Automated Data Publisher for Disaster Relief

Muhammad F. Ardiansyah,
National Yang Ming Chiao Tung University,
Taiwan
mfardiansyah.eed08g@nctu.edu.tw

Timothy William,
National Yang Ming Chiao Tung University,
Taiwan
timwilliam.cs06@nycu.edu.tw

Samer T. Talat,
Industrial Technology Research Institute,
Taiwan
talat@itri.org.tw

Li-Chun Wang,
National Yang Ming Chiao Tung University,
Taiwan
lichun@g2.nctu.edu.tw

Po-Lung Tien,
National Yang Ming Chiao Tung University,
Taiwan
polungtien@gmail.com

Maria C. Yuang,
National Yang Ming Chiao Tung University,
Taiwan
mariayuang@gmail.com

***Abstract*—Nowadays, aerial drones can provide for tremendous support in our modern daily life application such as public safety application. Aerial drone applications require extensive hardware processing for the huge data collected using distant cloud or edge resources. This motivates us to develop data publisher for aerial drone system especially when only limited bandwidth and power are available. In this paper, a dynamic data publisher system is proposed. The dynamic data publisher system will adjust the data publishing rate based on a trigger mechanism. As an example, object detection results is used as the trigger. In an aerial drone scouting mission, whenever a person in need of help object is detected, the dynamic data publishing system will increase the publishing rate and vice versa. Our experimental results show that the proposed dynamic data publisher system will decrease the number of frames transmitted to the edge significantly. As a result, the life time of the aerial drone battery is extended due to the reduce in data published to the edge.**

***Index Terms*—aerial drone, edge, 5G, data publishing**

I. Introduction

Recently, there has been a growing interest in aerial drones and their applications to various sectors in the economy and in particular for applications pertaining to public safety [1]. In particular, the autonomous drone scout scenario focus is on the need of a public safety agency such as drones utilization in a disaster relief mission. Drones are commissioned to scout a disaster area while utilizing efficient data collection and efficient transmission over 5G communication networks. The collected data are then transmitted to the edge of the network for cognitive processing and decision-making. The current solution for disaster relief used mature fog and edge computing infrastructure such as in [2], [3], [4]. In such solutions, the drone fleet control shall provide low-cost and efficient rescue missions while taking into consideration the communication and the computing capabilities over different tiers of platform [5], [6].

Notably, few works focus on addressing the efficient usage of the bandwidth in disaster impact area. The question raised here is how bandwidth contention issue in multi-drone aerial disaster relief mission should be managed. In addition, how to dynamically adjust data publishing rate when needed to save bandwidth. To validate raised issue, we did an initial analysis regarding the bandwidth consumption of FullHD video streaming with 30 Frames Per Second (FPS) over 4G LTE, 4G LTE-A, and 5G networks. Technical reports from [7], [8] are used as references. We compare between uncompressed with compressed image frames over different compression strength and summarize them into Fig. 1. Based on this figure, it motivates us that: (a) sending uncompressed images is impossible even using a 5G network, (b) even with image compression, a single aerial drone can easily exhaust the available bandwidth of LTE, and 4G LTE-A network. In 5G network where we have more bandwidth, multiple drones can be supported. Although, the problem of bandwidth contention still remains.

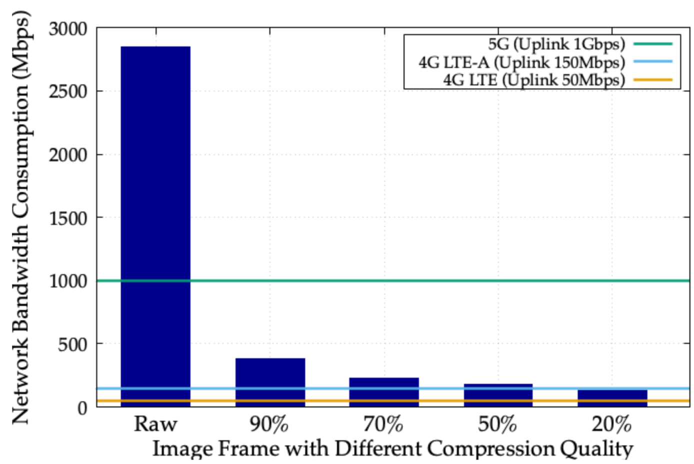

Fig. 1: Bandwidth Consumption of FullHD resolution (30FPS)

In this paper, we investigate the publisher system design to be able to dynamically control the publishing rate from the drone to edge based on the detection of Person in need of help (PiH) object as the trigger mechanism. PiH object detection and localization is considered as a real time scenario. Where the video, and GPS data from the aerial drone, as well as the PiH detection results from the object detection application running at the edge are all published in real-time continuously. The procedure of the dynamic data publisher is as follow:

- Start publishing data from the aerial drone to the edge at low data rate (e.g.: 10FPS) when no PiH object is found in an area.
- Whenever a PiH object is found in the area, increase the data publishing rate (e.g.: 30FPS) for period of time.
- Then, decrease the aerial drone data publishing rate again when no PiH object is found in an area.

