An adaptive task scheduler for a cloud of drones

* Hazzaa N. Alshareef
College of Computing and Informatics
Saudi Electronic University
Madinah, Saudi Arabia
h.alshareef@seu.edu.sa

Dan Grigoras
Department of Computer Science
University College Cork (UCC)
Cork, Ireland
grigoras@cs.ucc.ie

***Abstract*—Drones are now being widely used in different civilian applications, such as delivering shipments to consumers, as proposed by Amazon, and providing internet access to users, as offered by Facebook and Google. Drones can also contribute in emergencies by helping to find victims in places that are not reachable by rescuers, as well as assisting emergency centers to better manage a reported emergency. However, drones have a short flying time due to limited battery life. Therefore, a reliable strategy that minimizes energy consumption and uses collaborative working is required in order to increase drones' ability to operate for longer periods in emergency situations. This paper presents an adaptive task scheduler that allows tasks to be shared/transferred among the drones in a cloud of drones, in order to extend the operational time, achieve faster task execution and, at the same time, reduce the usage of each drone's resources. The ultimate result is an extension of battery life that leads to longer flying and service time for individual drones.**

Keywords—Cloud of drones; emergency; battery power; task scheduling; services

I. INTRODUCTION

There is a high level of interest in drones and their usage in civilian applications [1-7]. One of the most important features of using drones is their ability to reach places faster than humans. Furthermore, drones can reach locations that are sometimes difficult for humans to access and are less costly than, for example, helicopters [8]. Access difficulties could be due to the nature of these places or the risk associated with being there, as is the case with many post-disaster locations. With camera and other sensor capabilities, drones can survey the locations being monitored. One example could be managing rescue and recovery operations after catastrophic events (e.g., floods, fires, and earthquakes). Another strong feature of drones is their ability to provide means of communication using the drones' resources. For example, drones can act as communication relays to exchange data, as well as providing the ability to access the internet by acting as an access point or base station to the area over which they are flying.

However, drones face a number of challenges, including limited battery life and communication capabilities. Drones execute a small set of tasks (pre-loaded) that are known in terms of the resources they consume. For example, for video streaming, we can determine the energy consumed/minute or hour. The same applies to the Voice over Internet Protocol (VoIP). In practice, a number of solutions have been proposed to reduce these limitations and help save energy, which would lead to longer flying time. For instance, short-range line-of-sight (LoS) communication links can reduce the demand on drones' resources, particularly energy.

Building a cloud of drones was considered in a previous paper [9]. Drones can connect with each other to create a cloud of drones that can share information and, more importantly, resources. Since drones have a limited battery life, which renders them incapable of executing long tasks or even high priority tasks that need to be executed immediately, the idea is to use the benefits of a cloud of drones to share task executions among multiple drones and avoid intensive consumption of one drone's resources. The cost is, therefore, shared.

In this paper, we propose an adaptive task scheduler for multiple drones as part of the cloud of drones' model. Based on the energy level of each drone and the task priorities, the scheduler decides the order of execution and task sharing. This paper presents this scheduler in detail and evaluates its effectiveness.

The rest of the paper is organized as follows. Section II discusses existing systems and related work. Section III provides a list of tasks considered in our proposal and section IV presents the design of the adaptive task scheduler. Section V presents the experimental set-up and details the evaluation of the results. Section VI provides conclusions and plans for future work.

II. RELATED WORK

Both unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) can be considered to support rescue operations in emergencies. Loss of communication is one of the most critical issues that can limit their usage. In [10], an interesting approach is discussed that mainly aims at allowing the dynamic setting of communications between UGVs and UAVs with humans in the location of an emergency. Authors use decentralized (i.e. ad-hoc) networks, since it can be assumed that the current infrastructure is down. A mobile cloud is used in this paper to create a cloud of UGVs and UAVs. The store-and-forward routing protocol is modified to allow message routing without buffering packets. The authors rely on a well-known project, called Serval mesh, which provides a messaging protocol for disasters.

