

A Novel Swarm Unmanned Aerial Vehicle System: Incorporating Autonomous Flight, Real-Time Object Detection, and Coordinated Intelligence for Enhanced Performance



Murat Bakirci 

Unmanned/Intelligent Systems Lab, Faculty of Aeronautics and Astronautics, Tarsus University, Mersin 33400, Turkey

Corresponding Author Email: muratbakirci@tarsus.edu.tr

<https://doi.org/10.18280/ts.400524>

ABSTRACT

Received: 24 March 2023

Revised: 24 July 2023

Accepted: 12 September 2023

Available online: 30 October 2023

Keywords:

swarm system, unmanned aerial vehicle, communication, network topology, routing optimization

Presently, swarm Unmanned Aerial Vehicle (UAV) systems confront an array of obstacles and constraints that detrimentally affect their efficiency and mission performance. These include restrictions on communication range, which impede operations across extensive terrains or remote locations; inadequate processing capabilities for intricate tasks such as real-time object detection or advanced data analytics; network congestion due to a large number of UAVs, resulting in delayed data exchange and potential communication failures; and power management inefficiencies reducing flight duration and overall mission endurance. Addressing these issues is paramount for the successful implementation and operation of swarm UAV systems across various real-world applications. This paper proposes a novel system designed to surmount these challenges through salient features such as fortified communication, collaborative hardware integration, task distribution, optimized network topology, and efficient routing protocols. Cost-effectiveness was prioritized in selecting the most accessible equipment satisfying minimum requirements, identified through comprehensive literature and market review. By focusing on energy efficiency and high performance, successful cooperation was facilitated through harmonized equipment and effective task division. The proposed system utilizes Raspberry Pi and Jetson Nano for task division, endowing the UAVs with superior intelligence for navigating intricate environments, real-time object detection, and the execution of coordinated actions. The incorporation of the Ad Hoc UAV Network's decentralized approach enables system adaptability and expansion in response to evolving environments and mission demands. An efficient routing protocol was selected for the system, minimizing unnecessary broadcasting and reducing network congestion, thereby ensuring extended flight durations and enhanced mission capabilities for UAVs with limited battery capacity. Through the careful selection and testing of hardware and software components, the proposed swarm UAV system improves communication range, processing power, autonomy, scalability, and energy efficiency. This makes it highly adaptable and effective for a broad spectrum of real-world applications. The proposed system sets a new standard in the field, demonstrating how the integration of intelligent hardware, optimized task division, and efficient networking can overcome the limitations of current swarm UAV systems.

1. INTRODUCTION

The advent of Unmanned Aerial Vehicles (UAVs) has been a transformative force in various fields, facilitated by the decreasing costs of hardware and the integration of powerful embedded computers [1]. These autonomous flying platforms, laden with diverse sensors and equipped with robust processing capabilities, are deployed with ease and possess high mobility, making them instrumental in tasks such as aerial surveillance [2], delivery services [3], and search and rescue missions [4]. Lately, the potential of UAVs to augment surveillance systems in both public and private sectors has been recognized. However, the execution of mobile object tracking [5, 6] often surpasses the capability of a single UAV, necessitating the exploration of UAV swarms.

Compared to individual UAVs, UAV swarms offer an expanded potential, demonstrated by increased mission efficiency, enhanced processing capacity, greater flexibility, redundancy, and improved reliability. The ability of UAV

swarms to execute tasks such as reconnaissance, surveillance, and mapping of unexplored environments is attributed to the inter-communication capabilities of their intelligent control algorithms [7, 8].

The concept of UAV swarm systems revolves around the coordinated operation of multiple UAVs to achieve a common objective. This collective effort leverages collective intelligence, distributed sensing, and collaborative decision-making, allowing the swarm to accomplish tasks beyond the scope of a single UAV. Various configurations, such as grid, circular pattern, or adaptive formations, can be realized using formation control algorithms [9]. An efficient coordination of actions within the swarm is achieved through the use of coordination algorithms [10], while task allocation based on individual UAV capabilities and mission requirements ensures optimal resource utilization [11]. Additionally, the distributed sensing capabilities of UAV swarm systems enable comprehensive coverage of large areas [12].

With their inherent ability to increase efficiency, scalability,

and versatility, UAV swarm systems have found potential applications in border surveillance [13], critical infrastructure protection [14], and monitoring of large events [15]. In search and rescue operations, they can cover extensive areas, track targets, and provide real-time situational awareness [16]. UAV swarms equipped with sensors can monitor environmental parameters, such as air quality, water quality, or wildlife habitats [17], and can assist in precision agriculture by performing tasks like crop monitoring and spraying [18]. In areas with limited network connectivity, they can act as flying communication relays.

Despite these advantages, the high costs associated with professional-grade UAVs pose a challenge to the development of swarm UAV technology [19]. Affordable UAVs, although cost-effective, come with certain limitations such as reduced payload weight, flight time, communication range, and onboard processing power. To overcome these limitations, innovative software and hardware solutions are required to achieve efficient UAV swarm behavior. One such solution is to reduce the total number of UAVs by developing an efficient network architecture among swarm members [20].

In the central communication architecture, all communication in a system is routed through a central hub or node. This central entity serves as the focal point for transmitting and receiving data, coordinating communication between different devices or subsystems. In a decentralized architecture, communication and decision-making authority are distributed among multiple nodes or entities in the system. There is no central hub or single point of control. Instead, each node in the system operates independently and communicates directly with other nodes as needed. Each of these three decentralized architectures is distinguished by using backbone UAVs to communicate with the control station over longer distances, typically carrying higher quality transceivers and acting as a gateway for the swarm system. In addition to the network architecture, a routing protocol is a crucial component needed to determine how messages travel through the network in UAV swarms. In UAV ad hoc networks [21], UAVs communicate and collaborate without relying on a pre-existing infrastructure. Noor et al. [22] discusses several routing techniques that can be employed to establish and maintain communication between UAVs are discussed. The study, which focuses on routing messages with mesh networks, presents two perspectives: routing and flooding.

There are also various routing-based protocols considered for Flying Ad Hoc Network (FANET) purposes [23]. For example, Akhtar et al. [24] provides a comprehensive survey of various routing protocols compatible for UAV swarm applications. The study evaluates the performances of different proactive routing protocols under real-world scenarios. It focuses on five routing protocols: topology-based routing [25], location-based routing [26], clustering/hierarchical routing [27], swarm-based routing [28], and latency-tolerant network routing [29]. Topology-based routing protocols utilize information about the network's topology to make routing decisions. These protocols consider the network's physical or logical structure, such as the arrangement of nodes and the connectivity between them, to determine the best paths for data transmission. Location-based routing protocols, on the other hand, utilize the geographic location information of network nodes to make routing decisions. They take advantage of the physical location coordinates of nodes to determine the best paths for data transmission. Clustering or hierarchical routing protocols

organize network nodes into clusters or hierarchies to improve routing efficiency, scalability, and management in large-scale networks. These protocols divide the network into smaller groups or layers, where nodes within each cluster or hierarchy can communicate with each other more efficiently. Swarm-based routing protocols inspired by swarm intelligence, where multiple entities (nodes) in a network emulate the collective behavior of natural swarms or colonies to make routing decisions. These protocols aim to achieve self-organization, adaptability, and robustness in dynamic and distributed network environments [30]. Finally, latency-tolerant network routing protocols, also known as store-and-forward protocols, are designed to handle scenarios where nodes in a network may become temporarily disconnected or experience high latency. These protocols allow nodes to store outgoing packets in a buffer or queue until they are able to reconnect with another node or find a suitable path for forwarding the packets. Among them, topology-based routing protocols are the most popular, as there are numerous possible implementations and numerous articles comparing their performance for various applications.

The ground control of the swarm UAV system, given the selected network structure, is another critical facet. The work presented in study of Wang et al. [31] addresses the topic of mission planning for UAV swarms using QGroundControl, an open-source ground control station software used extensively for controlling UAVs. This approach automates the mission planning process for UAV swarms based on swarm configuration and flight plans, thereby reducing manual effort and complexity involved in coordinating multiple UAVs. Practical applications of UAV technology, equipped with object detection capabilities across diverse fields including environmental monitoring and disaster management, are demonstrated in study of Hossain and Deok-jin [32].

Another critical issue with considerable implications for network reliability in swarm UAV systems is the impact of mobility patterns. The mobility of individual UAVs can significantly affect the network's overall performance, communication efficiency, and reliability. The Distributed Pheromone Spray Mobility Model (DPSMM) and the Semi-Random Circular Motion Model (SRCM) are two preferred mobility models in large-area exploration and observation applications, as described in studies [33, 34], respectively.

The proposal of a general framework for detecting and tracking objects using UAVs with unusual hardware configurations is discussed in study of Hossain and Lee [35]. This framework allows for the integration of various object detection algorithms and accommodates UAVs with non-standard or unique hardware configurations, thereby expanding the possibilities for UAV-based object detection and tracking tasks.

However, a comprehensive framework that integrates the discussed aspects is conspicuously absent in existing research on UAV swarm systems. The majority of studies focus on modeling and simulation aspects without considering the financial feasibility of constructing the proposed UAV systems in reality. This research aims to address these gaps by proposing a cost-effective UAV that can form swarms, perform autonomous flight, and achieve real-time object detection using swarm intelligence. An effective communication network solution is also provided, offering sufficient range and bandwidth for coordinating swarms, aggregating live video feeds, and collecting real-time object detection data. The system's design aspects are elucidated in

detail, providing professionals with the required information to develop and tailor the platform for specific use cases. This study serves as an introduction to specialized UAV configurations and swarm operations, addressing the demand for intelligent individual or swarm UAVs with a low-cost solution.

2. SYSTEM OVERVIEW

2.1 Real-time ground monitoring

A UAV ground control station (GCS) is a system or software used to control, monitor, and communicate with UAVs. GCS serves as the interface between the operator or pilot and the UAV, allowing for real-time control, mission planning, data collection, and telemetry monitoring. Some popular GCS software can be listed as Mission Planner, QGroundControl and MAVProxy. QGroundControl (QGC) is an open-source GCS software used for controlling and monitoring UAVs. QGC provides a user-friendly interface that allows planning missions, controlling UAVs, monitoring flight parameters and analyzing telemetry data. It also supports various UAV platforms and is compatible with different autopilot systems, including ArduPilot and PX4. QGC facilitates the planning of complex missions by defining waypoints, survey patterns and other mission parameters. The software provides a map interface where users can interactively select locations, set altitude and speed, and configure mission-specific actions. QGC also provides active control of UAVs during flight missions, thus, operators can switch between different flight modes, adjust flight parameters, and monitor real-time telemetry data. The software displays real-time telemetry data from the UAV, including GPS position, altitude, speed, battery status, and sensor readings. This information helps to monitor the UAV's performance and make informed decisions during the flight mission. More importantly, QGC supports camera control for capturing images or videos during a mission. The UAV's camera parameters such as shutter speed, exposure and focus are configurable, which is an important feature for object detection applications. QGC provides a convenient interface for configuring and adjusting the parameters of the UAV's flight controller. It also offers tools for analyzing and visualizing flight data. Moreover, the software includes a simulation mode that allows simulating UAV missions without actually flying the vehicle. This feature helps in testing and verifying mission plans, ensuring the feasibility and safety of the proposed operations. The software is available for various platforms, including Windows, macOS, Linux, Android, and iOS, making it accessible to a wide range of users.