Intuitively, the proposed design will allow for the aerial drone to publish data efficiently. The aerial drone will first start to publish with low data rate, then with increased data rate for a period of time when a PiH is found, then throttle down the data publishing rate again or maybe stop completely when the drone moves on to find the next PiH. The proposed algorithm will throttle and increase data transmit rate based on the detection of PiH in different scenarios. The rest of the paper is organized as follows. Sec. II presents an overview for the state of art solutions. Also, Sec. III provides an overview of the Dynamic Data Publisher (DDP) system. Up next, Sec. IV presents some experimental results and discussion of the proposed DDP system. Finally, Sec. V draws the conclusions of this paper.

II. Related Works

One of the key challenges in drone fleet network is the capability of data transfer while maintaining a dynamic network. In real-time application, this capability shall preserve the latency and energy consumption [6] of the drone. In [6], the author proposed a solution to execute computations locally on the target drone or offload them to other available drones. Another data transmission scheme is presented in [9], where a 3D path planning with height optimization is calculated based on channel condition for 3D path planning. In [10], unified resource broker implementation can be used to provide a centralized sub-channel and transmission power allocation and multi-server task offloading scheme based on directed acyclic graph partition and edge server selection algorithms. In [11], an intelligent task offloading scheme can offload action utilizing the deep Monte Carlo Tree search. The Monte Carlo Tree search will find the lowest latency or power consumption for the drones. The aforementioned approaches are useful. However, rather than directly reducing the bandwidth or power consumption, it offload the tasks.

On the other side, several works focus on drone trajectory planning framework for drone fleet mission in an emergency environment [12], [13]. This includes real time object detection in the field and re-planning of drone fleet routes based on the object detection classification results [12]. Besides, drone fleet shall cope with different constraints such as coverage area, quality of service requirements and battery capacity. In [13], a complex conjunctional optimization solution is provided to minimize the flight battery consumption and QoS cost within the coverage area. Dynamic drone-to-drone scheme is presented in [14] to decide autonomously whether to transmit data without any intervention from infrastructures. This proposed scheme is able to achieve semi optimal sum-rates with low signal-to-noise values. In [15], a classifier is proposed to predict and analyze for abnormal behaviour at the edge. Then, the required reaction latency is reduced by avoiding the predicted abnormal condition for connected drone client. In addition, a framework presented in [16] depicts a judicious combination of drone-based processing and edge-based processing to reduce wireless bandwidth consumption without compromising for video analytic accuracy or latency. However, the proposed work did not address the application bandwidth requirement. Obviously, the aforementioned approaches did not consider for data publisher optimization. Unlike the state of art, our proposed scheme provides for efficient data publishing scheme for drone fleet in disaster missions.

III. Dynamic Data Publisher System

In this section, we elaborate our proposed in-drone system which is called Dynamic Data Publisher (DDP). This system aims to maintain the data transmission rate at a certain level to reduce the bandwidth usage. In general, there are N important processes in this work: (1) how drone determine transmission rate which is triggered by .... TO BE CONTINUED ...

A. Scenario selection

In this work, an aerial disaster relief mission is selected as the scenario where the rescue team deploys multiple drones and streaming in a real-time to search for Persons in need of help (PiH), denoted as PiH objects. The video stream is extracted into frames, pre-processed, and published to the edge server through a Base Station. On the edge, it performs an object detection, determine the detection status, such as, PiH object(s) are/is found, and sends back the detection results. We set the object detection as the trigger mechanism to determine publishing rate of the drone. The quoted detection algorithm methods above can be any type of detection algorithms, such as YOLOv3 [17], YOLOv4 [18], SSD [19], G-RCNN [20], or any other advanced object detection methods. Apart from that, other algorithm can also be used as the trigger mechanism. For example, drone battery level and drone flying speed.

B. Architectural Model

In Fig. 2 the architecture and the deployment of the dynamic data publisher system is presented. In this setup, we have the drone which acts as the source of the data, as well as the edge which acts as the compute unit. The drone the edge are connected via a wireless communication medium via a data transmission protocol, it can be either a 4G or 5G network. In the following subsection. The dynamic data publisher system

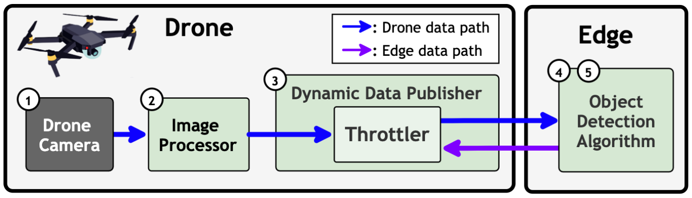

Fig. 2: Dynamic Data Publisher System Architecture and Deployment

has one key module which is the throttler module which is derived in Algorithm 1.

