Centralized Network Model Improvement System Integrated into UAV Swarm

Duy Loc Nguyen*, Van Bach Tran*, Ngoc Phong Dao†,
Dinh Tuan Tran ‡, Joohoo Lee ‡ and Anh Quang Nguyen*

* School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Vietnam
† Faculty of Information Technology, Posts and Telecommunications Institute of Technology, Vietnam
‡ College of Information Science and Engineering, Ritsumeikan University, Kyoto, Japan
E-mail: loc.nd182644@sis.hust.edu.vn, bach.tv182375@sis.hust.edu.vn, phongdn@ptit.edu.vn,
tuan@fc.ritsumei.ac.jp, leejooho@ieee.org, quang.nguyenanh@hust.edu.vn

***Abstract*—In recent years, UAVs have been applied in many fields, both civil and military. Combining individual UAVs into a swarm makes it possible for them to accomplish complex tasks and meet special requirements. With the advantages of a centralized network that is low cost, easy to deploy, and easy to manage, we focus on connecting the child UAVs through the parent UAV to create a central network. However, the centralized network still has some disadvantages that need to be improved: the fault tolerance of the system is still poor, the quality of service is not good (it is difficult to meet important events). To solve the above challenges, we propose centralized network model improvement system with the parent UAV as the central UAV connecting the child UAVs and a backup UAV; combined with Ground Control Station. Based on solutions in nature: backup and prioritization, the proposed system has overcome the above mentioned disadvantages of the centralized network.**

***Index Terms*—Unmanned Ariel Vehicles (UAV), Ground Control Station (GCS), Remote Server, Backup, QoS, Priority Event, Event-Driven Program**

I. Introduction

With the development of science and technology, UAV has been improved to integrate more features, reduce costs, and improve performance. Today, UAVs are very popular and are used in many fields. Such as search and rescue operations [1], instant delivery [2], remote sensing in agriculture [3], real-time forest fire monitoring [4], weather monitoring [5] or in military field: enemy reconnaissance [6], medical logistics [7].

In the natural environment, animals often have the habit of living in groups. The goal is to increase productivity and survivability. Based on the advantages of swarms in nature, it is necessary to combine robots [8] in general and UAVs in particular. Many researches have shown the applications of UAV swarm such as military applications of UAV swarm technology [9]; Evaluation and future direction of UAV swarm communication architecture [10].

There are many UAV swarm systems using distributed network, which can be mentioned: the maintenance and formation method based on distributed control [11], the distributed control scheme of the UAV swarm based on the heterogeneous roles [12]. However, there is very little research on systems using centralized networks for UAV swarm. Distributed networks have the following disadvantages: Complex network management, difficulties in data communication and synchronization, high maintenance deployment costs. With the above problems, the centralized network deployment can be solved: low cost, easy to deploy, easy to manage, control and control. In addition to the above advantages, the centralized network still has disadvantages that need to be improved: The system's fault tolerance is still poor, the quality of service is not good (a lot of data accumulates on the server causing delay, difficult to respond meet important and urgent events).

In this paper, we propose a centralized network model improvement system to control and monitor UAV swarm. The individual UAVs only need to focus on the specific task, while the control and processing tasks are handled by the **Local Server** (central UAV, also known as parent UAV) and the **Remote Server** (Ground Control Station). This improved system solves the limitations of centralized networks based on solutions in the nature. **Backup** is implemented to enhance the system's fault tolerance. When Local Server cannot reconnect to Remote Server, UAV backup automatically connects to Remote Server. At that time, the UAV swarm will communicate with the ground station through the backup UAV, instead of the parent UAV as before. **Prioritization** is used to address service quality problem. The system assigns priority of events to ensure that critical events are handled in a timely manner.

The rest of the paper is organized as follows. Section II discusses the work involved. Proposal for an improved system model is given in section III, followed by implementation in section IV. Then the simulation work of section V. Finally, section VI is the conclusion.

II. Related Work

In [13], the author raised the problem: "Who will control the swarm?" by showing the advantages of the centralized model over the distributed model. With a centralized model, data from all devices is collected in one place. This provides a panoramic view of the world, unlike the distributed model where each device has limited information about the state of the world. This allows the centralized model to better control

Corresponding Author: Anh Quang Nguyen

higher-level tasks such as system-wide situational awareness, decision making, and large-scale traffic planning. In addition, centralized models can take advantage of high-performance server to run powerful machine learning algorithms, allowing the central system to control quickly and accurately. Besides, with the expansion of the system, the centralized model will be easier than the distributed model.