The other software, Mission Planner, has some limitations and drawbacks that can negatively affect the design. The complexity of the user interface may require a learning curve and some effort to become proficient in using all available features. It is primarily designed for Windows operating systems, which can limit its availability for users on other platforms such as macOS or Linux. Moreover, it does not have dedicated mobile apps for iOS or Android devices. While it is possible to access Mission Planner via web browsers on mobile devices, the user experience may not be as optimized or feature-rich compared to the desktop version. This limitation may restrict the mobility and convenience of using

Mission Planner in the field. Mission Planner also relies on external tools, such as .NET Framework and drivers for communication with specific hardware, which may require additional installations and configurations. This is another factor that increases the cost of the entire system. MAVProxy, on the other hand, operates primarily through a command-line interface, which may not be as user-friendly or intuitive as graphical user interfaces offered by other two GCS software. Unlike QGC, MAVProxy does not provide built-in graphical visualizations of flight data, such as maps, charts, or 3D models, which is a critical drawback. Besides these, it has other disadvantages such as having a steeper learning curve, lack of user-friendly configuration, and limited cross-platform support. Ultimately, considering the advantages of QGC and the disadvantages of the other two software, QGC was preferred as the GCS for the UAV systems in this study.

2.2 Object recognition model

Object recognition is a computer vision process that encompasses the identification and localization of objects in images or videos. The main aim of object detection algorithms is to recognize the presence of objects in an image, categorize them into predefined classes, and furnish their respective bounding box coordinates or segmentation masks. There are two primary categories of object detection models: two-stage and single-shot detectors. In two-stage detectors, the process occurs in two steps: region proposal and object classification and localization. During the first step, the algorithm generates a set of potential object regions or proposals within the image. Various techniques can be used for region proposal, such as selective search, region proposal networks (RPNs), or anchor-based methods. These methods identify regions that are likely to contain objects based on their visual characteristics, such as edges, textures, or saliency. In the second step, the algorithm classifies the proposed regions into specific object categories and refines their bounding box coordinates or segmentation masks. This step involves feature extraction from the proposed regions using convolutional neural networks (CNNs) or other deep learning architectures. The extracted features are then fed into a classification module to determine the object category, while regression modules refine the bounding box coordinates or generate pixel-level masks. Some two-stage object detection algorithms commonly used in various applications are Region-based CNN (R-CNN), Faster R-CNN, Mask R-CNN, Cascade R-CNN, and Two Stage Detector (TSD). Faster R-CNN, one of the leading algorithms in this category, introduced the concept of region proposal networks (RPNs) to generate potential object proposals. It uses a CNN to extract features from the input image and then passes these features through an RPN to propose candidate regions. The proposed regions are subsequently classified and refined to generate the final object detection results. R-CNN, on the other hand, is an earlier version of Faster R-CNN and operates in a similar two-stage manner. It uses selective search or another region proposal technique to generate potential object proposals. These proposals are then individually classified and refined using CNNs. Two-phase object detection algorithms generally offer high accuracy, but can be computationally intensive compared to single-phase algorithms. They are commonly used in applications that require precise object localization and accurate classification, such as in advanced surveillance systems, autonomous driving, and detailed visual analysis. The two-stage design allows for a more thorough evaluation

of potential object proposals, leading to improved detection performance.

Single-shot detectors, on the other hand, bypass the region suggestion, instead processing the input image as a grid, and each cell is processed to determine whether an object of interest is present in the cell. Some commonly used single-phase object detection algorithms are You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD), RetinaNet, EfficientDet, CenterNet, and CornerNet. These single-phase object detection algorithms have been widely adopted due to their ability to provide real-time or near-real-time performance while maintaining competitive accuracy. They have applications in various domains, including autonomous driving, surveillance, robotics, and image analysis. Each algorithm has its unique characteristics and trade-offs in terms of accuracy, speed, and computational requirements. YOLO stands out as one of the most widely used single-phase object detection algorithms, known for its real-time performance. The algorithm partitions the input image into a grid and then predicts bounding boxes and class probabilities for each grid cell. YOLO has multiple versions, with YOLOv3, YOLOv4, and YOLOv5 being widely adopted. SSD, a widely used single-phase object detection algorithm, functions across various scales. It uses a set of predefined anchor boxes at different aspect ratios and scales to predict object locations and class probabilities. SSD achieves a trade-off between accuracy and speed.

The decision between single-shot and two-shot detectors requires striking a balance between speed and accuracy. Single-shot detectors are generally faster compared to two-shot detectors. They perform object detection in a single pass, making them well-suited for real-time applications where low latency is critical. Single-shot detectors achieve high inference speeds by simplifying the detection process and avoiding the need for explicit region proposal steps. Two-shot detectors tend to offer higher accuracy compared to single-shot detectors. They usually have better localization precision and can handle objects at various scales more effectively. Two-shot detectors often excel in precise object localization. By leveraging region proposals and refined bounding box regression, they can better capture object boundaries and provide accurate bounding box coordinates. Single-shot detectors may have slightly lower localization accuracy, especially for small or densely packed objects. Single-shot detectors often require less training time compared to two-shot detectors. They have a simpler architecture and fewer computational steps, resulting in faster training convergence. Two-shot detectors, on the other hand, involve two stages of training, which can be more computationally intensive and time-consuming. While this study does not delve into the selection and implementation of a machine learning model, the proposed hardware is designed to offer sufficient flexibility for investigating various options. This paper defines real-time object detection as the capability of processing images or videos with a satisfactory inference rate directly on the onboard hardware of UAVs.

The acceptable inference rate for a UAV object detection application can vary depending on the specific use case, mission requirements, and hardware capabilities. There are some general benchmarks and performance metrics reported in the literature that can guide the determination of an acceptable inference rate for UAV object detection applications. The first metric is the Frames Per Second (FPS) that measures how many frames the object detection algorithm

can process per second. For UAV applications like the one in this study, real-time or near-real-time performance is often desired to enable prompt decision-making and response. Acceptable inference rates can range from 15 FPS to 30 FPS or higher, depending on the application's needs. Latency is the time delay between capturing an image by the UAV's camera and obtaining the object detection results. Low latency is crucial for applications requiring quick reaction times, such as collision avoidance or dynamic tracking. Acceptable latency can range from a few milliseconds to a few hundred milliseconds. Accuracy is a critical metric for object detection applications. The acceptable accuracy level may vary based on the application's safety and reliability requirements. Common accuracy metrics include mAP (Mean Average Precision), IoU (Intersection over Union), and precision-recall curves. The object detection range, on the other hand, refers to the maximum distance from the UAV at which objects can be reliably detected. The acceptable range will depend on the UAV's altitude, camera specifications, and the application's detection requirements. For UAVs with limited onboard computing power, offloading some processing to a GCS or cloud server may be necessary. The acceptable inference rate should consider the data transmission capabilities and processing capabilities of both the UAV and the GCS. Moreover, if the UAV requires real-time object tracking, the acceptable inference rate should be sufficient to handle continuous tracking while maintaining accuracy. Conclusively, it is worth noting that the acceptable inference rate may vary across different UAV applications. For instance, in search and rescue missions or surveillance, which is the primary objective of the system proposed in this study, real-time performance and accuracy are of utmost importance. On the other hand, in environmental monitoring applications, slightly lower inference rates may be considered acceptable, provided the UAV can cover larger areas with longer flight times.

2.3 Transport layer protocol

Two transport layer protocols, transmission control protocol (TCP) or user datagram protocol (UDP), can be used to stream video from UAVs to the GCS over the network. While TCP is primarily designed for reliable data transmission, it can also be utilized for streaming applications with certain considerations. The UAV and the GCS first establish a TCP connection using a three-way handshake process. This ensures a reliable and ordered communication channel between the two devices. The video captured by the UAV's camera needs to be encoded into a format suitable for transmission. Common video encoding formats used in streaming applications include H.264, H.265 (HEVC), or VP9. The video encoding process compresses the video data, reducing its size while maintaining acceptable quality. The encoded video is divided into packets, which are then encapsulated within TCP segments. Each packet contains a portion of the video stream along with associated metadata. The UAV starts sending the TCP segments containing the video packets to the GCS over the established TCP connection. The packets are transmitted in sequential order, ensuring that the GCS can reconstruct the video stream correctly. The GCS receives the TCP segments containing the video packets. As TCP ensures reliable data delivery, the GCS can expect to receive all packets without loss or corruption. The GCS then reassembles the packets in the correct order to reconstruct the video stream. To ensure smooth video playback and handle variations in network conditions, the GCS typically employs

buffering mechanisms. The received video packets are stored in a buffer, allowing the GCS to maintain a steady video playback rate. Buffering helps handle network delays or fluctuations, preventing interruptions in the video stream.

Unlike TCP, UDP provides a connectionless, unreliable, and lightweight communication mechanism for data transmission between devices over an IP network. Each UDP datagram is treated independently and can be sent to the destination without prior handshaking. It does not provide built-in mechanisms for guaranteed delivery or error correction. Once a UDP datagram is sent, it is up to the receiving device to handle any packet loss or errors. There is no acknowledgment or retransmission mechanism in UDP. Additionally, it has minimal protocol overhead compared to TCP, making it more efficient in terms of bandwidth usage and processing resources. The absence of features like flow control and congestion control in UDP reduces complexity and allows for faster data transmission. It is often used for applications that prioritize low latency and efficiency over reliability. In a swarm system, it is possible to configure each UAV as a server controlled by a GCS. This setup allows for centralized control and coordination of multiple UAVs from a single GCS. By configuring each UAV as a server controlled by a GCS, the swarm system benefits from centralized control, efficient communication, and synchronized operations. This configuration enables coordinated mission execution, real-time monitoring, and flexible command and control capabilities across the swarm of UAVs.