An overview of the architecture workflow is as follows:

1) The drone camera captures the video data during flight.
2) Image processor module then extracts the image frame from the video and then perform image compression before passing it to the dynamic data publisher system.
3) This module takes as an input the object detection result from the edge and then decide on when to increase or throttle the data publishing rate to the edge.
4) At the edge, the object detection algorithm consumes the published data and then perform PiH detection.
5) Once PiH detection is completed, the object detection algorithm forwards the detection results back to the throttler for future decision making.

C. Dynamic Data Publisher Algorithm

The proposed DDP algorithm aims to adjust the total number of frames send to the edge server, from α to β, and vice versa. Let us assume that there N drones (D_i; $D = D_1, D_2, ..., D_N$) are deployed and started video streaming as soon as they are flying. Each drone is attached with a camera to stream video (V_i; $V = V_1, V_2, ..., V_N$) and is having its video streaming extracted into image frames (F_{ij}; $F = F_{i1}, F_{i2}, ..., F_{iM}$). These image frames may be discarded or published to the edge. Here, some image frames are discarded as they do not contain useful information (e.g. there is no PiH object found). Each drone D_i adjust its publishing rate with the following formula:

$$\mu_i(t) = \begin{cases} \alpha, & \text{if } \hat{y}_i(t) = \text{true} \\ \beta, & \textit{otherwise} \end{cases} \tag{1}$$

where $\hat{y}_i(t)$ acts as a flag to inform drone D_i to set the $\mu_i(t)$ into α if the value is $true$, otherwise set the value back as β; $\alpha\ and\ \beta$ are measured as FPS. In our scenario, we define $\hat{y}_i(t)$ as a detection status, where each captured frame F_{ij} in the edge will has its inference calculated and resulted some detected objects, hence, $\hat{y}_i(t)$ is expressed as:

$$\hat{y}_i(t) = \begin{cases} \text{True}, & \text{if } \sum PiH_{ij} > 0 \\ \text{False}, & \textit{otherwise} \end{cases} \tag{2}$$

where PiH_{ij} is the detected PiH objects on each frame F_{ij}. When there are more than one PiH objects detected, $\hat{y}_{ij}(t)$ will be set as $true$, otherwise, it is $false$.

On the drone side, initially DDP will only send image data to the edge in every 3 frames. Let say drone D_i streams video for one minute. There will be K number of frames extracted ($\sum_{j=0}^{K} F_{ij}; \forall F_{ik}\ in\ F_{iM}$) with P number of discarded frames and Q number of published frames ($P + Q = K$). By discarding some extracted image frames, the bandwidth consumption of each drone will be reduced and the expected number of supported drone will increase. The expected number of supported drones is calculated by:

$$\mathbb{E}(D) = \frac{\omega^{DL}}{\omega_i^{\theta}(t)} \tag{3}$$

where ω^{DL} is the bandwidth capacity of a certain network (i.e. 4G LTE has $50Mbps$ downlink capacity). $\omega_i^{\theta}(t)$ is the average bandwidth consumption of drone D_i at time interval t. Each drone's bandwidth consumption is given by:

$$\omega_i^{\theta}(t) = \frac{\sum_{i=0}^{T} \omega_i^{UP}(t)}{T} \tag{4}$$

where $\omega_i^{UP}(t)$ is the current uplink bandwidth consumption or Drone D_i in time interval t, and T is the time interval to calculate the average value. In this case, we set $T = 60$ as we calculate the average value in every minute.

Finally, the Dynamic Data Publisher algorithm 1 is given for each drone D_i.

Algorithm 1 Dynamic Data Publisher Algorithm

Require:
Current Publishing Rate of a drone $\mu_i(t)$;
  Default value is set as 10
Image frames of drone F_{ij};
  $F = \{F_1, F_2, ..., F_n\}; \forall i = frame\ j-th$;
Sign status from edge server $\hat{y}_{ij}(t)$;
Ensure:
$\mu_i(t)$ is updated based on the Sign status $\hat{y}_{ij}(t)$.

for $j = 1$ to M **do**
  Send image frame F_{ij} to edge server
  Has the edge server to consume and process image
  Has the edge server to calculate $\hat{y}_i(t)$
  Receive $\hat{y}_i(t)$ from the edge server
  if $\hat{y}_i(t) = true$ **then**
    Set $\mu_i(t) = \alpha$
  else
    Set $\mu_i(t) = \beta$
  end if
  Discard or send frame $F_{ij}(t)$ to the edge server according to $\mu_i(t)$.
end for

D. Dynamic Data Publisher Deployment

As for the deployment, the will be two systems deployed: (a) Dynamic Data Publisher (DDP) which is deployed on the drone, and (b) Detection Application (DetApp) which is deployed on the edge server. DPP acts as the video stream extractor and periodically publishes them to the edge server, and DetApp processes the captured images and detects the PiH object(s). Every time DetApp finished the detection, the results are sent to DPP, enabling it to adjust the current FPS value accordingly. Both systems are orchestrated under docker containerization. As illustrated in Fig. 3, steps of the deployment are detailed as follows:

