaluated, and more robust schemes can be introduced.

References

[1] M. Yuriyama and T. Kushida, "Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing," in *proceedings of the 13th International Conference on Network-Based Information Systems*, Sept 2010, pp. 1–8.

[2] S. Misra, S. Chatterjee, and M. S. Obaidat, "On Theoretical Modeling of Sensor Cloud: A Paradigm Shift From Wireless Sensor Network," *IEEE Systems Journal*, vol. 11, no. 2, pp. 1084–1093, June 2017.

[3] S. Y. M. Mahmoud and N. Mohamed, "Toward a Cloud Platform for UAV Resources and Services," in *4th IEEE Symposium on Network Cloud Computing and Applications (NCCA)*, June 2015, pp. 23–30.

[4] C. Luo, J. Nightingale, E. Asemota, and C. Grecos, "A UAV-Cloud System for Disaster Sensing Applications," in *81st IEEE Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–5.

[5] B. Nogales, V. Sanchez-Aguero, I. Vidal, F. Valera, and J. Garcia-Reinoso, "A nfv system to support configurable and automated multi-uav service deployments," in *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, ser. DroNet'18. ACM, 2018, pp. 39–44.

[6] S. Jeong, O. Simeone, and J. Kang, "Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, March 2018.

[7] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, F. Ono, and R. Miura, "On a novel adaptive uav-mounted cloudlet-aided recommendation system for lbsns," *IEEE Transactions on Emerging Topics in Computing*, no. 1, pp. 1–1, 2018.

[8] B. H. Y. Lee, J. R. Morrison, and R. Sharma, "Multi-UAV control testbed for persistent UAV presence: ROS GPS waypoint tracking package and centralized task allocation capability," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 1742–1750.

[9] H. Park, B. H. Y. Lee, and J. R. Morrison, "Persistent UAV security presence service: Architecture and prototype implementation," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 1800–1807.

[10] T. Bai, J. Wang, Y. Ren, and L. Hanzo, "Energy-Efficient Computation Offloading for Secure UAV-Edge-Computing Systems," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2019.

[11] S. Park, C. S. Shin, D. Jeong, and H. Lee, "DroneNetX: Network Reconstruction Through Connectivity Probing and Relay Deployment by Multiple UAVs in Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, Nov 2018.

[12] M. H. Kim, S. P. Kim, and S. Lee, "Social-welfare based task allocation for multi-robot systems with resource constraints," *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 994 – 1002, 2012.

[13] M. Kim, H. Baik, and S. Lee, "Resource Welfare Based Task Allocation for UAV Team with Resource Constraints," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 3, pp. 611–627, Mar 2015.

[14] B. Galkin, J. Kibilda, and L. Da Silva, "A stochastic model for uav networks positioned above demand hotspots in urban environments," *IEEE Transactions on Vehicular Technology*, 2019.

[15] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, "On a novel deep-learning-based intelligent partially overlapping channel assignment in sdn-iot," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 80–86, 2018.

[16] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in sdn-iot: A deep learning approach," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5141–5154, 2018.

[17] A. B. Atkinson, "On the measurement of inequality," *Journal of Economic Theory*, vol. 2, no. 3, pp. 244 – 263, 1970.