3. ARCHITECTURAL DESIGN FOR THE SWARM UAV SYSTEM

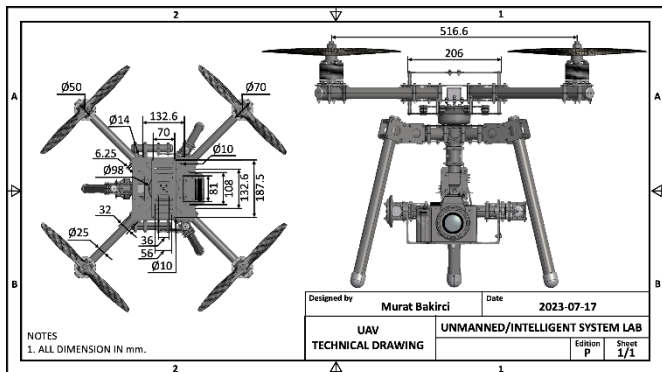


Figure 1. CAD illustration of an individual member of the proposed swarm UAV system

An essential focus of this work centered on achieving a cost-effective implementation of the swarm UAV system. This entailed significant emphasis on meticulously designing and deploying the equipment and swarm network for each UAV, as depicted in Figure 1 through a CAD illustration. By prioritizing cost-effectiveness, it is aimed to optimize resource utilization while preserving the desired performance of the swarm system. Consequently, thoughtful selection of cost-effective UAV equipment that aligns with the application's requirements becomes crucial. This includes choosing UAV platforms, sensors, communication modules, and other necessary components that strike a balance between cost and performance. Designing the UAV system with scalability and modularity in mind allows for cost-effective expansion and maintenance. Each UAV should be designed to be easily

integrated into the swarm network, allowing for seamless deployment and scalability as the system grows. Developing an efficient and cost-effective communication network among the UAVs is essential. This includes selecting appropriate communication protocols, optimizing data transmission methods, and considering the use of low-cost communication technologies while ensuring reliable and secure data exchange. Moreover, employing standardized protocols and interfaces enables interoperability between different UAVs and components. This promotes compatibility, simplifies integration, and allows for the use of cost-effective off-the-shelf solutions, reducing development and maintenance costs.

3.1 Network topology configuration

In object detection applications, the network plays a crucial role in transmitting videos captured by cameras integrated into UAVs back to the GCS for operators to analyze. This enables real-time monitoring, analysis, and decision-making based on the video feed. The video captured by the cameras on the UAVs needs to be encoded into a suitable format for efficient transmission over the network. Commonly used video compression standards such as H.264 or H.265 (also known as AVC or HEVC) can be employed to reduce the video file size while preserving acceptable image quality. The video data needs to be transmitted wirelessly from the UAVs to the GCS. This can be achieved using various wireless communication technologies, such as Wi-Fi, cellular networks (4G/5G), or dedicated UAV communication systems. Video transmission typically requires a significant amount of bandwidth, especially if high-resolution videos are being streamed. Effective bandwidth management techniques, such as adaptive streaming or quality-of-service (QoS) mechanisms, can be implemented to optimize video transmission while ensuring efficient network utilization. In object detection applications, real-time video transmission is often crucial to enable timely analysis and response. Minimizing latency in the network becomes important to maintain a near-real-time video feed. This can be achieved through optimized network configurations, low-latency communication protocols, and prioritizing video traffic over other data.

To maximize the swarm footprint in an outdoor environment, the routing protocol used in the UAV swarm system should provide a reasonable outdoor range. This range determines the effective communication distance between UAVs within the swarm and influences the overall coverage area and scalability of the system. Since the aim of the study is to minimize the cost of the swarm, it is reasonable to focus on unlicensed spectrum technologies. Unlicensed spectrum technologies provide cost-effective communication options without the need for acquiring expensive licenses. Some commonly used unlicensed spectrum technologies for UAV swarm communication are Wi-Fi, bluetooth, Zigbee and LoRa. Wi-Fi (IEEE 802.11) operates in unlicensed frequency bands and offers reasonable data rates and coverage. It is widely available, cost-effective, and supports a variety of communication needs within a limited range. Bluetooth provides short-range communication suitable for close-range coordination and control within the swarm. It is a low-power, low-cost option commonly used for device-to-device communication. Zigbee (IEEE 802.15.4) is a low-power wireless communication standard designed for low-cost and low-data-rate applications. It operates in unlicensed bands and is well-suited for short-range communication between nearby

UAVs or within a localized swarm. LoRa (Long Range), on the other hand, is a long-range, low-power wireless communication technology that operates in unlicensed frequency bands. It enables communication over several kilometers, making it suitable for extending the range of UAV swarm communication without requiring expensive licensed spectrum.

The 802.11n IEEE standard, with its data rates of up to 600 Mbps and an outdoor range of up to 250 meters, can indeed be a suitable wireless communication technology for swarm UAV systems. This standard offers higher data rates compared to previous Wi-Fi standards, making it suitable for transmitting large amounts of data, such as video streams, telemetry, and control information, within the UAV swarm. The higher data rates enable faster communication and can support real-time or near-real-time applications. With its outdoor range, it provides a sufficient coverage area for UAV swarm operations. This range allows UAVs to communicate with each other and the GCS over a significant distance, supporting missions that require coverage of larger areas or extended flight ranges. 802.11n is also a widely adopted and commonly available Wi-Fi standard, which means that it is likely to be supported by many off-the-shelf devices, including UAVs, routers, and GCS equipment. The ubiquity of 802.11n makes it convenient to integrate into swarm UAV systems without requiring specialized or proprietary hardware. Moreover, as a standard Wi-Fi technology, 802.11n provides interoperability between different devices and vendors. This ensures compatibility and facilitates communication between UAVs and the GCS, even if they are from different manufacturers or operate in diverse environments. The higher data rates and extended range of 802.11n support scalability in swarm UAV systems. As the swarm grows and more UAVs are added, the 802.11n standard can handle the increased communication demands, allowing for efficient coordination and collaboration among the UAVs. Considering all these advantages it provides, 802.11n standard was preferred for the wireless communication technology of the designed system.

Since Wi-Fi has been chosen as the desired wireless communication technology for the swarm UAV system, the appropriate network topology should be determined to facilitate efficient communication and coordination within the swarm. Flying Ad Hoc Networks (FANETs) are a specific type of network topology designed for communication and coordination among UAVs. FANETs enable UAVs to establish dynamic, self-configuring networks without relying on any fixed infrastructure. FANETs can be categorized into four architectures, which can be grouped into two categories based on their characteristics as centralized and decentralized architectures. The first architecture shown in Figure 2a is central communication architecture refers to a centralized approach where a GCS acts as the central hub for communication and control within the swarm. The architecture serves as a centralized point for mission planning, decision-making, data aggregation, and command dissemination. It facilitates bidirectional communication between the GCS and the UAVs, allowing for the exchange of telemetry data, control commands, mission updates, and other relevant information. The GCS collects data from the UAVs, aggregates it, and performs necessary processing or analysis. This involves combining telemetry data from multiple UAVs, generating situational awareness, or executing algorithms for cooperative tasks such as formation flying or coordinated mission planning. The GCS also makes decisions based on the

received data, mission objectives, and operator inputs. It can issue commands or directives to the individual UAVs, guiding their behavior, trajectory, or mission tasks. The GCS also monitors and supervises the UAVs' status and ensures compliance with mission requirements.

Ad Hoc UAV Network, which is a decentralized architecture, refers to a self-configuring network established among UAVs without the need for any fixed infrastructure or central control. In this network, as shown in Figure 2b, UAVs communicate with each other directly, forming dynamic connections and collaborating to achieve common objectives. Each UAV acts as a node in the network and collaboratively builds and maintains network connections with other UAVs based on their relative positions and communication capabilities. A UAV in the swarm, called *backbone* UAV, acts as a front between the GCS and the swarm by acting as a gateway between other UAVs in the swarm. As its name suggests, these networks employ ad hoc routing protocols that enable UAVs to forward data packets to other UAVs, establishing multi-hop communication. These protocols consider factors such as UAV mobility, link quality, energy constraints, and network topology changes to facilitate efficient and reliable data transmission. The network topology in these networks is highly dynamic due to the mobility of UAVs. As UAVs move, join, or leave the network, the network topology dynamically adapts and reconfigures to maintain connectivity and efficient data exchange.

A Multi-Group UAV Network, another decentralized architecture, refers to a network architecture that involves the formation of multiple groups or sub-networks within a larger swarm of UAVs. In this configuration, UAVs are organized into distinct groups, with each group operating as a cohesive unit with its own internal communication and coordination mechanisms as shown in Figure 2c. UAVs in the swarm are divided into multiple groups based on specific criteria such as mission requirements, geographical locations, functional roles, or communication constraints. Group formation allows for focused coordination and communication within smaller subsets of UAVs, enabling efficient task allocation and management. Within each group, UAVs establish direct communication links or use short-range communication technologies to exchange information, share data, and coordinate their actions. Intra-group communication facilitates local coordination, cooperative sensing, and collaboration within the individual groups. Each group may have a designated group leader, backbone UAV, responsible for coordinating the activities of the UAVs within that group. Group leaders facilitate intra-group communication, task assignment, data fusion, and overall coordination. They serve as intermediaries for inter-group communication, relaying information between their respective groups and the larger network.

A Multi-Layer UAV Ad Hoc Network, on the other hand, refers to a network architecture that organizes UAVs into multiple layers or tiers, each with its own communication and coordination mechanisms as shown in Figure 2d. Each layer represents a distinct level of the hierarchy, with UAVs organized into different groups or sub-networks based on their assigned layer. UAVs communicate within their respective layers, forming intra-layer communication links. Intra-layer communication facilitates local coordination, data exchange, and collaborative tasks within the UAVs belonging to the same layer. UAVs within the same layer share a common communication range and may employ specific protocols or

frequencies optimized for that layer. The hierarchical structure allows for task assignment and allocation within the network.

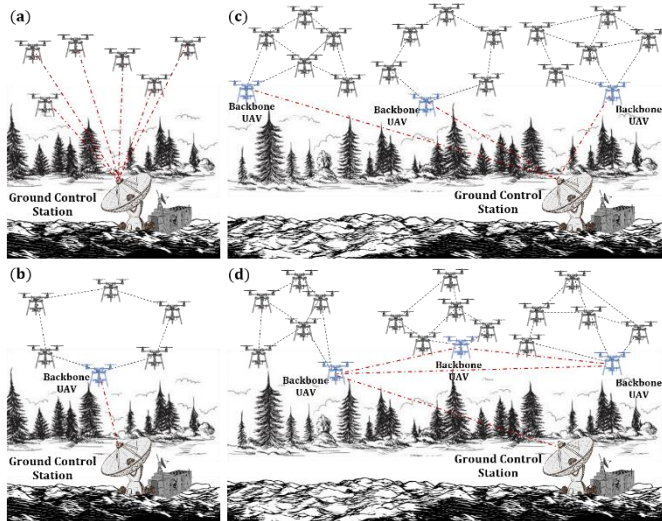


Figure 2. UAV network architectures: Central communication network (a), UAV ad hoc network (b), multi-group UAV network (c), multi-layer UAV ad hoc network (d)

Higher-layer UAVs can delegate tasks or commands to lower-layer UAVs, enabling distributed mission planning, data collection, or coordinated actions. The layered architecture facilitates efficient task distribution and enhances the scalability of the swarm system.