1) Deploy and start DetApp system on the edge server.
2) Deploy the DDP system on the drone and send the drone on the disaster impacted area.
3) Once DDP is running, it extracts video stream into image data, and each image may be published to the DetApp according to the throttler algorithm. On the pre-processing step, the image is compressed and is being tagged with useful information, such frame id, drone id, publish timestamp, and GPS
4) On the same time DPP also send the telemetry data which cannot be enriched on the tagging pipeline, such as, expected bandwidth usage and transmission property.
5) On the edge, DetApp consume the image data, prepare the data for the detection: (a) extracts the consumed message and (b) de-compresses the image.
6) Once data has been pre-processed, the image data is sent to the detection algorithm to detect PiH object(s).
7) Every time DetApp finished the detection, it sends the results to the DDP.
8) On the drone side, the detection result of each image is used to adjust the FPS accordingly. When PiH object(s) found, adjust the FPS into 30FPS, otherwise, set as 10FPS.

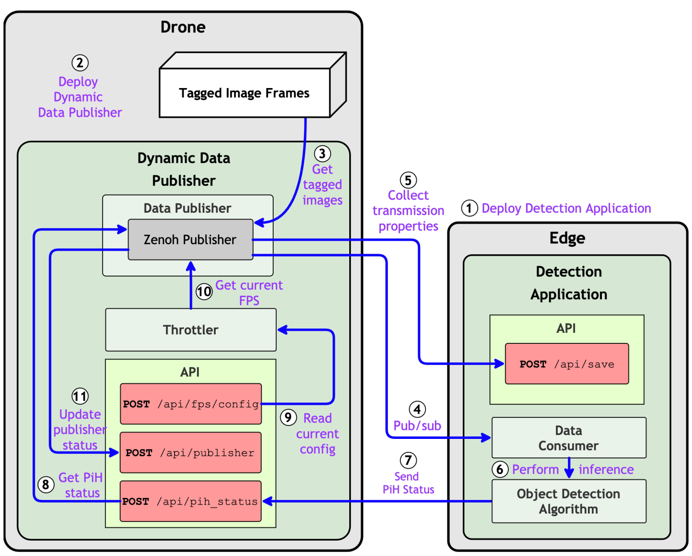

Fig. 3: Dynamic Data Publisher Deployment

E. Energy Consumption

We adopt the energy consumption model derived in [6] and adjust the calculation accordingly. The total energy consumption of drone D_i is calculated based on (a) the computation of extraction and compression of image frames, and (b) communication due to the publishing of extracted image frames to the edge.

1) Computation Energy Consumption: Each drone D_i will perform two actions, first, extracting video streaming from the camera. Second, compressing the image with certain level of compression. On the drone, the Latency to deal with the processing of images in every second is expressed as:

$$T_i^{frame}(t) = \frac{\rho \times \beta_i(t) \times \phi_{ij}(t) \times \mu_i(t)}{f_i} \tag{5}$$

where ρ $(0 \geqslant \rho \geqslant 1)$ is an offloading coefficient; $\beta_i(t)$ $(\beta \geqslant 0)$ is a computational complexity of the task [21] to process a tuple of images in a second which can be derived as:

$$\beta_i(t) = \frac{f_i}{\phi_{ij}(t) \times \mu_i(t)} \tag{6}$$

and f_i is the CPU frequency of drone D_i, while $\phi_{ij}(t)$ is the compressed and encoded image of Frame F_{ij} which can be given by:

$$\phi_{ij}(t) = ENCODE(\Psi(F_{ij}), \gamma) \tag{7}$$

where $\Psi(F_{ij})$ is the original image size of frame F_{ij} (e.g. a FullHD image has a size of $5.93MB$); γ is the compression strength of the image $(0 > \gamma \leqslant 95)$.

Finally, the computation energy consumption of each drone D_i can be formulated as:

$$E_i^{comp}(t) = k \times f_i^{\sigma} \times T_i^{frame}(t) \tag{8}$$

where f_i^{σ} is the computational power of drone D_i; σ $(2 \geqslant \sigma \geqslant 5)$ denotes the path loss exponent. According to [22] σ can be set as 3; According to [23] k is a positive constant. As in [22] k can be set as 1.25×10^{-26}.