An interesting use of drones in emergencies is brought by [11], which provides a means of communication in the case of a large-scale disaster. The idea is to implement LTE femtocell base stations on drones to offer an alternative network infrastructure if the current one is affected. Authors designed an algorithm to help identify the number of drones necessary to cover an affected area, as well as the optimal locations for the flying drones.

A flying ad-hoc network (FANET) [12] protocol has been suggested due to some of the limitations of using multiple drones in an application, such as the need for expensive and complicated hardware to allow drones to communicate with ground base stations or a satellite. The reliability of these communications/links and the high possibility of disconnections are other limitations to take into consideration. According to the paper, a FANET is viewed as a modified form of MANET and vehicular ad-hoc network (VANET) that can be used to cope/deal with drones' special requirements, including their high degree of mobility, frequent changes in topology, a much longer average distance between nodes, different data delivery strategies from different types of sensors, etc. The paper includes interesting investigations of drones' networked issues. There are also some recent implementations of FANETs, such as in [13 - 15].

The idea of scheduling tasks in drones is not new. Many researchers have tried to minimize the impact of executing tasks to save energy. However, most of this research focuses on scheduling tasks and CPU usage, such as in [16, 17].

To summarize, comparing our work with the previously mentioned papers, we find that most of the research deals with CPU management aspects and how power consumption can be reduced based on CPU usage. Some papers consider classifying tasks and running each type of task in a different CPU core to achieve more efficient management of the power consumption. This research is concerned with sharing tasks among drones in a way that draws less from the drones' resources, which will lead to less power consumption when the task load is divided among multiple drones. Other papers collect basic information, such as task execution time and the speed of each task worker (a VM in some papers or the CPU core in others), which we do not believe is sufficient to decide which task should be executed by which worker. Here, more information is collected about each task, as well as extra data about each drone, to determine which drone should execute a task, and assess the ability to run a task among a number of drones at the same time.

III. EXAMPLES OF TASKS

Tasks can be anything that a drone can run and a user needs in the case of an emergency. However, for the purpose of this paper, two types of task are selected, as follows.

1) Video streaming (live view)

A task could be flying a drone to a particular place, starting a real-time video recording, and streaming it to the cloud to be watched in the emergency management centre or by one of the rescue teams near the emergency location. This is a demanding task because it requires a continuous video feed to a local user or remote centre. Multiple drones can be involved in this task to provide wide coverage of an emergency location, as well as for handing over from one drone to another if/when needed.

One important point is that this is a built-in task of commercial drones, controlled remotely by the user. However, the scheduler can signal to the drone to start/stop this. The task will start if there is no higher priority task and will stop if a new task of higher priority is ready.

2) VoIP

One of the most important services provided in the case of an emergency is voice communication. Rescuers can talk to each other as well as contact the emergency management office to request help or special equipment, for example. Hence, the task given to drones is to act as communication relays. Using the proposed scheduler, the service can still operate even if the drone that runs this service leaves or becomes unavailable due to low battery, by handing the task over to a different drone that has the capability to continue executing the task and ensure the service is provided to the users.

IV. ADAPTIVE TASK SCHEDULER

The concept behind the adaptive task scheduler is to share tasks among multiple drones. Two possibilities are considered: (i) using a *centralized* solution with the benefits of all the data being collected in one place and consistent decisions based on these data (as in Figure

1); and (ii) a *distributed* approach, in which drones directly negotiate with their neighbours with regard to how they share/split a task (see Figure 2).

A. Centralized approach

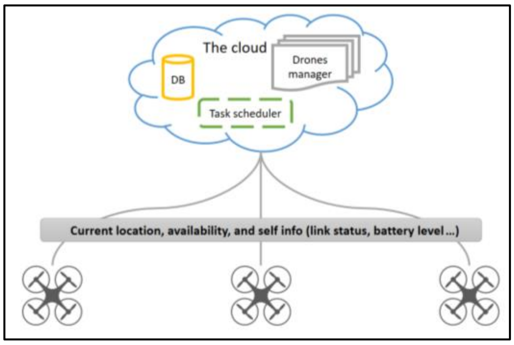

Figure 1 Centralized approach to allocating tasks

The centralized management component fulfils two main roles:

1) Collects and stores information from all drones in the cloud of drones.