Given that the swarm UAV used in this study is a homogeneous quadcopter wing structure and will primarily be employed for aerial surveillance applications, the UAV Ad Hoc Network architecture appears to be well-suited for such a scenario. The fact that the swarm consists of homogeneous quadcopter UAVs implies that all the UAVs have similar characteristics, including flight capabilities, speed, maneuverability, and flight patterns. This homogeneity simplifies communication and coordination within the swarm, as the UAVs can operate based on shared protocols and behavior patterns. However, regarding the scalability of the network, its ability to accommodate a growing number of UAVs while maintaining effective communication, efficient resource utilization, and overall network performance needs to be addressed. Scaling an ad hoc UAV network can pose various challenges due to the dynamic and decentralized nature of the network. The network topology plays a crucial role in scalability. Ad hoc UAV networks can adopt different topologies, such as a mesh network or a hierarchical structure. Mesh networks provide better scalability as they allow for direct communication between UAVs, while hierarchical structures can manage large-scale networks by organizing UAVs into groups with designated leaders. The choice of routing protocols is also essential for scalability. Efficient routing protocols should adapt to the network size and dynamically adjust routes based on changing network conditions, such as UAV mobility, link quality, and network congestion. Moreover, as the number of UAVs in the network increases, the available bandwidth must be carefully managed and allocated among the UAVs to avoid congestion and ensure fair resource utilization. Dynamic bandwidth allocation and spectrum management techniques can be employed to optimize resource allocation and enhance scalability.

3.2 Wireless LAN and routing protocol configuration

When selecting a WLAN network standard for a UAV swarm, several factors should be considered, including data rate requirements, range, interference, and compatibility with UAV and GCS equipment. The choice would also depend on the number of UAVs involved, the distance between the UAVs, and the desired level of performance and reliability. IEEE 802.11n standard is widely adopted and offers higher data rates compared to earlier standards. It supports both the 2.4 GHz and 5 GHz frequency bands, providing flexibility in channel selection and potential for reduced interference. 802.11ac standard provides even higher data rates compared to 802.11n, reaching several Gbps. This standard operates primarily in the 5 GHz frequency band, offering the advantage of reduced interference from other devices operating in the 2.4 GHz band. The latest WLAN standard, IEEE 802.11ax, brings improvements in data rate, capacity, and efficiency. It supports both 2.4 GHz and 5 GHz frequency bands and introduces advanced technologies, enabling simultaneous connections with multiple devices and enhancing network performance in crowded environments. For a UAV swarm, it would be beneficial to choose a WLAN standard that offers a balance between data rate, range, and reliability. The choice would also depend on the size of the swarm and the density of UAVs in a given area. If the UAV swarm requires high data rates and long-range communication, 802.11ac or 802.11ax would be suitable options. These standards offer higher throughput and improved spectral efficiency, allowing for better performance in crowded environments. Alternatively, if the UAV swarm operates at shorter distances or requires a lower data rate, 802.11n might be sufficient. It provides a good balance between performance and cost. In any case, it is essential to consider potential interference and congestion issues in the deployment area. UAV swarms typically operate in congested environments, and careful channel planning, frequency management, and interference mitigation techniques should be implemented to ensure reliable and uninterrupted communication.

Flooding and routing are two distinct approaches used to guide messages through a network. Flooding is a simple approach in which a message is sent to all connected nodes in the network as shown in Figure 3. Each node that receives the message then retransmits it to all its neighboring nodes, excluding the node from which it received the message. This process continues until the message reaches all nodes in the network. The key advantage of flooding is its simplicity, thus, it does not require any complex routing algorithms. Flooding is inherently robust as it propagates messages through all available paths, making it suitable for scenarios where there may be network failures or dynamic topology changes. However, since the message is sent to all nodes, there is a high possibility of message duplication, resulting in network congestion and increased overhead. Flooding can also cause redundant transmissions and consume significant network resources, especially in large networks, which can lead to decreased performance.

Routing involves determining a specific path for a message to follow from the source node to the destination node as shown in Figure 3. Routing algorithms are used to make intelligent decisions about how to forward messages based on network topology, node states, and routing metrics. These algorithms aim to find the optimal or near-optimal paths for message delivery, reducing message duplication and network

congestion. Routing allows for more efficient utilization of network resources, making it better suited for large networks. They can adapt to changes in network conditions, such as node failures or topology changes, by recalculating routes dynamically. On the other hand, these algorithms can be complex to design and implement, requiring careful consideration of various factors, such as network topology, link costs, and routing protocols. Routing adds overhead to the network as nodes need to maintain routing tables and exchange routing information with neighboring nodes.

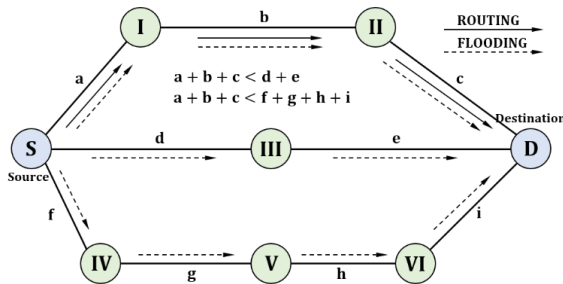


Figure 3. Flooding and routing strategies through a network

To summarize, flooding is a straightforward approach where messages are broadcasted to all nodes, while routing involves intelligent decision-making to determine specific paths for message delivery. Flooding is simple but may lead to message duplication and increased overhead. Routing, on the other hand, aims to optimize message delivery, resulting in better efficiency and scalability, but at the cost of added complexity. The choice between flooding and routing depends on the specific requirements of the network, including factors like network size, reliability, and the desired trade-off between simplicity and efficiency. Flooding approach in a UAV swarm network, while providing reliable packet delivery, can lead to increased bandwidth usage and associated costs. The use of high-quality radio transceivers capable of handling the increased bandwidth requirements come at a higher cost compared to standard counterparts. Therefore, the routing approach is preferred for the designed system.

Considering the routing protocols, topology-based protocols are popular in FANETs for UAV swarm applications due to their effectiveness in managing communication and coordination among the UAVs. These protocols utilize the network's topology information to make routing decisions, optimize resource utilization, and enhance overall network performance. Examples of topology-based routing protocols used in FANETs include Optimized Link State Routing

(OLSR), Destination-Sequenced Distance Vector (DSDV), Ad-Hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR), and Hybrid Wireless Mesh Protocol (HWMP). OLSR is designed to establish and maintain routes in a distributed manner by exchanging routing information among nodes within the network. It aims to optimize link state routing for efficient communication and reduced overhead. DSDV is based on the distance vector routing algorithm and aims to establish and maintain routing tables at each node in the network. DSDV provides a distributed and loop-free routing mechanism for efficient communication in dynamic environments. AODV establishes routes between nodes only when necessary, optimizing the use of network resources and reducing control overhead. It offers reactive routing, meaning that it creates routes on-demand based on specific data transmission needs. DSR is designed to establish and maintain routes between nodes dynamically, allowing for efficient communication in dynamic and mobile environments. DSR is characterized by its source routing mechanism, where the complete route is included in the packet header, allowing for flexibility and adaptability in route selection. HWMP combines both proactive and reactive routing approaches to provide efficient and reliable communication in dynamic environments.

The performances of the mentioned topology-based routing protocols were tested with a series of simulations. The simulations were carried out with the network simulator ns-2, and the mobilization of the UAVs was created with the open-source simulation platform Webots. While UAVs varying between 20 and 80 were used in the simulations, the speed ranges of the UAVs were adjusted to be between 15 m/s and 30 m/s, and the area where the flight mission would be performed was 5 km×5 km. Other simulation parameters are: transmission rate is 4 Mbps, packet size is 512 bytes, MAC protocol is 802.11n, max packet size in queue is 50, channel capacity is 2 Mbps, and antenna type is omni directional. The performances of the protocols were evaluated with packet delivery ratio (PDR), average latency, and throughput metrics. PDR indicates the reliability or success rate of packet delivery in a communication network. It represents the ratio of the number of packets successfully received at the destination to the total number of packets sent from the source. Average latency is used to measure the average time it takes for a data packet to travel from the source node to the destination node. Throughput, on the other hand, is a measure of the data transfer rate in a network, indicating the amount of data that can be transmitted over the network within a specific time period.

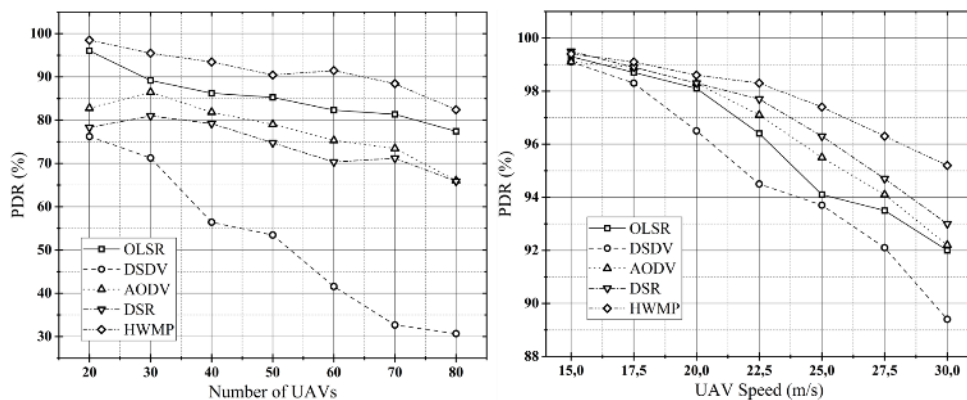


Figure 4. PDR performance metric results of topology-based routing protocols

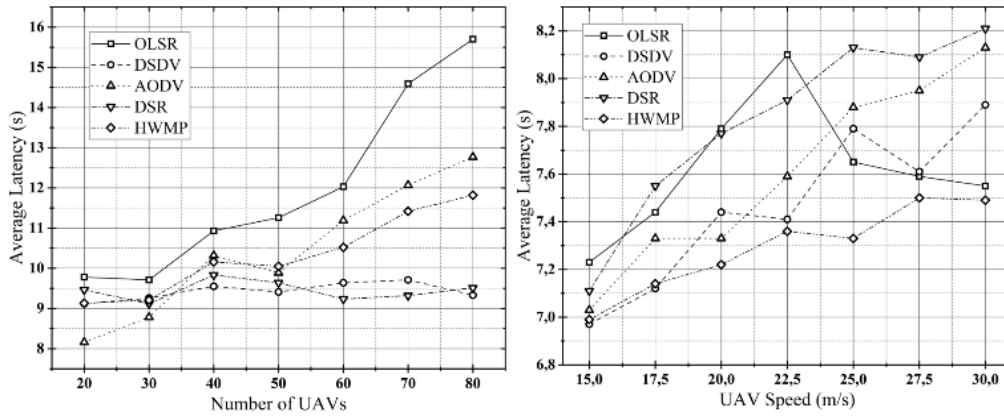


Figure 5. Average latency performance metric results of topology-based routing protocols