2) Communication Energy Consumption: We adopt the calculation of the communication energy consumption from [24], [25], [26], and it can be expressed as:

$$E_i^{comm}(t) = \left(P_i^{Rx} \times \frac{\rho \times \beta_i(t) \times \phi_{ij}(t) \times \mu_i(t)}{\alpha_i^{UP}(t)}\right) + E_i^{Rx} \tag{9}$$

where $\alpha_i^{UP}(t)$ represents as the uplink rate of the drone D_i to the edge server; E_i^{Rx} denotes the transmission latency to receive back the data from the edge server for each published image frame F_{ij}. Formula from [24], [25] is used to derive following calculation:

$$\alpha_i^{UP}(t) = \omega_i^{UL} \times log_2\left(1 + \frac{P_i^{Tr} \times g(i,BS)^{-\sigma} \times |h_0|}{N_0}\right) \tag{10}$$

where ω_i^{UL} is the uplink channel bandwidths between drone D_i and the Base Station; P_i^{Tr} is the transmission power of the drone D_i; h_0 represents the complex Gaussian channel coefficient based on the complex normal distribution CN(0,1)

[27]; N_0 denotes the Additive White Gaussian Noise (AWGN) [28]; $g(i, BS)$ acts as the distance between Drone D_i with the Base Station. By adopting [29] $g(i, BS)$ can be derived as:

$$g(i, BS) = [(X_{BS}-X_i)^2+(Y_{BS}-Y_i)^2+(Z_{BS}-Z_i)^2]^2 \quad (11)$$

where $g(i, BS) \geqslant r$; r is the maximum communication radius of Drone D_i to the Base Station.

Furthermore, since we define the application on the edge server will perform an object detection which is resulting much more smaller transmission data size [30] (e.g. responding total number of detected objects), the transmission latency to receive back the results (E_i^{Rx}) is neglected. Therefore, the total communication energy consumption of drone D_i can be described as:

$$\begin{aligned} E_i^{comm}(t) &= P_i^{Rx} \times \frac{\rho \times \beta_i(t) \times \phi_{ij}(t) \times \mu_i(t)}{\alpha_i^{UP}(t)} \\ &= P_i^{Rx} \times \frac{\rho \times \beta_i(t) \times \phi_{ij}(t) \times \mu_i(t)}{\omega_i^{UL} \times log_2\left(1 + \frac{P_{Tr}(t) \times g(i,BS)^{-\sigma} \times |h_0|}{N_0}\right)} \end{aligned} \quad (12)$$

3) Total Energy Consumption: In this work, the moving energy consumption calculation is excluded as we assume that each drone D_i will perform similar moves. Finally, total energy consumption can be derived as follows:

$$E_i(t) = E_i^{comp}(t) + E_i^{comm}(t) \quad (13)$$

IV. Results and Discussion

In this section, we evaluate our proposed approach under a simulation environment with real hardware configuration parameters. ... .

A. Configurations

Referring to [6], [31], [32], [33] the system parameters of DDP system are summarized in Table I.

TABLE I: System parameters of DDP

Parameter	Value		Parameter	Value
ω_i^{UL}	1 MHz		α	30 FPS
N_o	$-100\ dBm$		β	10 FPS
$P_i^{Tx}(t)$	1.258 W		T	60 seconds
h_0	$CN(0,1)$		γ	$10-90$
k	1.25×10^{-26}		ω_i^{DL} (4G-LTE)	50Mbps
f_i	$1.43\ GHz$		ω_i^{DL} (4G-LTE-A)	150Mbps
σ	3		ω_i^{DL} (5G)	1000Mbps

Drone Transmission power use 1.258 W [34] ....

B. Another subsection ...

In the experiment is the experiment defined before, this beginning can modified to be more accurate and define more about the experiment or refer to it if already introduced a simulation of the object detection algorithm is used instead of a full object detection algorithm. This is done so that the detection results of a video can be consistent. Another reason for the use of simulation is simplicity, as we don't need the full object detection algorithm to test for the performance of the dynamic data publisher. The simulated object detection algorithm is based on the implementation in [35]. This specific implementation is chosen as it is one of the object detection implementation designed to perform disaster relief mission. The object detection simulation outputs a boolean (True/False) value once every 33ms (for a 30FPS video) to indicate if PiH object is detected in an image frame or not. The sample video used is 1 minute and 30 seconds of length, with a total of 2698 image frames. In the experiment, we set the publishing rate to 10FPS when no PiH is found, and 30FPS when a PiH is found. In the following subsections, experiment results regarding DDP system response time, bandwidth savings, energy consumption, as well as applicability of DDP system towards other mission is presented.