2) Distributes or allocates tasks to one drone or multiple drones based on several factors, two of which are more important than the others:

- The status of the selected drone(s).
- The task characteristics, including task priority and estimated energy consumption/time unit (secs).

Two methods are used to collect the status of each drone: *push,* whereby each drone sends current information periodically, or once a change occurs; for example, if it flies to another location or its battery level is low; and *pull,* for which a request is sent to all connected drones asking for their current status, including location, battery level, and availability. This request is sent before a new task or set of tasks is allocated for execution.

The benefit of the push method is that knowledge about all the connected drones is already acquired. As a result, once a new task is dispatched, it will be directly allocated/scheduled to the most suitable drone(s). However, using this method will have a negative impact on the drones' resources (especially battery life). With the second method (pull), drones are contacted only if a new task arrives and a decision needs to be taken regarding which drone(s) will execute the task. This method requires less extensive communication. As a result, it will have a positive impact on the drones' resources (e.g., battery life).

Once the drones' status has been collected, they will be categorized in different groups to decide which drone(s) is suitable for which task. For example, drones that have a full (or almost full) battery can be used for long tasks, whereas drones that are close to each other can share their resources to execute a heavy computational or resource-intensive task(s). The following are the categories in which each drone can be included:

- Drones that can handle heavy computational or resource-intensive tasks, such as video streaming.
- Drones that can handle short and/or small tasks, such as accessing an interactive map or redirecting a help request to the emergency centre.

B. Distributed approach

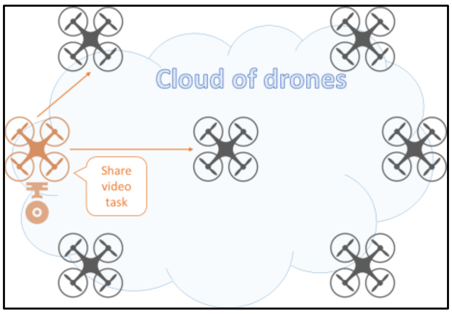

Figure 2 Drones negotiating task sharing inside a cloud of drones

As shown in Figure 2, a cloud of drones can be created that other drones could join and ask a neighbour to share a task. These drones can, therefore, cooperate locally inside the cloud of drones to execute the task. Two methods are offered in this research:

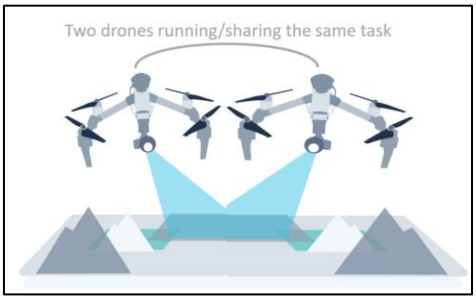

Figure 3 Two drones running/sharing the same task

- **<u>Direct communication</u>**: a drone can look for another drone to join and share a running task. For example, a drone might run a task and need another drone to share the task to save resources or provide wider coverage. In this case, a direct request is sent to a neighbour in the cloud of drones to share the task. Figure 3 presents a graphical explanation of how one drone can connect directly to another one to share the execution of a task, such as providing a live view of a certain location. However, an issue might arise here because of the sharing aspect, such as overlapping in the visual area (e.g. two drones sending a video of the same location). Thus, the drone that shared the request should take responsibility for avoiding this situation by changing location, for example, or providing a location that needs to be covered by the invited drone.

- **<u>Dedicated cluster</u>**: a drone might run a high priority task, such as VoIP communication for team members on the ground and look for other drones to share the task. The drones create a cluster to share the task, then invite other drones in the same area to join the cluster. Figure 4 shows a drone providing a VoIP service to ground users then starting to create a cluster of drones to share the execution of the VoIP service to the end users.