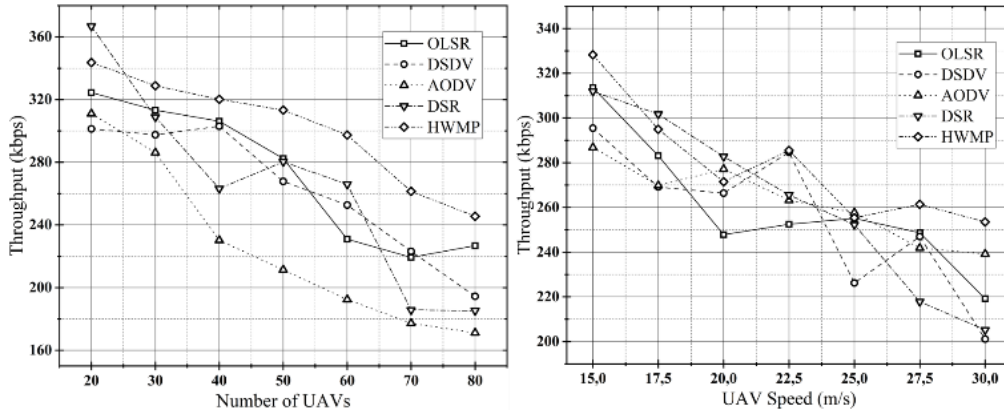


Figure 6. Throughput performance metric results of topology-based routing protocols

The simulation results are shown in Figures 4, 5 and 6. Figure 4 illustrates the variation of PDR based on the number and speed of UAVs, Figure 5 depicts the variation of average latency according to the number and speed of UAVs, and Figure 6 shows the variation of throughput based on the number and speed of UAVs. In general, HWMP outperforms other methods, demonstrating notably higher values for both PDR and throughput as the number and speed of UAVs vary. Conversely, when UAV speed changes, HWMP exhibits the lowest average latency value, whereas this value remains at an intermediate level when the number of UAVs changes.

3.3 Unmanned aerial vehicle

A flight controller is a crucial component of a UAV that manages the aircraft's flight operations. It is essentially the brain of the UAV, responsible for controlling various aspects of its flight, stability, and navigation. The flight controller's hardware includes a microcontroller that processes the sensor data and executes control algorithms. This central processing unit (CPU) is responsible for performing calculations, running control loops, and managing the overall flight operations. Pixhawk flight controllers are designed to be compatible with a wide range of UAV platforms, making them versatile for different types of aircraft, including multirotors, fixed-wing planes, helicopters, and even hybrid configurations. They run on open-source software firmware, primarily the ArduPilot software stack. ArduPilot is a robust and well-established open-source autopilot system that provides a comprehensive set of flight control features, autonomous flight modes, and mission planning capabilities. It offers a flexible and customizable platform for UAV development and

experimentation. QGroundControl allows users to configure and customize various parameters of the Pixhawk flight controller. It provides a graphical interface to adjust settings such as flight modes, control gains, sensor calibration, and communication options. The integration between Pixhawk flight controllers and QGroundControl provides a comprehensive ground control solution for configuring, monitoring, and controlling UAVs. It simplifies the setup process, enables autonomous mission planning and execution, and allows users to monitor flight data in real-time. All these beneficial properties mentioned are the source of motivation to choose Pixhawk and ArduPilot within the scope of this study.

To expand the capabilities of the UAVs, Broadcom 2711, which refers to the system-on-chip (SoC) model used in the Raspberry Pi 4 single-board computer (SBC), was preferred. The Broadcom BCM2711 chip, specifically designed for the Raspberry Pi 4, serves as the main processing unit of the SBC. It incorporates a quad-core ARM Cortex-A72 processor running at up to 1.5 GHz. This provides improved performance compared to previous Raspberry Pi models. The SoC integrates a VideoCore VI GPU, which offers enhanced graphics capabilities, supporting 3D graphics acceleration, video decoding, and encoding. It also supports onboard dual-band Wi-Fi (2.4 GHz and 5 GHz) and Bluetooth 5.0, enabling wireless connectivity and IoT applications. Thus, together with these features, it is aimed to realize advanced capabilities such as swarm networking, swarm formation and coordination and embedded decision-making. Furthermore, Panda Wireless PAU06 Wi-Fi adapter is used in the UAV swarm to provide wireless connectivity to individual UAVs within the swarm. The adapter is a reliable and affordable solution for adding

wireless connectivity to UAVs that lack built-in Wi-Fi capability or require an upgrade to a faster or more stable connection. This facilitates coordination, sharing of information, and synchronization among the swarm members. The adapter connects to the UAV's onboard computer or flight controller via a USB interface. It also enables the UAVs to establish a wireless connection with a GCS or network. This allows for real-time monitoring, control, and data exchange between the GCS and the UAV swarm. The adapter can be configured to create an ad hoc network or join an existing wireless network. This network can then serve as the communication backbone for the UAV swarm, enabling seamless connectivity and information exchange.

When considering antennas for the UAV swarm, it's important to choose an antenna type that suits the specific requirements of the swarm's communication system and operational needs. It is essential to select an antenna operating in the frequency range suitable for the communication system, which are commonly used for Wi-Fi communication in UAVs. The antenna's gain determines its ability to radiate and receive signals effectively. Higher gain antennas can provide longer-range coverage, but may be more directional. An antenna with an appropriate gain to meet design range requirements while maintaining omnidirectional coverage must be preferred. The physical dimensions and weight of the antenna, as they can impact the UAV's balance, aerodynamics, payload capacity and flight performance, also need to be considered. A few common antenna types that may be suitable for UAV swarm communication include Patch, Yagi-Uda, Helical, and Omni Directional antennas. For the proposed system, the most suitable of these antenna types is omnidirectional, and therefore Phoenix CAO868 antenna is preferred. These omnidirectional antennas radiate and receive signals in all directions equally. They provide a 360-degree coverage pattern and are suitable when maintaining communication between multiple UAVs within the swarm is crucial.

A neural accelerator, also known as a neural processing unit (NPU) or artificial intelligence (AI) accelerator, is a specialized hardware component designed to accelerate the execution of neural network models. When it comes to object detection applications in UAVs, a neural accelerator can significantly enhance the performance and efficiency of the detection process. By offloading the intensive computational tasks involved in object detection to a dedicated hardware accelerator, real-time and efficient detection can be achieved. The NVIDIA Jetson Nano, Google Coral Dev Board (GCDB) 4GB, and Intel Movidius are examples of small and lightweight neural accelerator platforms that are well-suited for mounting on UAVs. These platforms offer powerful AI acceleration capabilities while being compact and energy-efficient, making them suitable for onboard AI processing in UAV applications. The NVIDIA Jetson Nano features a powerful GPU and a quad-core ARM Cortex-A57 CPU, providing high-performance AI processing. With its small form factor and low power consumption, the Jetson Nano is commonly used for real-time AI inference tasks on UAVs. The GCDB is a compact AI development platform featuring Google's Edge TPU (Tensor Processing Unit). It offers hardware acceleration for AI workloads, including object detection, and is designed for low-power edge computing. The GCDB's small size and light weight make it suitable for UAV applications where space and weight constraints are important. Intel Movidius, on the other hand, is a family of AI accelerator chips designed for edge computing. These chips, such as the

Intel Movidius Neural Compute Stick, provide high-performance AI inference capabilities in a compact and portable form factor. They are optimized for low-power applications and can be easily integrated with UAV systems for on-device AI processing.

All three of these platforms offer a combination of powerful AI acceleration, compact size, and low power consumption, making them suitable for mounting on UAVs. They enable real-time object detection and other AI tasks directly on the UAV, allowing for efficient and autonomous operation. However, in terms of price, GCDB is more expensive than other options. Although Intel Movidius Stick, seems like a cheaper solution, it requires a Raspberry Pi or another computer to run, which increases the cost. This puts the NVIDIA Jetson Nano ahead in terms of price. Another comparison parameter is performance values that can be measured in terms of frames per second. The fact that Intel Movidius has relatively lower performance than Jetson Nano and GCDB makes it strictly not preferred. Among the remaining two neural accelerators, GCDB's TPU and its machine learning models allow Jetson Nano to get ahead of it, but it's limited to TPU-optimized TensorflowLite models, limiting the possibilities to use it with custom layers, greatly reducing its flexibility. This directly shows that the only preferable piece of hardware is the Jetson Nano with Inception V4, which is the most flexible device thanks to its ability to run models from any framework. Jetson Nano was chosen for the proposed customizable UAV system, as it provides the best balance between cost, flexibility and performance among the options available.

4. THE PROPOSED SYSTEM AND ITS DEPLOYMENT

The Holybro Pixhawk 4 flight controller is used to control the movement and navigation of the UAVs. It plays a crucial role in managing various aspects of the drone's flight and interacts with other peripherals and sensors to ensure safe and reliable operation. Extremely light and small in size, this controller includes the Neo-M8N GPS module as well as various internal sensors such as the ICM-20689 accelerometer, BMI055 gyroscope, IST8310 magnetometer, and MS5611 barometer. The Neo-M8N GPS module, which comes as part of the set, leverages the IST8310 integrated magnetometer to improve the accuracy in determining the UAV's heading. The flight controller and other peripherals on the UAVs are powered by the Pixhawk's power module, PM02, which also offers voltage regulation and current measurement capabilities. The PM02 is responsible for reducing the high voltage output from the Li-Po battery, which is typically around 22.2 volts (for a 6S Li-Po battery), to a lower and stable voltage suitable for powering the flight controller and other electronic components. The reduced voltage is typically 5 volts, which is the standard operating voltage for many flight controllers and onboard devices. As in many UAV setups, a power distribution board is also used as part of the hardware to manage power distribution to various components on the UAV, including motors, electronic speed controllers (ESCs), and other onboard devices. The primary function of the power distribution board is to distribute power from the main battery to the individual motors and ESCs that drive the propellers. It acts as a central hub for power connectivity, ensuring that each motor and ESC receives the appropriate voltage and current to function correctly. An RC (Radio Control) controller is used

to provide manual control and intervention by the pilot. Having an RC controller as part of the UAV setup provides an additional layer of safety and control, giving the pilot the ability to intervene and make real-time decisions during flight. This is particularly valuable in situations where autonomous systems may not be able to account for every possible scenario, or when there is a need for human decision-making in critical situations.