In [14], the author presented the problem of swarm UAV operation in both centralized and distributed networks. Although using a distributed network reduces the operator's workload but it can add complexity to the system. This not only increases the development cost compared to centralized networks, but also makes it difficult for the system to meet the safety level. Another problem with reducing operator workload on a UAV network is that increasing system autonomy may cause the operator to lose focus and ignore the system. The problem is that when the system fails, it is difficult for the operator to control, test and find the fault.

In [15], the author proposed a centralized data-driven communication architecture between the UAV swarm and the ground control station. With this communication architecture, problems have been solved: coordinating multiple UAVs; efficiently sharing bandwidth according to the priority and urgency of data; avoiding useless transmission of the same data by many UAVs. However, the architecture proposed by the author only applies to GCS as the center of the centralized network, not to mention the UAV as the center.

In this similar context, it is consistent with our proposal, which is to propose a centralized network model and improve the disadvantages of this model. This improved system model will be detailed in the next section.

III. Improved System Model

A. System overview

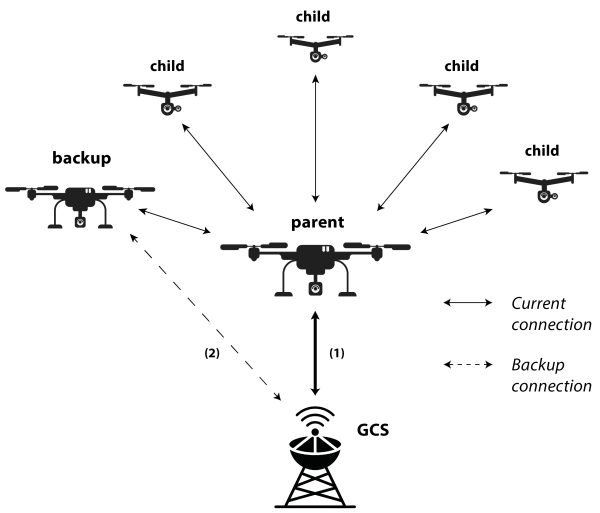

Fig. 1: Centralized network architecture of the system

The centralized network system includes child UAVs connected to the parent UAV and a ground station to monitor and control the UAV swarm when needed. The above connections are all TCP connections. The point of improvement here is adding a backup UAV. The purpose of the backup UAV is to act as an intermediary for data transmission from the UAV swarm to the ground station. When the parent UAV cannot connect to the GCS (connection 1 is broken), the backup UAV connects to the GCS (connection 2 is established).

Unlike the traditional programming system (sequential programming, where the instructions are executed in order from top to bottom), this system uses an event-driven programming. Active objects in the system (parent UAV, child UAV, Remote Server) communicate with each other through events. Events are posted to each object's FIFO queue. One more improvement is that events are assigned a priority and sorted by that priority.

B. Hardware

Below is the block diagram of the hardware into the central UAV. The hardware of the central UAV and the backup UAV are designed the same to serve the backup function of the system.

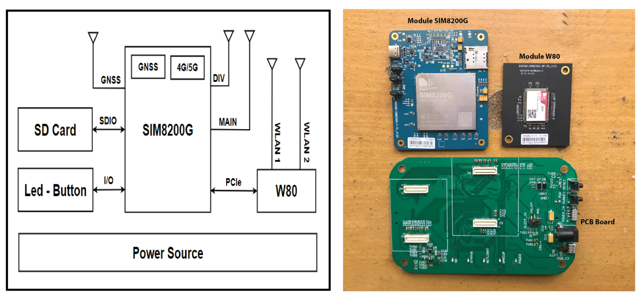

Fig. 2: Block diagram of hardware

1) Processing Module: The SIM8200G is Multi-Band 5G NR/LTE-FDD/LTE-TDD/HSPA+ module solution in a LGA type which supports R15 5g NSA/SA up to 4.0 Gbps data transfer. Support global navigation satellite system: GPS, GLONASS, BEIDOU, GALILEO,... This module is designed to support the proposed system to collect and process information; to communicate with the Ground Control Station.