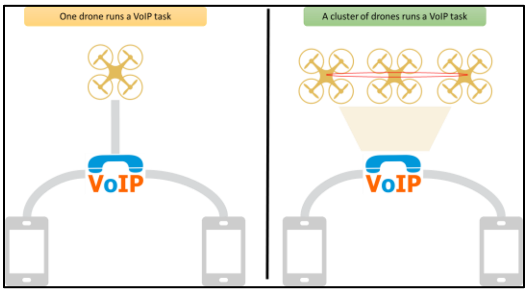

Figure 4 VoIP task run by one or a cluster of drones

C. Collaboration protocol

Before sharing a task, the drones need to reach agreement by exchanging messages. Therefore, a simple messaging protocol is proposed. Each drone should have an IP address or ID so that the drones are able to locate each other. Messages include sending a sharing request and replying to that request. Another type of message can be notifying other drones about an action that is about to happen, such as a drone leaving to charge the battery. Table 1 shows these types of messages with a description of each.

Table 1 Examples of messages exchanged in the collaboration protocol

Message	Sent by	Description
task_sharing	Requester drone	A drone that requests task sharing sends a request to one of its neighbours asking to share the execution of a task.
accept/reject	Selected drone	A reply is returned by the drone addressed. Depending on the status, the reply will accept or reject the task-sharing request.
ACK	Both drones	Acknowledgement (ACK) of the request.
Synch	Both drones	The two drones have to share the progress of the task and synchronize the execution of the running task.
share data	Requester drone	If the task requires data during its execution, the requester drone attaches all the necessary data to the message.
leaving	Both drones	If a drone detects a low battery and is about to leave, a message is broadcast to all connected drones.

D. Tasks handover framework

As part of the new scheduler, drones can transfer the execution of a running task to another drone that is part of the same cloud of drones. This migration of tasks might be needed due to some issue in the drone running the task, such as a low battery level or the drone needing to run a higher priority task that has just been allocated. Simply put, a drone that wishes to transfer the execution a task can send a request to all reachable neighbours to plan the task execution transfer. The task's description and progress are attached to the request to ensure that the task runs smoothly once the transfer is made. Once a reply is received from a neighbour to handle this task, the execution of a task transfer is started.

To avoid wasting resources, a timer is used here to wait for a reply from reachable neighbours. If the timer is due and no reply has been received, another attempt is allowed. However, if the timer for the second attempt is due and no reply is received, the task will be suspended until the drone is able to resume it (e.g., having recharged its battery and returned to operating normally).

Figure 5 illustrates the steps needed for one drone to transfer a task to another one. Figure 6 shows the type of messages that will be exchanged between a drone that is executing a task and the drone that will take over the task in order to execute it. In other words, one drone sends a request after attaching all the information required and the receiver drone sends an acknowledgement (ACK) message if it agrees to take over the task.

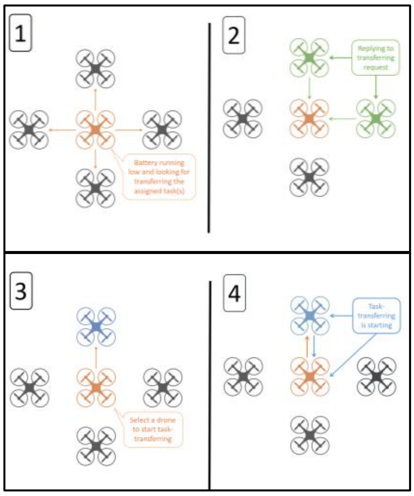

Figure 5 Steps needed for a drone to transfer the execution of a running task to another drone in the same cloud of drones

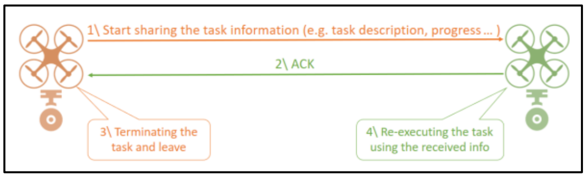

Figure 6 Messages exchanged between a drone that wants to hand over a running task and the drone selected to take over the task