C. DDP System Response Time

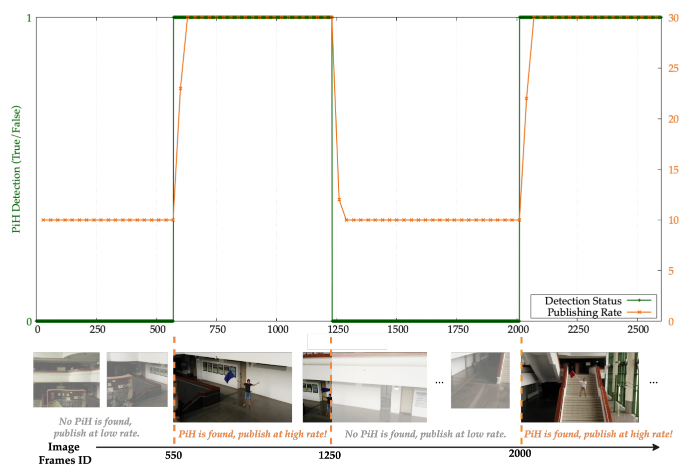

Fig. 4: Dynamic Data Publisher Responsiveness Test

In Fig. 4, the performance of DDP system is evaluated. The y-axis FPS measurement is taken once every second. There are two instances where PiH object is present. The first instance of PiH object occurrence is at image frame ID 571-1230, the second instance is at image frame ID 2011-2698. In the first instance, the dynamic data publisher algorithm starts to increase the data publishing rate at image frame ID 581, and then it reaches maximum data publishing rate at image frame ID 630. This gives a delay of 49 image frames or below 2 seconds for the dynamic data publisher algorithm to reach maximum data publishing rate. Next, when the PiH object is no longer detected. In the first instance of PiH object occurrence, the PiH object is no longer detected starting from image frame ID 1231. The dynamic data publisher starts to discard the publishing rate from image frame ID 1260, and then it reaches the minimum data publishing rate at image frame ID 1290. This gives a delay of 30 image frames or 1 second for the dynamic data publisher algorithm to reach minimum data publishing rate. Details on how the dynamic data publisher system performs for the second instance of PiH object occurrence (frame ID 2011-2698) are the following. The dynamic data publisher system starts to increase the data publishing rate at frame ID 2022, and then reaching the maximum data publishing rate at frame ID 2070. This gives a delay of 48 image frames to reach maximum data publishing rate.

D. Bandwidth Consumption

In Fig.5, the bandwidth consumption of image frames publishing under different image compression quality. Here, uncompressed image is not used as it is significantly larger in size and hence takes up more bandwidth. It is true that compressed image will have lower quality compared to uncompressed image, but this difference of quality will not affect the object detection algorithm detection results. A 'person', and 'flag' object will still be able to be detected. Although, visually, the image frame will look worse, the lower the quality setting are. The object detection algorithm that consumes the data will still be able to perform inferencing and produces the same detection results.

In this figure, higher image compression quality will produce better image (less pixelated when zoomed in), but with the drawback of increase in image size. As for the network bandwidth consumption, it is measured for the publishing of 30 image frames per second. In this comparison, we used a sample video for showcasing TV display and is available on YouTube [36]. We chose this video as it has varying scenes and is available across different video resolutions. The test is done by first extracting the image frames from the first 30 seconds of the sample video, and then publishing it to the edge. In addition, we can see that by using standard publishing method, the 4G LTE network will not be able to support for the publishing of HD image frames. Even by lowering the compression quality, the publishing of image frames will saturate the available bandwidth of 4G LTE network. As for 4G LTE-A network, HD image frames publishing is now possible under standard image publishing with image compression quality of 50. Finally, for 5G network, with more abundant network bandwidth, 4K image frames publishing with image compression quality of up to 70 is possible under standard image publishing. With that said, a significant bandwidth saving can be achieved when DDP system is implemented to dynamically adjust the data publishing rate. In the case of 4G LTE network, HD image frames publishing is not possible, albeit only with image compression quality of 10. In the case of 4G LTE-A, the publishing of HD, Full HD, as well as 2K video are now possible under various compression quality settings. Moreover, for 5G, with the utilization of DDP system, resolutions up to 4K will be supported. Using 5G, image frame with resolution up to 2K will be able to be supported fully while preserving compression quality of 90. With DDP system, we can expect to be able to support for publishing of image frame with higher resolution compared to standard publishing.