2) Communication Module: W80 is a small, low-power, Wi-Fi module based on Qualcomm QCA-6391 chipset. The module conforms to IEEE standard 802.11a/b/g/n/ac/ax. The module is integrated with the WLAN processor and 2.4G/5G RF transceivers. It support data communication between SIMCom 5G module SIM8200G through PCIe interface. This module is designed to support the proposed system to create a local area network, connecting UAVs; help the central UAV communicate with other UAVs.

• *Note: Hardware supports WPS function - automatic wifi connection without entering SSID and password. The child UAVs can automatically connect to the parent UAV within a certain range and merge into the swarm.*

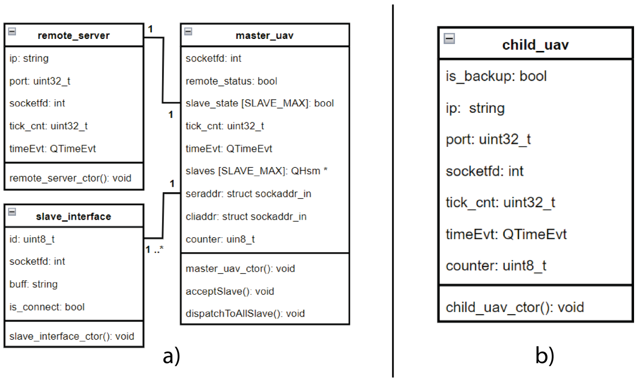

Fig. 3: a) Class Diagram of Parent UAV b) Class Diagram of Child UAV

C. Software

1) Class Diagram: In the parent UAV (Fig. 3a), three class are formed: the remote interface class, the master class and the slave interface class. A master class is linked to a remote interface class for the central UAV to communicate with GCS, control and receive data remotely. A master class is associated with at least one slave interface class. The slave interface class represents the backup UAV or other child UAVs connected to the parent UAV.

In the child UAV (Fig. 3b), a class is formed, which is the child_uav class. This class is designed to support the child UAV to communicate with the parent UAV. The is_backup attribute indicates whether this UAV is a backup UAV.

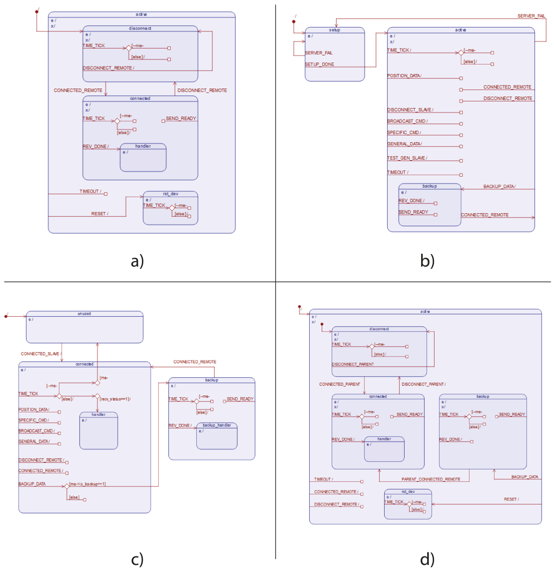

Fig. 4: a) Remote's state machine in parent UAV b) Master's state machine in parent UAV c) State Machine of the Slave Interface in the parent UAV d) State machine in backup UAV

2) State Machine: In Fig. 4a is the machine state of the remote_server class. It includes states: disconnect, connected, handler, reset. Disconnect state represents the state of loss of connection with GCS. When the CONNECTED_REMOTE_SIG event trigger, the system enters the connected state and can communicate remotely. The handler status indicates that the system is processing data that the GCS sends to the central UAV.

In Fig. 4b is the machine state of the master class. When in the setup state, the central UAV sets up and creates a local server so that the child UAVs can connect. Once in the active state, the central UAV can communicate with the child UAVs through events sent to the slave interface class as discussed below.

In Fig. 4c is the machine state of the slave interface class. The unused status indicates that there are no UAV connected to the interface. When there is a connected UAV, switch to connected state. Then, the central UAV can transmit and receive data to the child UAV through this interface class. The handler state is responsible for processing the information sent by the child UAV and forwarding it to the central UAV.

In Fig. 4d is the machine state of the child_uav class running on the hardware of the child UAV and the backup UAV when the backup is not enabled. It includes states: disconnect, connected, handler, reset. The disconnect state represents the disconnection state with the parent UAV. When the CONNECTED_PARENT_SIG event trigger, the system enters the connected state and can communicate with the central. The handler status indicates that the system is processing data coming from the central UAV.