E. Task scheduler

Each drone runs the task scheduler which decides, based on the *battery level and the priority of the task*, to stop other tasks. The scheduler informs other drones of this decision, to split the task with other drones, engaging in communication using the protocol mentioned above, or to call for another drone. In practice, we are suggesting a new drone task model, in which the set of tasks is controlled both locally and remotely. There are three main scheduler scenarios once a new task is received.

1) *The drone is idle*

The scheduler accepts the task if capable of running it (i.e., it has enough battery). However, if it is not able to run the incoming task due to limited resources, a clustering approach can be used.

2) *The drone is busy running a task*

The scheduler will check the priority of the newly incoming task with the current running task.

- If the new task has *higher* priority, the drone *stops* the current task and *starts* immediately the new one.
- If the new task has *lower* priority, the drone queues the new task.

3) *The drone is engaged in a cluster to share the execution of a task*

The scheduler will check the priority of the incoming task with that of the task executed by the cluster.

- If the new task has *higher* priority, the drone *stops* the current execution and notifies the members of the cluster of this action. It then *starts* executing the new task.
- However, if the new task has *lower* priority, the drone schedules the new task for after the current one has finished.

F. Data security

This paper does not deal with security but focuses instead on scheduling tasks executed by drones and how these can be shared or even transferred from one drone to another. However, some existing security protocols could fit easily here to protect data, such as using secure channels in communications between drones inside the cloud of drones and with end users. Moreover, a light encryption mechanism can be used if it can be verified that there will be no negative impact on the drones' resources.

V. Evaluation and Experimental Results

This section presents detailed information about the cost of the previously mentioned tasks (video and VoIP) in terms of power for the purpose of helping drones decide either to reject or accept a task (or share a task) based on how much the task received will consume or how many resources it might need if using a *distributed* approach. The idea of transferring the execution of a task from one drone to another to determine the impact of this in terms of both drones' batteries is also tested.

Since it is difficult to modify an existing drone's hardware (i.e. control unit), we used several Raspberry Pi to conduct our experiments. Each Raspberry Pi acts as a drone control unit. In general, there is no direct way to profile/report the power consumption or battery usage of a Pi. Some developers use external equipment (i.e. a power detector) and others calculate the cost from the CPU usage. For the purpose of this paper, we collected the power usage using a 10000 mAh (36 Wh) Cellularline portable power supply [18] that has a built-in screen to show the amount of power as a percentage. The following subsections present the experimental results of the two proposed tasks as well as a scenario for transferring tasks execution between drones.

1) *Video task results*

After ensuring the power bank was fully charged (100%), we plugged the Pi into the power bank. We started the video using a webcam that is connected to the Pi via a USB port. Once the video previewed on the screen, the change in the portable power bank was recorded. We found that running a video stream from the webcam for 30 minutes consumed 4% of the battery capacity.

Since the portable power has 10,000 milliampere hours (mAh) of battery capacity that means running the video task on the Pi will consume 400 mAh, although this amount of power is also shared with other functions on the Pi, such as booting the Pi and other operating system (OS)-related tasks (i.e. communication and control). Therefore, we have to exclude these services and tasks from this rate.

To obtain the amount of battery consumed by the Pi without the video task, we plugged the Pi into the same power bank after we ensured it was fully charged (100%). We ran the Pi in idle mode for 30 minutes (i.e. no video task running) to determine the amount of energy consumed. We found that running the Pi in idle mode for 30 minutes consumed around 2% of the portable battery, therefore the actual consumption which was 200 mAh.

We used a 5v portable battery (1 watt-hour or Wh). However, the v-value in real drones, such as the Phantom 4, is 15v, which would consume 3 Wh.