4.1 UAV on-board avionics

Figure 7 depicts the schematic of the on-board hardware components along with their interconnection in the proposed UAV system. In the avionics, Raspberry Pi 4 (RP4) was thoughtfully incorporated as a backup computer, providing an additional layer of redundancy and enhancing the overall system's reliability. Simultaneously, Jetson Nano, with its exceptional capabilities in onboard object detection, was purposefully selected to bolster the system's intelligence and empower it with real-time object recognition and analysis capabilities. Both devices have unique strengths and can complement each other to enhance the overall functionality and performance of the UAV. RP4 is a versatile single-board computer capable of running a full-fledged operating system (like Linux) and supporting a wide range of applications. It can handle general-purpose computing tasks, communication with peripherals, and interfacing with various sensors. RP4 can also be used for flight control algorithms, handling sensor data, and performing navigation calculations. It can handle telemetry data and communication with GCS or other remote devices, and facilitates real-time data transmission and communication with the pilot or ground operators. The Jetson Nano, on the other hand, is a powerful edge AI computing platform specifically designed for running AI and deep learning workloads. It can perform complex tasks, such as onboard object detection, image recognition, and real-time analysis of visual data. By using it for onboard object detection, the UAV can process visual data locally without relying on a remote server or cloud, reducing latency and enhancing response times for critical tasks. Jetson Nano can enable the UAV to detect and recognize objects in its environment, which is valuable for applications, such as obstacle avoidance, target tracking, or environment mapping. By combining the capabilities of the RP4 and the Jetson Nano, the UAV system benefits from a balanced computing setup. RP4 handles flight control, navigation, and communication tasks, while the Jetson Nano brings AI and deep learning capabilities for onboard perception and object detection. This division of tasks will allow the UAV to perform advanced autonomous operations, efficiently process visual data, and execute complex tasks that would be challenging for a single computing device alone.

Coordinating between the Jetson Nano and RP4 for sharing a video stream while allowing each device to use the video at the desired frame rate per second (FPS) requires careful communication and synchronization between the two devices. A general approach to achieve this coordination should be as follows. UAV's camera captures the video frames and sends them to both the Jetson Nano and the RP4 simultaneously. They process and encode the received frames independently. A network connection between the Jetson Nano and RP4 can

then be established using Wi-Fi. Both devices need to communicate and negotiate the desired frame rate for video processing and transmission. This is done through a communication protocol, such as TCP/IP or UDP, where both devices exchange information about the desired frame rate. To ensure synchronization between the devices, timestamps should also be included in the video frames or sent separately along with the frames. This way, both devices can accurately timestamp each frame they receive, even if there are slight variations in frame rates. Each device needs to manage a buffer or queue to handle incoming video frames. The buffer size can be adjusted based on the frame rate difference between the two devices to avoid overflows or underflows. The devices may need to adapt their frame rate based on the available processing power and computational load. For example, the Jetson Nano might process frames at a higher frame rate than the Raspberry Pi if it has more computational resources.

Sending images of detected objects over the network to the GCS lead to network saturation, especially when multiple UAVs are detecting objects in multiple frames simultaneously. This result in increased network bandwidth consumption and potential delays in transmitting critical data, impacting real-time decision-making and overall system performance. Several strategies can be employed to address this challenge. Instead of sending the entire image, extracting and transmitting relevant metadata about the detected objects, such as bounding box coordinates, confidence scores, and object classifications, is a useful option. This reduces the amount of data transmitted while providing essential information to the GCS for decision-making. A priority mechanism can be implemented to determine which objects are more critical to transmit in real time. Low priority objects or objects with low confidence scores can be buffered or aggregated before transmission to reduce network load. If high resolution images are not crucial for object identification, downsampling of images before transmission is another option. Because lower-resolution images consume less bandwidth while still conveying essential information. Additionally, image compression algorithms (e.g., JPEG, PNG) can be used to reduce the image size before transmission. This helps minimize the network load without compromising critical details. Moreover, as proposed, UAVs can be equipped with built-in processing capabilities and decision-making algorithms. The UAV can evaluate the significance of detected objects locally and decide whether to transmit the data immediately or wait for a more appropriate time.

On the other hand, it is also possible to address the problem through event-based or multi-level communication. Event-based communication is a potential solution, where the UAV only sends an alert or notification to the GCS when a significant event occurs, such as detection of high-priority objects or potential threats. Alternatively, as discussed in the previous section, a multi-level communication architecture can also be used, optimizing network resources, where critical data is transmitted through a high-priority link and less critical data is sent through a lower-priority link. By employing these strategies and optimizations, the system can effectively manage network congestion while providing the GCS with essential information about detected objects in real-time, ensuring efficient and reliable operation of the UAV swarm.

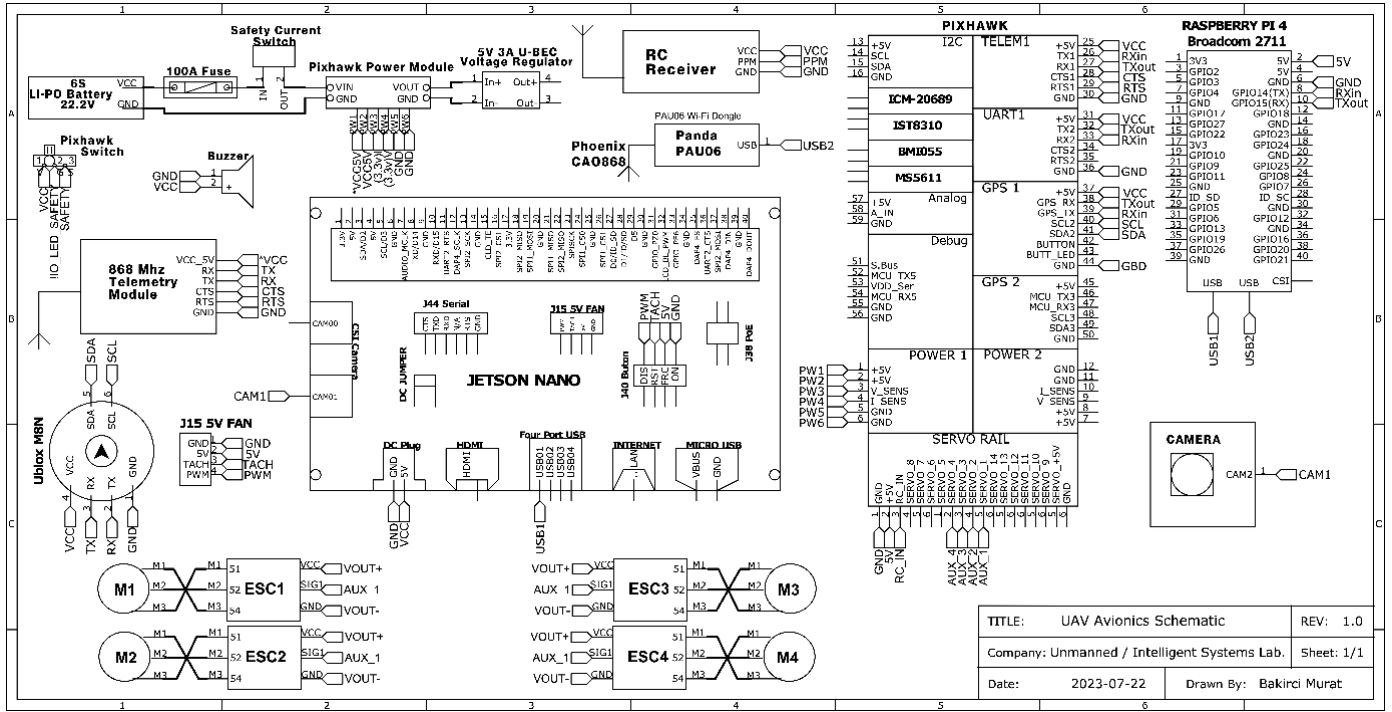


Figure 7. On-board avionics schematic of a swarm member UAV

4.2 Control framework

The lightweight and low-bandwidth MAVLink (Micro Air Vehicle Link) protocol is used for communication between the GCS and the flight controller of the UAV. MAVLink is specifically designed for resource-constrained systems like UAVs, where bandwidth and processing power are limited, making it an ideal choice for efficient communication. MAVLink messages are used to exchange data between the GCS and the flight controller. These messages are organized into packets, which consist of a header, payload, and optional checksum for error checking. Each message contains a specific set of fields with data related to UAV status, telemetry, and control commands. Typically, telemetry, command, parameter, and mission message types are used in MAVlink communication. Telemetry messages provide real-time information about the UAV's status, including its position, attitude, altitude, GPS coordinates, battery voltage, system health, and other sensor readings. They allow the GCS to monitor the UAV's flight and health status. Command messages are used to send flight commands from the GCS to the UAV. Examples of commands include takeoff, land, change altitude, set waypoints, and perform specific flight maneuvers. These commands guide the UAV's behavior and mission execution. Various parameters and configurations on the flight controller are set or updated through parameter messages. Parameters can include control gains, flight modes, sensor calibrations, and other settings that influence the UAV's behavior. Mission messages, on the other hand, define pre-planned mission waypoints and actions. The GCS can upload a series of waypoints and actions to the UAV, and the flight controller executes them in autonomous mode. In summary, using the MAVLink protocol, uninterrupted and reliable communication is established with UAVs, and real-time monitoring, control and task execution is provided.

To route MAVLink messages from the GCS to Pixhawk flight controller using a RP4 as the intermediate device, the RP4 must be set up as a network bridge or a TCP proxy. This

configuration will allow the MAVLink messages to pass through the RP4, ensuring they reach the appropriate flight controller. In the network bridge configuration, the RP4 acts as a transparent bridge between two network interfaces, allowing data to flow seamlessly between them. In this case, the RP4 would have two network interfaces: one connected to the GCS and another connected to the Pixhawk. In the TCP proxy configuration, the Raspberry Pi intercepts and forwards TCP packets containing MAVLink messages between the GCS and the Pixhawk. The RP4 acts as an intermediary, receiving MAVLink messages from the GCS, processing them if needed, and then forwarding them to the Pixhawk. Both methods enable the RP4 to facilitate communication between the GCS and the Pixhawk, allowing the MAVLink messages to be efficiently transmitted and received. The choice between network bridge and TCP proxy depends on the specific requirements of the UAV system and the network architecture being used. In the proposed system, creating a bidirectional channel between the UART connection of the RP4 and Pixhawk, as well as the TCP connection established with the GCS over the mesh network, will enable seamless and simultaneous communication in both directions. The bidirectional channel will then allow data to flow between the RP4, Pixhawk, and the GCS, facilitating real-time monitoring, control, and telemetry exchange. This configuration is achieved through the script shown in Table 1. Thus, the RP4 is set up to relay MAVLink messages from the GCS to the Pixhawk.