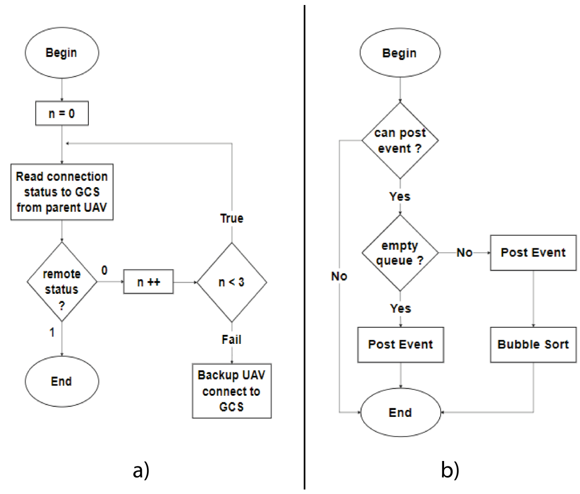

Fig. 5: a) Backup Algorithm b) Priority Algorithm

3) Algorithms: The backup algorithm is implemented on the backup UAV and is shown in Fig. 5a. The parent UAV checks the connection status with the GCS and sends that status information to the backup UAV. When the status information reaches 3 times of disconnection, the backup UAV immediately connects to the GCS and becomes a common data transmission point between the UAV swarm and the ground station.

The priority algorithm is implemented on the parent UAV and is shown in Fig. 5b. When there is a request to post an event to the queue, it is first checked if the event can be posted. Then check for an empty queue. If the queue is empty then post the event and finish. If the queue still has unhandled events, post new events and use the sorting algorithm. That will ensure the event with the highest priority is retrieved the next time.

IV. Implementation

A. QP Framework

QP™/C Real-Time Embedded Framework (RTEF) is a lightweight implementation of the Active Object model of computation specifically tailored for real-time embedded (RTE) systems [16].

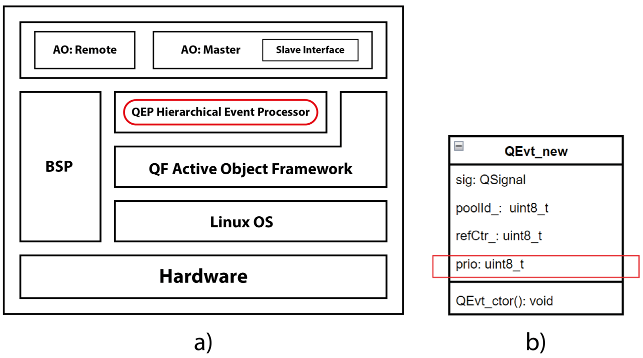

Fig. 6: a) System Software Architecture b) Class Diagram of new QEvt

B. Priority Event

a) FIFO Queue: **QEQueue** is a data structure in QP (Quantum Platform) frame used to describe the ring-type FIFO queue. Queues are managed by attributes: FrontEvt (The pointer in front of the Queue, indicating the state of the Queue, where the event is retrieved and handled), Head (The offset where the next event will be inserted into the Queue), Tail (The offset by which the next event should be passed to FrontEvt).

b) Proposed Event: **QEvt** is a data structure in the QP (Quantum Platform) framework used to represent events in embedded systems. It provides a standard way to represent an event's information and pass it through states and event handlers in the application.

In this paper, the events in the system are assigned with priority levels from 0 to 3 (Table I). The lower the priority value, the higher the priority. Based on the attribute available in QEvt, we have proposed a new event class by adding a prio field (Fig. 6b). This field represents the priority of the event. The priority algorithm is implemented in the QActive_post_(event) function in the qf_actq.c file of the QP/C source code [17].