To summarize, running a video task for 30 minutes will consume around 200 mAh, which is almost the same as the power needed to run a Pi in idle mode for 30 minutes. This type of task needs 3 Wh to run on real drones, such as the DJI Phantom.

We can do some calculations based on the specifications for drones provided in the marketing information. For instance, the Phantom 4 comes with a 5,350 mAh battery, which allows up to 30 minutes of flying in calm conditions at a constant speed. This is around half the capacity of the portable power bank we used in our experiment. However, running the video task will share this battery with the flying task, which means the total flying time will be reduced. If we consider the results collected from running the video task (200 mAh) and subtract these from the battery capacity of a real drone, it will be affected by around 3-4%, which is not that much.

2) *VoIP task results*

The Pi was running a VoIP server and connected to a fully charged power bank (100%). Two Android-based mobile devices were used in this experiment. After registering the two Android devices as VoIP clients on the VoIP server that runs on the Pi, a call was established. We kept the line active for 30 minutes between the two clients. To ensure a voice was transferred through the communication link, we played a piece of music during the call.

We found that running an active VoIP call between two mobile devices managed by a Pi consumed around 7% of the battery capacity. As mentioned, the portable power has 10,000 milliampere hours (mAh) of battery capacity, which means 700 mAh.

From the previous section, the idle cost of running a Pi without executing a task is 200 mAh. Therefore, the actual cost of running a VoIP service between two clients through a Pi is 500 mAh. Since we used a 5v portable battery, the energy consumed is 2.5 Wh.

In summary, running a VoIP task to serve two end users for 30 minutes will consume around 500 mAh, which is around 2.5 times the power needed to run a Pi in idle mode for 30 minutes or running a video task. Based on the market specification of a well-known drone (Phantom 4), running a VoIP task could consume around 7.5 Wh (with a v-value of 15). To ensure the reliability of this task configuration, we managed to maintain a call between two registered Android devices for up to 1 hour without issues.

B. *Tasks handover results*

This section tests the idea of transferring the execution of a running task from one drone to another.

1) *Experiment scenario*

We assumed the following scenario: a drone is running two tasks at the same time (video and VoIP) and both are running normally. However, following a trigger (e.g., low battery level), one of these tasks is moved to another drone that is within reach and part of the same cloud of drones. Technically, the moved task is terminated in the first drone and started in the second drone.

2) *Experiment set-up*

For testing, two Raspberry Pis were used. They:

- were connected to the same network and each was assigned an IP address;
- were running a continuous communication channel (TCP socket) to listen to incoming messages;
- had VoIP server capability;
- were equipped with a camera, and
- were connected to a portable power bank that was fully charged (100%).

The first Pi (we called it Pi_A) runs a VoIP server (serving two clients) and in the same time running a video task. The second Pi (Pi_B) is idle and within reach of Pi_A. We defined the trigger here as a 30-minute timer. Once the timer is due, the execution of video task is transferred from Pi_A to Pi_B, after terminating the task in Pi_A. We reset the timer (30 minutes) to assess the impact of this transfer on both Pis in terms of battery usage.

3) *Experiment results*

Running both tasks (VoIP and video) on Pi_A for 30 minutes consumed 9% of the 10,000 mAh power bank (900 mAh). After the 30-minute timer was due, the video task was transferred to Pi_B, which consumed 6% of the 10,000 mAh power bank (600 mAh). We observed the change in power consumption after the second 30-minute timer was due and found that the consumption of Pi_B increased by 1% by 1% of the 10,000 mAh power bank (100 mAh) compared with the consumption before receiving and running the transferred task. The battery consumption of Pi_A decreased by 2% of the 10,000 mAh power bank (200 mAh) after terminating and moving one of the running tasks. Figures 7 and 8 present these results graphically.