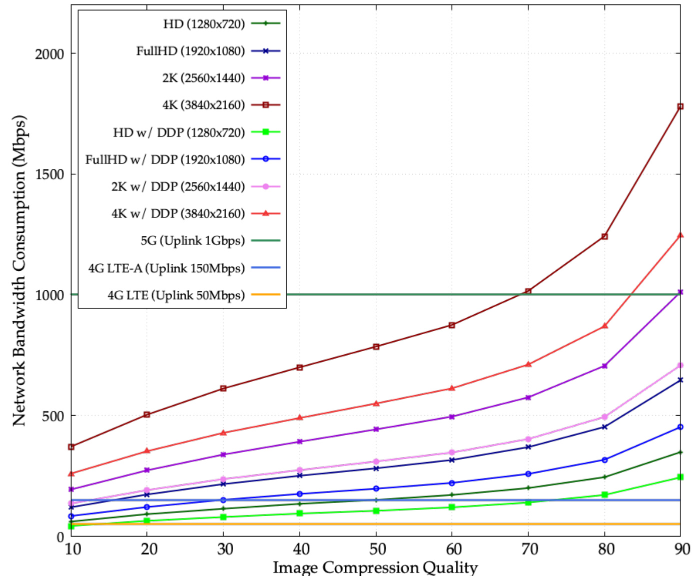

Fig. 5: Bandwidth Consumption vs. Compression Quality

In Fig.6, another experiment is performed to estimate the number of drones supported during a disaster relief mission. In this experiment, image compression quality of 30 is chosen based on the results that we gathered in Fig.5. This specific number is chosen as this is the maximum image compression quality setting that can be supported by the 4G LTE-A network to perform image frame publishing with the standard data publishing method. Result for 4G LTE network is not included in this figure as it has inadequate bandwidth to support for the publishing of HD image frames. No matter if the publishing is with or without the DDP system. In the figure, we can see that in the case of HD image frames publishing under 4G LTE-A with standard publishing method, only one drone stream will be supported. Switching to 5G, the number of supported drone streams increase dramatically. With standard publishing method, up to 9 drone streams will be supported. While in the case of adding DDP, up to 13 drone streams will be supported. This behavior is also reflected across other resolutions. In general, by using DDP, on average we can expect to support for 30% more drone streams.

E. Energy Consumption

To be added once the Energy formula is fixed and cross-checked.

F. Discussion

In test results mentioned in previous sub-section, we perform image frame data publishing to showcase the responsiveness and bandwidth saving that can be achieved by the DDP system. However, one thing to note is that DDP is not only applicable to the publishing of image frames, but also to other

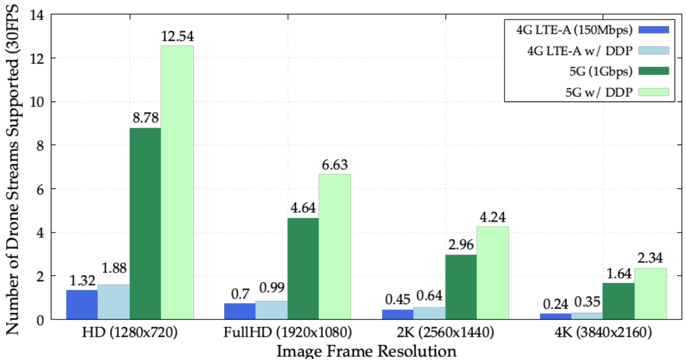

Fig. 6: Expected Number of Drones vs. Image Resolution

data publishing use case. Image frame here can be abstracted to any data frame that is gathered from on-board sensors on a drone or any similar devices and is published continuously or over a period of time during the mission to any receiving end. One example of other data frames used for other use cases are LiDAR data. LiDAR data are varying in sizes and can take up to several MBs, just as in the case of image frame. Other use cases performing mission other than disaster relief, or image frame publishing, can also benefit from DDP system.

V. Conclusion

In this paper, we have proposed a dynamic data publisher system as a solution that reduce the data publishing bandwidth utilization. Experiment results showed that the proposed DDP system is able to perform dynamic data publishing that consumes less bandwidth. This leads to the ability to deploy up to 30% more drones in a mission. Apart from that, since it utilizes past detection results coming from the edge, the dynamic data publisher has a low overhead as it does not add new computation load. Making it a viable addition to a pre-existing system that has data the need to perform data publishing, regardless of the object detection algorithm or to a further extent, regardless of the trigger algorithm used. Remember to add wording for energy saving once formula is ready!

Acknowledgment

This work has been partially funded by the H2020 EU/TW joint action 5G-DIVE (Grant #859881).

References

[1] Akhloufi MA, Couturier A, Castro NA. Unmanned Aerial Vehicles for Wildland Fires: Sensing, Perception, Cooperation and Assistance. Drones. 2021;5(1). Available from: https://www.mdpi.com/2504-446X/5/1/15.

[2] Conceição F, Guimarães C, Cominardi L, Talat ST, Ardiansvah MF, Zhang C, et al. Empowering Industry 4.0 and Autonomous Drone Scouting use cases through 5G-DIVE Solution. In: 2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit); 2021. p. 265–270.

[3] Guimarães C, Groshev M, Cominardi L, Zabala A, Contreras LM, Talat ST, et al. DEEP: A Vertical-Oriented Intelligent and Automated Platform for the Edge and Fog. IEEE Communications Magazine. 2021;59(6):66–72.

[4] Vaddi S, Kumar C, Jannesari A. Efficient Object Detection Model for Real-Time UAV Applications. CoRR. 2019;abs/1906.00786. Available from: http://arxiv.org/abs/1906.00786.

[5] Zhou Y, Rao B, Wang W. UAV Swarm Intelligence: Recent Advances and Future Trends. IEEE Access. 2020;8:183856–183878.

[6] Rahbari D, Alam MM, Moullec YL, Jenihhin M. Fast and Fair Computation Offloading Management in a Swarm of Drones Using a Rating-Based Federated Learning Approach. IEEE Access. 2021;9:113832–113849.

[7] NTT DOCOMO I. White paper: 5G Evolution and 6G. -: NTT DOCOMO; 2020. -.

[8] LODOLO M. LTE Evolution: Standardization & Deployment – The Long Run to 5G. -: Program Manager Mobile Broadband; 2018. -.