TABLE I: List of Events by Priority

Critical (0)	High (1)	Medium (2)	Low (3)
SPECIFIC_CMD_SIG	TIME_TICK_SIG	TIMEOUT_SIG	GENERAL_DATA_SIG
RESET_SIG	BROADCAST_CMD_SIG	POSITION_DATA_SIG	MEDIA_DATA_SIG
SEND_DONE_SIG	SEND_READY_SIG	CONNECTED_SLAVE	
REV_DONE_SIG	CONNECTED_REMOTE_SIG	DISCONNECT_SLAVE_SIG	
SETUP_DONE_SIG	DISCONNECT_REMOTE_SIG		
Important control commands, signaling in the system	*common command and common coordination, command from GCS*	*Events from child UAVs*	*Command to transmit general information, image information, video*

C. Backup Mode

In the parent UAV, check the connection status with the GCS. When there is no connection, trigger the DISCONNECT_REMOTE_SIG event, increase the variable counter by 1 and send status information to other UAVs (including backup UAV) via TCP packets . This packet is in JSON format and has a RemoteStatus field that represents the connection status with the GCS. When the counter = 3, there is no connection to the 3rd time, the central UAV will switch to backup mode. Specifically, the parent UAV will pack and send data to the backup UAV so that the backup UAV can be sent to the Remote Server instead of sending it directly as before.

In the backup UAV, receive the status information of the GCS from the central UAV by receiving the above TCP packet. When there is no connection, fire the DISCONNECT_REMOTE_SIG event and increment the counter by 1. When the counter = 3, the backup UAV will switch to backup mode. The backup UAV will connect TCP to the GCS, becoming the intermediate point between the GCS and the UAV swarm and continue to perform the task.

V. Simulation

A. Environment Setup

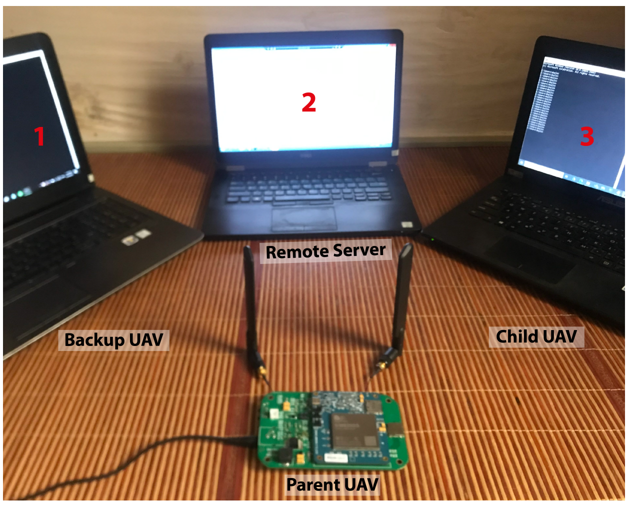

Fig. 7: System simulation setup

Fig. 7 shows the simulation environment setup. Due to hardware limitations, the article uses 3 more laptops running software to represent each object in the system. Laptop 1 represents the backup UAV running the backup algorithm program. Laptop 2 represents the ground station. Laptop 3 runs the program of the child UAVs. Specifically, in laptop 3 running a multi-threaded program, each thread creates a socket to communicate with the parent UAV.

B. Script

1) Backup: The purpose of this simulation is to measure the **interruption time** of the backup process. That is the time from when the system loses connection with GCS for the 3rd time until the backup UAV establishes a new connection with GCS. This part runs a system simulation with 2 parameters: the number of UAVs (including the parent UAV, backup UAV) and position sampling cycle of the UAV swarm. The above parameter affects the bandwidth of the system; data transfer performance between the parent UAV and the backup UAV to perform the backup process.

2) Priority: Randomly post with four different events (event 2, event 5, event 3, event 6) to the queue. The purpose of the simulation is to measure the **response time** of the event. This response time is the time from when the event starts being posted to the queue until the event is removed from the queue for processing. Simulation is performed in 2 cases. Case 1 is Normal simulation (Not implemented the priority algorithm in the queue). Case 2 is Priority simulation (Implemented the priority algorithm in the queue). In this case, each event is assigned a different priority.

C. Result

TABLE II: Results of measuring interruption time (ms)

		Position Sampling Cycle									
		0.01s	0.02s	0.03s	0.04s	0.05s	0.07s	0.1s	0.2s	0.5s	1.0s
Number of UAVs	05	-	33	38	30	41	42	37	48	50	41
	10	-	-	-	47	63	66	50	63	43	43
	20	-	-	-	-	-	60	56	42	36	34
	30	-	-	-	-	-	-	-	-	52	62

1) Backup: Looking at Table II, we see two things:

- The first is that there are some cases where the interrupt time is unknown. This means that the backup UAV cannot switch to backup mode. The reason is that the packet lost connection to GCS from the parent UAV cannot be transmitted to the backup UAV. This error occurs more when the number of UAVs increases. As the number of UAVs increases, the bandwidth of each connection will decrease. Consider the case of 20 UAVs: when the position sampling cycle for transmission to the parent UAV of each UAV is reduced to 0.05s, the interrupt time is unknown, backup does not occur. To make sure backup happens, the position sampling cycle is increased, as shown in Table II. As such, the backup process occurs depends on the number of UAVs and the position sampling cycle.
- The second is that once the backup has occurred, the interrupt time does not depend on the number of UAVs or the position sampling cycle but it depends on the speed of establishing connection of backup UAV with GCS.