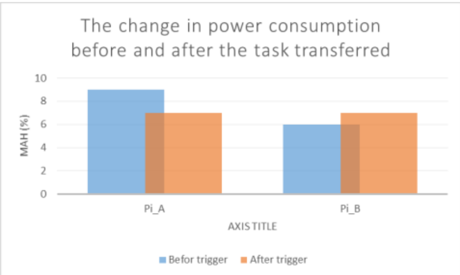

Figure 7 Bar chart showing the impact of using the proposed idea on the battery level of two Pis

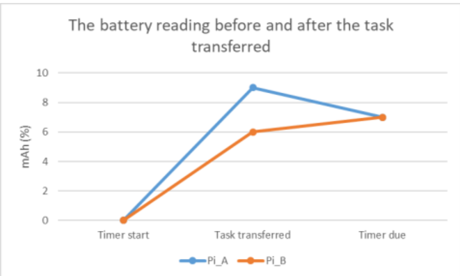

Figure 8 Graph showing the readings for both power banks used during the experiment

1) *Experiment summary*

The proposed task scheduler can improve drones' resources by reducing battery consumption once a task is shared or handed over to other drones.

2) *Experiment limitations*

One of the most noticeable limitations is associated with the power bank used in the set-up. The bank has a large capacity, which makes it difficult to read the change in running tasks, especially if the task is of short duration. We decided to examine tasks that would run for 30 minutes because most drones operate for 20-30 minutes. We were able to establish that the total percentage indicated on the power bank screen decreased faster when running two tasks compared with undertaking one task.

In addition, moving from one percentage to the next needs a Pi to consume 100 mAh to show the new percentage on the power bank screen, which means that even consuming 99 mAh is still considered as the same percentage. This could be acceptable since we ran these experiments as a proof of concept to show there is a positive impact of using the proposed task scheduler.

However, in future work, we need to investigate another way of reducing the power consumption. For instance, using a smaller power bank with less capacity so we can observe the change in percentage more precisely would be recommended. Alternatively, we could run tasks of longer duration (e.g., 1 hour or more).

VI. CONCLUSION AND FUTURE WORK

Drones are becoming increasingly useful in emergencies, as they have unique features that can help in such situations.

However, drones face a number of issues that make their use in emergencies difficult. The main issue is battery capacity, which mainly affects flying duration. Some drones cannot fly continuously for more than a few minutes (7-10) and some less than that. Even the more expensive drones cannot fly for more than 30 minutes. Therefore, extending battery capacity is essential. A larger battery would affect the ability to fly. Therefore, the direction should lie in minimizing the usage of a drone's battery. This could happen by reducing power consumption.

In this paper, we proposed an adaptive task scheduler that can run on a drone and make decisions that will help reduce power consumption. This scheduler can force running tasks to stop or to terminate if another task with higher priority needs to be executed. Furthermore, this scheduler can help one drone create a cluster of drones inside a cloud of drones. The aim is to engage multiple drones in executing one task that is resource intensive. Similarly, cloud computing capability is introduced here as a centralized solution to assign/share tasks between drones in a cloud of drones. This means that task management will be shifted to the cloud, which will have a positive impact on drones' resources, especially battery life.

We ran a number of experiments to ensure our proposed task scheduler is valid and able to achieve its goals. For instance, we executed video and VoIP tasks to determine how much these would consume and to discuss the impact of these tasks on real drones in terms of power. We can state that running a video task will consume the same amount of battery needed to run a drone with basic functions. In addition, it takes around double this amount to run a VoIP communication between two clients for 30 minutes.

The idea of transferring the execution of a running task from one drone to another was also tested. A drone can reduce battery usage by moving a running task to another drone inside a cloud of drones.

However, one critical task that drones run and that consumes most of the battery's capacity is flying. Therefore, we need to establish how much energy the flying task will consume and compare it with the available battery level/capacity. Since the goal is to increase the flying time, or at least not reduce it, the proposed scheduler could help by sharing tasks among multiple drones. For example, two drones could run a task whereby each drone can perform part of the task but consume less of the battery power normally required for the task.

As future work, we are planning to extend the experiments by testing the centralized approach as well as adding more tasks, such as proposing a new drone model for emergency management. Furthermore, we will consider the use of energy harvesting systems to deal with the limited battery life of drones by finding another source of power that can feed drones and improve their battery level.