[9] Huang S, Li L, Pan Q, Zheng W, Lu Z. Fine-Grained Task Offloading for UAV via MEC-Enabled Networks. In: 2019 IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops); 2019. p. 1–6.

[10] Zhang X, Pal A, Debroy S. EFFECT: Energy-efficient Fog Computing Framework for Real-time Video Processing. In: 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid); 2021. p. 493–503.

[11] Chen J, Chen S, Luo S, Wang Q, Cao B, Li X. An intelligent task offloading algorithm (iTOA) for UAV edge computing network. Digital Communications and Networks. 2020;6(4):433–443. Available from: https://www.sciencedirect.com/science/article/pii/S2352864819303037.

[12] Siemiatkowska B, Stecz W. A Framework for Planning and Execution of Drone Swarm Missions in a Hostile Environment. Sensors. 2021;21(12).

[13] Shi L, Xu S. UAV Path Planning With QoS Constraint in Device-to-Device 5G Networks Using Particle Swarm Optimization. IEEE Access. 2020;8:137884–137896.

[14] Ban TW. An Autonomous Transmission Scheme Using Dueling DQN for D2D Communication Networks. IEEE Transactions on Vehicular Technology. 2020;69(12):16348–16352.

[15] Mukherjee A, Mukherjee P, De D, Dey N. QoS-aware 6G-enabled ultra low latency edge-assisted Internet of Drone Things for real-time stride analysis. Computers & Electrical Engineering. 2021;95:107438. Available from: https://www.sciencedirect.com/science/article/pii/S0045790621003980.

[16] Wang J, Feng Z, Chen Z, George S, Bala M, Pillai P, et al. Bandwidth-Efficient Live Video Analytics for Drones Via Edge Computing. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC); 2018. p. 159–173.

[17] Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv:180402767. 2018.

[18] Bochkovskiy A, Wang CY, Liao HYM. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:200410934. 2020.

[19] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. Ssd: Single shot multibox detector. In: European conference on computer vision. Springer; 2016. p. 21–37.

[20] Pramanik A, Pal SK, Maiti J, Mitra P. Granulated RCNN and multi-class deep sort for multi-object detection and tracking. IEEE Transactions on Emerging Topics in Computational Intelligence. 2021.

[21] Munoz O, Pascual-Iserte A, Vidal J. Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. IEEE Transactions on Vehicular Technology. 2014;64(10):4738–4755.

[22] Dinh TQ, Tang J, La QD, Quek TQ. Offloading in mobile edge computing: Task allocation and computational frequency scaling. IEEE Transactions on Communications. 2017;65(8):3571–3584.

[23] Miettinen AP, Nurminen JK. Energy efficiency of mobile clients in cloud computing. HotCloud. 2010;10(4-4):19.

[24] Ye D, Wu M, Tang S, Yu R. Scalable fog computing with service offloading in bus networks. In: 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud). IEEE; 2016. p. 247–251.

[25] Wang J, Jiang C, Zhang K, Hou X, Ren Y, Qian Y. Distributed Q-learning aided heterogeneous network association for energy-efficient IIoT. IEEE Transactions on Industrial Informatics. 2019;16(4):2756–2764.

[26] Hou X, Ren Z, Wang J, Zheng S, Cheng W, Zhang H. Distributed fog computing for latency and reliability guaranteed swarm of drones. IEEE Access. 2020;8:7117–7130.

[27] Goodman NR. Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction). The Annals of mathematical statistics. 1963;34(1):152–177.

[28] Goldsmith A. Wireless Communications; 2005.
[29] Zhang S, Zhang H, Di B, Song L. Cellular UAV-to-X communications: Design and optimization for multi-UAV networks. IEEE Transactions on Wireless Communications. 2019;18(2):1346–1359.
[30] Chen X, Jiao L, Li W, Fu X. Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Transactions on Networking. 2015;24(5):2795–2808.
[31] Hou X, Ren Z, Cheng W, Chen C, Zhang H. Fog based computation offloading for swarm of drones. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE; 2019. p. 1–7.
[32] Lai CC, Wang LC, Han Z. The coverage overlapping problem of serving arbitrary crowds in 3d drone cellular networks. IEEE Transactions on Mobile Computing. 2020.
[33] Chang W, Meng ZT, Liu KC, Wang LC. Energy-Efficient Sleep Strategy for the UBS-Assisted Small-Cell Network. IEEE Transactions on Vehicular Technology. 2021;70(5):5178–5183.
[34] Zhang H, Hanzo L. Federated learning assisted multi-UAV networks. IEEE Transactions on Vehicular Technology. 2020;69(11):14104–14109.
[35] Ardiansyah MF, William T, Abdullaziz OI, Wang LC, Tien PL, Yuang MC. EagleEYE: Aerial edge-enabled disaster relief response system. In: 2020 European Conference on Networks and Communications (EuCNC). IEEE; 2020. p. 321–325.
[36] Samsung 4K Demo- New York City. [Online]. Available:https://youtu.be/TYrX7w1uIq8;.