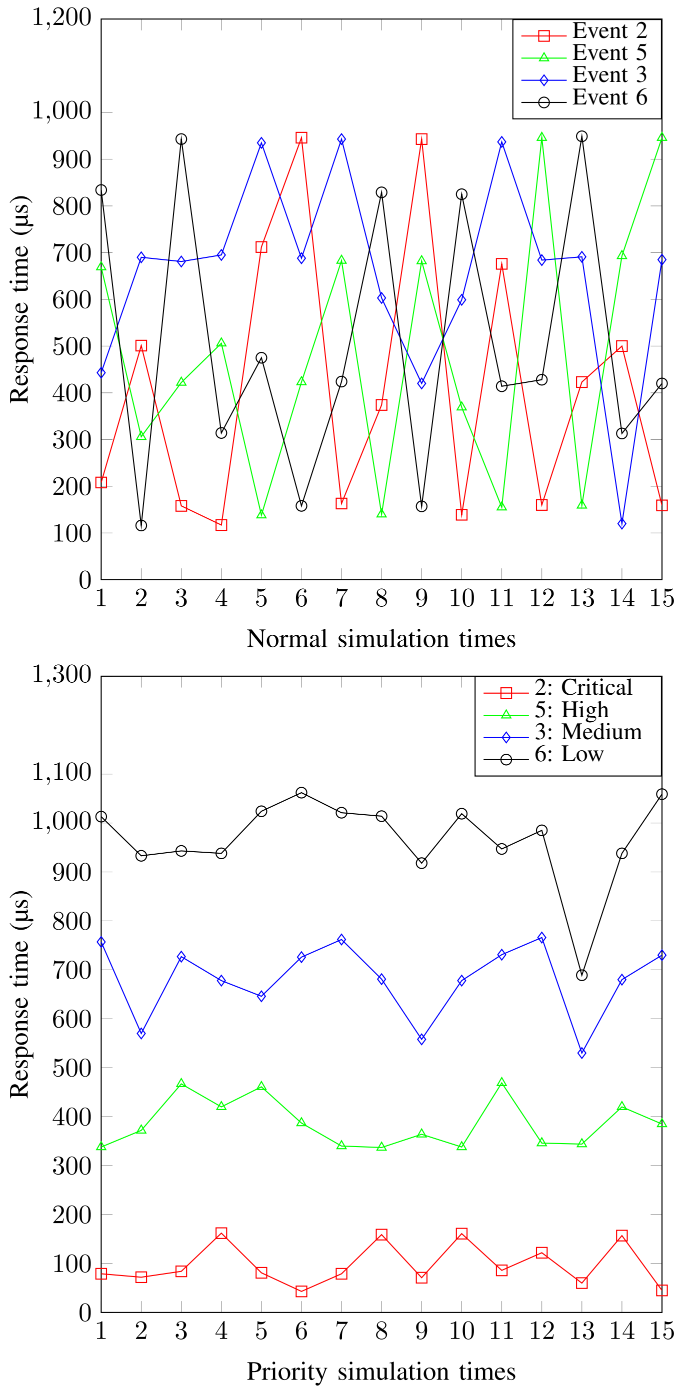

Fig. 8: Simulation results of response time with Normal simulation (top) and Priority simulation (bottom)

2) Priority:

- In Fig. 8 (top), each event is the same, the response time of the 4 events varies with the order the events are posted

in each simulation times. The first posted event has the lowest response time and the last posted event has the highest response time.
- In Fig. 8 (bottom), the response time of 4 events varies by priority and not by the order in which events are posted in each simulation times. Specifically, the critical priority event has the fastest response time, followed by the high priority event, and finally the low priority event. We also see, the average response time of critical event is lower in the priority simulation case than in the normal simulation case.