REFERENCES

[1] J. George, P.B. Sujit and J. Sousa, "Search strategies for multiple UAV search and destroy missions", *Journal of Intelligent and Robotics Systems,* vol. 61, no. 1-4, pp. 355-367, 2011.

[2] Z. Sun, P. Wang, M.C. Vuran, M. Al-Rodhaan, A. Al-Dhelaan and I.F. Akyildiz, "BorderSense: border patrol through advanced wireless sensor networks", *Ad Hoc Networks*, vol. 9, no. 3, pp. 468-477, 2011.

[3] C. Barrado, R. Messeguer, J. López, E. Pastor, E. Santamaria and P. Royo, "Wildfire monitoring using a mixed air-ground mobile network", *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 24-32, 2010.

[4] A. Cho, J. Kim, S. Lee and C. Kee, "Wind estimation and airspeed calibration using a UAV with a single-antenna GPS receiver and pitot tube", *IEEE Trans on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 109-117, 2011.

[5] I. Maza, F. Caballero, J. Capitán, J.R. Martínez-De-Dios and A. Ollero, "Experimental results in multi-UAV coordination for disaster management and civil security applications", *Journal of Intelligent and Robotics Systems*, vol. 61, no. 1-4, pp. 563-585, 2011.

[6] H. Xiang and L. Tian, "Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle", *Biosystems Engineering*, vol. 108, no. 2, pp. 174-190, 2011.

[7] E. Semsch, M. Jakob, D. Pavlicek and M. Pechoucek, "Autonomous UAV Surveillance in Complex Urban Environments", *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (Volume 02),* Milan, Italy, 2009, pp. 82-85.

[8] H. Péter, D. László and V.P. János, "Civilian use of drones in the life of mining rescue helicopters", *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI),* Budapest, Hungary, 2016, pp. 000299-000302.

[9] H. Alshareef and D. Grigoras, "Multi-service cloud of drones for multi-purpose applications", *16th International Symposium on Parallel and Distributed Computing (ISPDC)*, Innsbruck, Austria, 2017, pp. 165-174.

[10] L. Baumgärtner et al., "Emergency communication in challenged environments via unmanned ground and aerial vehicles", *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, San Jose, CA, 2017, pp. 1-9.

[11] M. Deruyck, J. Wyckmans, L. Martens and W. Joseph, "Emergency ad-hoc networks by using drone mounted base stations for a disaster scenario", *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, New York, NY, 2016, pp. 1-7.

[12] I. Bekmezci, O.K. Sahingoz and S. Temel, "Flying ad-hoc networks (FANETs): a survey", *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254-1270, 2013.

[13] I. Bekmezci, I. Sen and E. Erkalkan, "Flying ad hoc networks (FANET) test bed implementation", *2015 7th International Conference on Recent Advances in Space Technologies (RAST)*, Istanbul, Turkey, 2015, pp. 665-668.

[14] A. Savita, P. Sharma and S.K. Tiwari, "Cloud based decentralized approach for FANET", *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, Paralakhemundi, India, 2016, pp. 2017-2020.

[15] D.Y. Kim and J.W. Lee, "Topology construction for flying ad hoc networks (FANETs)", *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, South Korea, 2017, pp. 153-157.

[16] I. Zhuravska, S. Borovlyova, M. Kostyria and O. Koretska, "Efficiency improvement of using unmanned aerial vehicles by distribution of tasks between the cores of the computing processor", *Technology Audit and Production Reserves*, vol. 6, no. 2(38), pp. 4-13, 2017.

[17] S.J. Nam, Y.G. Kim and S.W. Chung, "An energy-efficient task scheduler for mobile web browsing", *2017 IEEE International Conference on Consumer Electronics (ICCE),* Las Vegas, NV, 2017, pp. 188-189.

[18] Portable charger [Online]. Available at: https://www.cellularline.com (Accessed: 27 Oct 2018).