Lines 2-4 in the algorithm describe the process of configuring a TCP server with port number 5760 on the Raspberry Pi. The TCP server is configured to listen for incoming connections on a specific port. In this case, the port number is set to 5760. This is a common port used for MAVLink communication. The RP4 initiates the TCP server, and it starts listening for incoming connections on port 5760. The GCS software, QGroundControl, establishes a TCP connection to the RP4's IP address and port 5760. This allows the GCS to send MAVLink messages to the RP4 and vice

versa. Once the TCP connection is established between the GCS and the RP4, MAVLink messages are exchanged between them. The RP4 acts as a bridge, routing the MAVLink messages between the GCS and the flight controller, Pixhawk, via the UART connection. The TCP server on the Raspberry Pi enables bidirectional communication between the GCS and the Pixhawk as discussed. The GCS can send control commands, mission updates, and requests for telemetry data to the UAV through the TCP connection. Simultaneously, the Pixhawk can send telemetry data, status updates, and responses to the GCS through the RP4's TCP server.

Table 1. MAVLink router configuration file

#MAVLink Router Configuration File
[General]
TcpServerPort=5760
ReposrtStats=false
MavlinkDialect=common
[UartEndpoint serial0]
Device=/dev/serial0
Baud=56700

Lines 6 and 7 in the configuration establishes the UART (serial) communication on the RP4, specifically connecting to the telemetry port of the Pixhawk. This configuration effectively creates another endpoint that bridges the TCP and UART communication channels. The UART pins of the RP4 are configured to interface with the telemetry port of the Pixhawk. The baud rate is set to 56700, which is the communication speed used for MAVLink communication between the RP4 and the Pixhawk. With the UART configured, MAVLink messages received from the GCS over the TCP connection (established at port 5760) can now be forwarded to the Pixhawk through the UART connection. Likewise, telemetry data and messages from the Pixhawk can be transmitted back to the GCS through the RP4's TCP connection. The RP4, acting as a central hub, routes MAVLink messages between the TCP and UART endpoints. It receives MAVLink messages from the GCS over the TCP connection and forwards them to the Pixhawk through the UART. Similarly, MAVLink messages received from the Pixhawk over the UART are transmitted back to the GCS over the TCP connection. Similarly, the bridge formed by lines 6 and 7 enables bidirectional communication between the GCS and the flight controller.

In a multi-UAV swarm scenario, assigning unique MAVLink System IDs to each UAV (flight controller) and each GCS is essential to prevent ID collisions and ensure smooth communication and coordination among all devices. MAVLink uses system IDs to address unicast messages, which are messages sent from one specific sender (e.g., GCS or flight controller) to a specific recipient (e.g., a particular UAV in the swarm). To ensure that messages are delivered accurately and to the intended recipient, each device must have a unique system ID. If multiple UAVs or GCSs in the same network have the same system ID, there can be conflicts and confusion, leading to incorrect message delivery or processing. Having unique IDs for each device prevents such collisions and ensures proper message routing. In the UAV swarm, since efficient coordination among individual UAVs is crucial, each UAV needs to receive commands and telemetry data tailored to its unique role in the swarm. Unique system IDs enable targeted communication, allowing swarm coordination without interference. In a scenario where multiple GCSs are

employed to oversee the swarm, it is essential for each GCS to possess a distinct system ID. This enables seamless communication between each GCS and the UAVs without conflicting with other GCSs in the network.

To implement unique MAVLink system IDs, unique ID assignment, GCS system ID setting, and configuration steps are performed respectively. First, a unique system ID is assigned to each UAV in the swarm. For example, each UAV is assigned an ID starting from 1 (e.g., UAV#1 - system ID: 1, UAV#2 - system ID: 2, etc.). Then it is necessary to ensure that each GCS has its own unique system ID. This is especially important if multiple GCSs are used to control different groups of UAVs within the swarm. Finally, the system ID is configured in each UAV's flight controller settings and in each GCS software, QGroundControl. This is typically done in the settings or configuration options of the respective devices or software.

Planning missions for both individual UAVs and UAV swarms requires careful consideration of several critical aspects to ensure safe and efficient operation. The first key aspect that need to be taken into account during mission planning is mission duration requirements that need to be evaluated, including the time required to complete all objectives and return to the starting point or land safely. In this context, factors such as battery capacity, endurance and energy consumption of onboard devices should also be considered to estimate the total duration of the mission. Communication range between the GCS and the UAVs also need to be evaluated. It is essential to ensure that the UAVs can maintain a reliable connection with the GCS throughout the mission. Communication technologies such as long-range radio modems or satellite communications may also be used if the mission involves long-distance or beyond line-of-sight operations. Another issue is the possible overlap of flight paths of swarm members. For UAV swarms, flight paths should be planned that allow sufficient separation and avoid collisions between individual UAVs. Overlapping flight paths can enhance mission robustness, redundancy, and coverage, especially when performing tasks like surveillance, mapping, or search and rescue.

To avoid possible collisions among UAVs or with other objects in the airspace, collision avoidance algorithms and mechanisms can be implemented, especially for UAV swarms. These systems can be based on proximity sensors, onboard cameras, or communication-based collision avoidance protocols. Since the proposed system does not include collision avoidance hardware, efforts should be made to prevent the flight paths of different UAVs from intersecting. This problem can be solved with QGroundControl, which provides an effective visual interface. For this, after defining clear mission objectives, certain waypoints are determined for UAVs to follow through QGroundControl. Waypoints are GPS coordinates or geographic locations that define the UAV's flight path and tasks. They can be used for navigating between specific points, performing aerial surveys, or following pre-defined flight patterns. With a similar approach, obstacles or no-fly zones along the planned flight path are also determined on the software and included in the pre-flight task list. Additionally, GPS coordinates of UAVs can be monitored in real-time with telemetry data during the mission and intervened when necessary. Having such ability to adjust the mission on-the-fly adds flexibility and adaptability to changing conditions. Moreover, this problem can be largely remedied by planning emergency procedures in case of system

failures, loss of communication or unexpected events. For example, procedures to return UAVs to a safe place or make an emergency landing could be included in the pre-flight task list. By taking these aspects into consideration, mission planners can create effective and safe UAV missions, whether for individual UAVs or UAV swarms, to accomplish various tasks efficiently and with optimal use of resources.

5. CONCLUSIONS

A highly versatile swarm UAV system specifically tailored for surveillance and object detection applications is introduced. The suggested system offers inclusive design criteria, enabling straightforward customization for diverse scenarios and user needs. The core focus of the design lies in establishing a robust communications infrastructure and equipping the UAVs with formidable built-in processing power. Choosing Wi-Fi (802.11n) offered a significant advantage in terms of substantially higher data transfer speeds compared to Zigbee and LoRa. This advantage proved to be crucial for bandwidth-intensive applications such as live video streaming, where the faster data transmission capability of Wi-Fi enabled smooth and seamless real-time video streaming without any notable delays. The collaboration between Raspberry Pi and Jetson Nano, utilized in the proposed system, resulted in a cost-effective solution. While the Raspberry Pi serves as an economical base for the system, the Jetson Nano functions as an efficient neural accelerator, optimizing AI-related tasks without incurring excessive costs. This symbiotic relationship between the two devices brought significant benefits to the swarm UAV system by equipping the UAVs with the intelligence to navigate complex environments, detect objects in real-time, and execute coordinated actions. By incorporating the decentralized approach of the Ad Hoc UAV Network into the proposed system, it gains the ability to seamlessly expand and adapt to evolving environments and mission requirements. Due to its inherent characteristics, this network is particularly well-suited for dynamic and unpredictable environments, significantly enhancing the mission performance of the proposed swarm UAV system. Thus, the ability of UAVs in the system to self-organize and maintain communication without relying on fixed infrastructure makes them highly adaptable in scenarios where the terrain or mission requirements change rapidly. Selecting the routing protocol for the proposed system has resulted in more efficient data transmission, achieved through the minimization of unnecessary broadcasting and reduced network congestion. This results in more optimal resource utilization and improved overall system performance. Even better, routing consumes less power compared to flooding, as it requires fewer broadcast transmissions. For UAVs with limited battery capacity, this energy-efficient approach ensures longer flight durations and extended mission capabilities.

5.1 Limitations

While choosing Wi-Fi (802.11n) offers higher data transfer speeds, it comes with the trade-off of a limited communication range compared to other wireless technologies. In environments with extensive distances between UAVs or when operating in remote areas, the range of Wi-Fi may

present challenges. The Raspberry Pi and Jetson Nano come with inherent limitations in terms of processing power and payload capacity. Consequently, the system may encounter constraints when handling computationally intensive tasks or carrying heavy payloads. The decentralized communication feature of Ad Hoc UAV Network may result in potential communication interference and reliability issues, especially in crowded or high-interference environments. The effectiveness of communication and coordination among UAVs could be affected in such scenarios. While the system is designed to be adaptable in dynamic environments, extreme weather conditions, unpredictable terrains, or challenging environmental factors could pose limitations to the UAVs' overall performance and functionality. The system's scalability may face challenges as the number of UAVs in the swarm increases. Given that UAVs operate on limited battery power, balancing the power consumption of all components, including the AI processing on Jetson Nano, will reduce the flight times, and thus, mission success. Additionally, the decentralized nature of Ad Hoc UAV Network may introduce security concerns regarding unauthorized access or interception of communication. Furthermore, without collision avoidance hardware, the UAVs may be prone to collisions with each other or obstacles, posing a safety risk during swarm operations, especially in crowded or complex environments. This may also impose limitations on the complexity of missions that the swarm system can undertake.

5.2 Future work

To enhance the communication range of Wi-Fi in the swarm UAV system, various techniques can be employed, including signal amplification, signal relaying, or the implementation of directional antennas. Exploring the development of dynamic channel allocation algorithms can be instrumental in efficiently managing the Wi-Fi spectrum and mitigating interference among UAVs. Applying decentralized decision-making algorithms that leverage the collaboration between Raspberry Pi and Jetson Nano may enable UAVs to make coordinated decisions based on local observations and AI processing. Implementing energy-efficient strategies for managing power consumption between Raspberry Pi and Jetson Nano can balance computational loads and optimize power utilization. This will be crucial to extending UAV flight times and enhancing overall mission endurance. Employing new and advanced object detection models, such as YOLOv8, can optimize Jetson Nano's real-time performance for object detection tasks. Additionally, fine-tuning existing models or exploring novel architectures can result in improved detection speed and accuracy. Collaboration and information sharing among Jetson Nano-equipped UAVs can be explored to collectively enhance their object detection capabilities. Moreover, new multi-hop communication protocols can be formulated to extend the range and coverage of the Ad Hoc UAV Network, particularly in scenarios with sparse or challenging terrain. Designing more efficient routing protocols tailored specifically for swarm UAV systems may result in a reduction of communication overhead and improved data transmission efficiency. Adaptive routing strategies that can dynamically adjust route selection based on changing network conditions and mission requirements can be developed. This capability will enable the system to respond to dynamic environments and optimize performance.