VI. Conclusion and Future Works

In the paper, we have presented the centralized network improvement system integrated in the UAV swarm. Based on the solution in nature, the system has improved the disadvantages of the centralized network.

The proposed improvement system has solved 2 problems: fault tolerance and quality of service. First, the system will automatically switch to backup mode when the central UAV loses connection with the Ground Control Station. Then the swarm will communicate with GCS through UAV Backup. Second, the system handles events according to the priority of that event, ensuring that important events are handled in a timely manner.

For future work, in priority contribution, we will use an optimal sorting algorithm to further improve event response times. In backup contribution, we will improve the data transmission efficiency between the parent UAV and the backup UAV as the number of UAVs increases and the location sampling period decreases. In addition, we will expand the network with more UAVs and control the system in a realistic flight environment.

Acknowledgment

This research is funded by the Vietnam Ministry of Science and Technology (MoST) under project number NDT/AT/22/09.

References

[1] Mario Silvagni, Andrea Tonoli, Enrico Zenerino and Marcello Chiaberge. Multipurpose UAV for Search and Rescue Operations in Mountain Avalanche Events. Geomatics, Natural Hazards and Risk, 2017.

[2] Qiuchen Gu, Tijun Fan, Fei Pan, Chong Zhang. A vehicle-UAV operation scheme for instant delivery. Computers & Industrial Engineering, vol 149, November 2020, pages 18-33.

[3] Lei Deng, Zhihui Mao, Xiaojuan Li, Zhuowei Hu, Fuzhou Duan, Yanan Yan. UAV-based multispectral remote sensing for precision agriculture: A comparison between different cameras. ISPRS Journal of Photogrammetry and Remote Sensing, vol 146, December 2018, pages 124-136.

[4] Eni Dwi Wardihani, Magfur Ramdhani, Amin Suharjono, T.A. Setyawan, Sidiq Syamsul Hidayat, Helmy, Sarono Widodo, Eddy Triyono, F.Saifullah. Real-time forest fire monitoring system using unmanned aerial vehicle, 2018, vol. 13, no 6, pages 1587-1594.

[5] J. Cecil, Ph.D. A conceptual framework for supporting UAV based cyber physical weather monitoring activities — IEEE Conference Publication — IEEE Xplore. April 2018

[6] Online Available: https://ieeexplore.ieee.org/abstract/document/6761569/ (accessed on May 2,2023)

[7] Daniel T. Lammers, James M. Williams, Jeffrey R. Conner, Emily Baird, Omar Rokayak, John M. McClellan, Jason R. Bingham, Richard Betzold, Matthew J. Eckert. Airborne! UAV delivery of blood products and medical logistics for combat zones - Lammers - 2023 - Transfusion - Wiley Online Library, 27 March 2023.

[8] Haibin Duan, Mengzhen Huo, Yanming Fan. From animal collective behaviors to swarm robotic cooperation. February 2023.

[9] Zhu Xiaoning. Analysis of military application of UAV swarm technology — IEEE Conference Publication — IEEE Xplore.

[10] Mitch Campion, Prakash Ranganathan, and Saleh Faruque. A Review and Future Directions of UAV Swarm Communication Architectures. 2018 IEEE International Conference on Electro/Information Technology

[11] Xiaowei Fu, Jing Pan, Haixiang Wang, Xiaoguang Gao. A formation maintenance and reconstruction method of UAV swarm based on distributed control. Aerospace Science and Technology, September 2020.

[12] Jiang ZHAO, Jiaming SUN, Zhihao CAI, Yingxun WANG, Kun WU. Distributed coordinated control scheme of UAV swarm based on heterogeneous roles. Chinese Journal of Aeronautics, vol 35, Issue 1, January 2022, pages 81-97.

[13] Andrew Myers. Who will control the swarm? — Stanford University School of Engineering, June 19, 2017.

[14] Missy Cummings. Operator Interaction with Centralized Versus Decentralized UAV Architectures, January 2015.

[15] Amira Chriki, Haifa Touati, Hichem Snoussi, Farouk Kamoun. UAV-GCS Centralized Data-Oriented Communication Architecture for Crowd Surveillance Applications.

[16] Online Available: https://www.state-machine.com/qpc/index.html (accessed on May 2,2023)

[17] Online Available: https://www.state-machine.com/qpc/struct_q_active.html#a1a81b9fd06d9c0aa5dea32d194f0552b (accessed on May 2,2023)