5.3 Potential impacts on real world applications

The potential impact of the proposed swarm UAV system on real-world applications could be transformative and highly beneficial in various fields. With its distinctive features, the proposed system can play a pivotal role in disaster response and relief operations. Through its real-time object detection capabilities, it can assess damage, locate survivors, and facilitate communication in areas affected by natural disasters, thereby enhancing the efficiency of rescue efforts. By integrating Raspberry Pi and Jetson Nano in the system, precise monitoring of crops and soil conditions in agricultural fields becomes possible, allowing farmers to optimize irrigation, fertilization, and pest control, thereby leading to increased crop yields and improved resource efficiency. The combination of Wi-Fi communication and real-time data processing can facilitate remote monitoring of infrastructures, identify potential issues, and prioritize maintenance activities. The proposed system's ability to cover large areas and collect real-time data can be utilized for environmental monitoring and conservation efforts, including tracking wildlife and monitoring deforestation. In scenarios where communication infrastructure is damaged or lacking, the Ad Hoc UAV Network can establish temporary communication networks, facilitating connectivity for emergency responders or remote regions. The system's collaborative capabilities can enhance situational awareness and communication in military and security operations. It can also assist in urban planning and traffic management by providing real-time traffic data, monitoring congestion, and aiding in urban development assessments. Additionally, the proposed system's Wi-Fi communication and routing protocol can optimize delivery and logistics operations, enabling efficient last-mile delivery and supporting supply chain management in challenging terrains. Furthermore, the system can be instrumental in scientific research and exploration, including studying wildlife behavior, monitoring climate patterns, and exploring inaccessible regions.

REFERENCES

- [1] Fan, B., Li, Y., Zhang, R., Fu, Q. (2020). Review on the technological development and application of UAV systems. *Chinese Journal of Electronics*, 29(2): 199-207. <https://doi.org/10.1049/cje.2019.12.006>
- [2] Ecke, S., Dempewolf, J., Frey, J., Schwaller, A., Endres, E., Klemmt, H.J., Tiede, D., Seifert, T. (2022). UAV-based forest health monitoring: A systematic review. *Remote Sensing*, 14(13): 1-12. <https://doi.org/10.3390/rs14133205>
- [3] Li, Y., Min, L., Dandan, J. (2022). Application of unmanned aerial vehicles in logistics: A literature review. *Sustainability*, 14(21): 14473. <https://doi.org/10.3390/su142114473>
- [4] Ajith, V.S., Jolly, K.G. (2021). Unmanned aerial systems in search and rescue applications with their path planning: A review. *Journal of Physics: Conference Series*, 2115(1): 1-12. <https://dx.doi.org/10.1088/1742-6596/2115/1/012020>
- [5] Bakirci, M., Ozer, M.M. (2023). Avionics system development for a rotary wing unmanned combat aerial vehicle. *IEEE 18th International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, Romania, pp. 327-334. <https://doi.org/10.1109/ICCP56966.2022.10053956>
- [6] Petersen, M., Samuelson, C., Beard, R.W. (2021). Target tracking and following from a multirotor UAV. *Current Robotics Reports*, 2(3): 285-295. <https://doi.org/10.1007/s43154-021-00060-7>
- [7] Zhang, H., Wang, L., Tian, T., Yin, J. (2021). A review of unmanned aerial vehicle low-altitude remote sensing use in agricultural monitoring in China. *Remote Sensing*, 13(6): 1221. <https://doi.org/10.3390/rs13061221>
- [8] Bakirci, M., Ozer, M.M. (2023). Surveillance, reconnaissance and detection services for disaster operations of IoT-based eVTOL UAVs with swarm intelligence. *11th International Symposium on Digital Forensics and Security*, Chattanooga, TN, pp. 1-6. <https://doi.org/10.1109/ISDFS58141.2023.10131702>
- [9] Wang, H., Rubenstein, M. (2020). Shape formation in homogeneous swarms using local task swapping. *IEEE Transactions on Robotics*, 36(3): 597-612. <https://doi.org/10.1109/TRO.2020.2967656>
- [10] Weng, L., Liu, Q., Xia, M., Song, Y.D. (2014). Immune network-based swarm intelligence and its application to unmanned aerial vehicle (UAV) swarm coordination. *Neurocomputing*, 125: 134-141. <https://doi.org/10.1016/j.neucom.2012.06.053>
- [11] Peng, Q., Wu, H., Xue, R. (2021). Review of dynamic task allocation methods for UAV swarms oriented to ground targets. *Complex Systems Modeling and Simulation*, 1(3): 163-175. <https://doi.org/10.23919/CSMS.2021.0022>
- [12] He, Y. (2020). Mission-driven autonomous perception and fusion based on UAV swarm. *Chinese Journal of Aeronautics*, 33(11): 2831-2834. <https://doi.org/10.1016/j.cja.2020.02.027>
- [13] Fabra, F., Zamora, W., Reyes, P., Sanguesa, J.A., Calafate, C.T., Cano, J.C., Manzoni, P. (2020). MUSCOP: Mission-based UAV swarm coordination protocol. *IEEE Access*, 8: 72498-72511. <https://doi.org/10.1109/ACCESS.2020.2987983>
- [14] Burkle, A., Segor, F., Kollmann, M. (2011). Towards autonomous micro UAV swarms. *Journal of Intelligent & Robotic Systems*, 61: 339-353. <https://doi.org/10.1007/s10846-010-9492-x>
- [15] Saffre, F., Hildmann, H., Karvonen, H., Lind, T. (2022). Monitoring and cordoning wildfires with an autonomous swarm of unmanned aerial vehicles. *Drones*, 6: 301. <https://doi.org/10.3390/drones6100301>
- [16] Arnold, R.D., Yamaguchi, H., Tanaka, T. (2018). Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *Journal of International Humanitarian Action*, 3: 18. <https://doi.org/10.1186/s41018-0045-4>
- [17] Liu, Y., Nie, J., Li, X., Ahmed, S.H., Lim, W.Y.B., Miao, C. (2021). Federated learning in the sky: Aerial-ground air quality sensing framework with UAV swarms. *IEEE Internet of Things Journal*, 8(12): 9827-9837. <https://doi.org/10.1109/JIOT.2020.3021006>
- [18] Mukherjee, A., Misra, S., Raghuvanshi, N.S. (2019). A survey of unmanned aerial sensing solutions in precision agriculture. *Journal of Network and Computer Applications*, 148: 102461. <https://doi.org/10.1016/j.jnca.2019.102461>
- [19] Mohammed, F., Idries, A., Mohamed, N., Al-Jaroodi, J., Jawhar, I. (2014). UAVs for smart cities: Opportunities

- and challenges. International Conference on Unmanned Aircraft Systems, Orlando, FL, pp. 1-6. <https://doi.org/10.1109/ICUAS.2014.6842265>
- [20] Mukherjee, A., Keshary, V., Pandya, K., Dey, N., Satapathy, S.C. (2018). Flying ad hoc networks: A comprehensive survey. In: Satapathy, S., Tavares, J., Bhateja, V., Mohanty, J. (eds). Information and Decision Sciences. Advances in Intelligent Systems and Computing, 701, pp. 639-580, Springer, Singapore. https://doi.org/10.1007/978-981-10-7563-6_59
- [21] Beegum, T.R., Idris, M.Y.I., Ayub, M.N.B., Shehadeh, H.A. (2023). Optimized routing of UAVs using bio-inspired algorithm in FANET: A systematic review. IEEE Access, 11: 15588-15622. <https://doi.org/10.1109/ACCESS.2023.3244067>
- [22] Noor, F., Khan, M.A., Al-Zahrani, A., Ullah, I., Al-Dhlan, K.A. (2020). A review on communications perspective of flying ad-hoc networks: Key enabling wireless technologies, applications, challenges and open research topics. Drones, 4(4): 65. <https://doi.org/10.3390/drones4040065>
- [23] Montoya, A.M., Sanchez-Aarnoutse, J.C. (2021). A comparative performance evaluation of routing protocols for flying ad-hoc networks in real conditions. Applied Sciences, 11(10): 4363. <https://doi.org/10.3390/app11104363>
- [24] Akhtar, Z., Garg, H., Singh, S.K. (2022). A survey: Recent advances in hierarchical clustering routing protocols. In: A. Kumar, G. Ghinea, S. Merugu, T. Hashimoto, (eds) Proceedings of the International Conference on Cognitive & Intelligent Computing. Cognitive Science & Technology, pp. 293-304. https://doi.org/10.1007/978-981-19-2350-0_29
- [25] Fang, W., Zhang, W., Shan, L., Assefa, B., Chen, W. (2020). Hierarchical routing protocol in wireless sensor network: A state-of-the-art review. International Journal of Computational Science and Engineering, 22(1): 107-113. <https://doi.org/10.1504/IJCSE.2020.107260>
- [26] Khan, M.K., Shiraz, M., Shaheen, Q., Butt, S.A., Akhtar, R., Khan, M.A., Changda, W. (2021). Hierarchical routing protocols for wireless sensor networks: Functional and performance analysis. Journal of Sensors, 2021(7459368): 1-18. <https://doi.org/10.1155/2021/7459368>
- [27] Benmansour, F.L., Labraoui, N. (2021). Comprehensive review on swarm intelligence-based routing protocols in wireless multimedia sensor networks. International Journal of Wireless Information Networks, 28(2): 175-198. <https://doi.org/10.1007/s10776-021-00508-9>
- [28] More, A., Kale, R. (2022). Review on recent research trends and applications in delay tolerant networks. 6th International Conference on Computing, Communication, Control & Automation, Pune, pp. 1-9. <https://doi.org/10.1109/ICCUBEA54992.2022.10011041>
- [29] Ramirez-Atencia, C., Camacho, D. (2018). Extending QGroundControl for automated mission planning of UAVs. Sensors, 18(7): 2339. <https://doi.org/10.3390/s18072339>
- [30] Kuiper, E., Nadjm-Tehrani, S. (2006). Mobility models for UAV group reconnaissance applications. International Conference on Wireless and Mobile Communications, Bucharest, pp. 30-33. <https://doi.org/10.1109/ICWMC.2006.63>
- [31] Wang, W., Guan, X., Wang, B., Wang, Y. (2010). A novel mobility model based on semi-random circular movement in mobile ad hoc networks. Information Sciences, 180(3): 399-413. <https://doi.org/10.1016/j.ins.2009.10.001>
- [32] Hossain, S., Deok-jin, L. (2019). Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices. Sensors, 19(5): 3371. <https://doi.org/10.3390/s19153371>
- [33] Kuiper, E., Nadjm-Tehrani, S. (2011). Geographical routing with location service in intermittently connected MANETs. IEEE Transactions on Vehicular Technology, 60(2): 592-604. <https://doi.org/10.1109/TVT.2010.2091658>
- [34] Wang, W., Guan, X., Wang, B., Wang, Y. (2010). A novel mobility model based on semi-random circular movement in mobile ad hoc networks. Information Sciences, 180(3): 399-413. <https://doi.org/10.1016/j.ins.2009.10.001>
- [35] Hossain, S., Lee, D. (2019). Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices. Sensors, 19(5): 3371. <https://doi.org/10.3390/s19153371>