Article

A Smart Contract-Based Algorithm for Offline UAV Task Collaboration: A New Solution for Managing Communication Interruptions

Linchao Zhang *, Lei Hang, Keke Zu, Yi Wang

Posted Date: 21 November 2024

doi: 10.20944/preprints202411.1684.v1

Keywords: Blockchain; Smart Contract; UAV Swarm; Collaborative Decision-Making; Offline Task Coordination

Article

A Smart Contract-Based Algorithm for Offline UAV Task Collaboration: A New Solution for Managing Communication Interruptions

Linchao Zhang [1,2,*,†,‡], **Lei Hang** [3,‡] , **Keke Zu** [2,‡] **and Yi Wang** [2,‡]

[1] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2] Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou 324003, China
[3] Department of Fintech, Shanghai Normal University Tianhua College, Shanghai 201815, China
[*] Correspondence: zhanglinchao@csj.uestc.edu.cn
[†] Current address: Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou 324003, China.
[‡] These authors contributed equally to this work.

Abstract: Environmental factors and electronic interference often disrupt communication between UAV swarms and ground control centers, requiring UAVs to complete missions autonomously in offline conditions. However, current coordination schemes for UAV swarms heavily depend on ground control, lacking robust mechanisms for offline task allocation and coordination, which compromises efficiency and security in disconnected settings. This limitation is especially critical for complex missions, such as rescue or attack operations, underscoring the need for a solution that ensures both mission continuity and communication security. To address these challenges, this paper proposes an offline task coordination algorithm based on blockchain smart contracts. This algorithm integrates task allocation, resource scheduling, and coordination strategies directly into smart contracts, allowing UAV swarms to autonomously make decisions and coordinate tasks while offline. Experimental simulations confirm that the proposed algorithm effectively coordinates tasks and maintains communication security in offline states, significantly enhancing the swarm's autonomous performance in complex, dynamic scenarios.

Keywords: blockchain; smart contract; UAV swarm; collaborative decision-making; offline task coordination

1. Introduction

With the rapid advancement of UAV technology, UAV swarm systems are being increasingly deployed in diverse mission scenarios, such as disaster rescue, logistics, and military operations, due to their efficiency and flexibility [1]. In particular, it has shown significant advantages in flexible response in complex environments. However, environmental factors and electronic interference often interfere with the communication between the drone cluster and the ground control center, causing communication interruption. At this time, the drone needs to complete the task without central control [2]. This places extremely high demands on the autonomous and collaborative capabilities of the drone. Therefore, it is crucial for drone swarms to have efficient offline autonomous decision-making and collaborative capabilities in complex environments. Most current research relies on ground control support and focuses on UAV collaboration under continuous communication, lacking systematic exploration of task allocation and coordination mechanisms in offline environments. This limitation compromises task efficiency and communication security, particularly when UAV swarms perform complex missions in disconnected settings [3].

In recent years, research has increasingly explored the use of blockchain and smart contracts to address the need for autonomous task execution in offline UAV (Unmanned Aerial Vehicle) systems [4]. Blockchain, as a decentralized and tamper-resistant ledger technology, offers a secure environment for data exchange within UAV swarms, ensuring mission data is transmitted and stored safely and effectively mitigating data security risks in offline scenarios. In such settings, traditional centralized control mechanisms often struggle with information flow and task continuity due to a lack of real-time communication and remote support. By contrast, the censorship-resistant, distributed nature of blockchain provides a more resilient operational foundation for UAV swarms. The timestamping capability of blockchain enables each task step to be precisely recorded, protecting against potential data tampering. Furthermore, blockchain's immutability and transparency allow all mission-related information and decisions to be verified independently by each node, enhancing transparency and trustworthiness throughout mission execution.

Smart contracts further enable automation of predefined coordination rules, supporting key processes for UAV swarms operating offline [5]. Unlike traditional manual intervention or centralized control, smart contracts ensure that the rules and conditions set during mission planning are strictly followed, facilitating automated task response. This is particularly valuable in scenarios requiring UAV swarms to make autonomous decisions and respond immediately to emergencies. For instance, tasks like route adjustments and dynamic resource allocation can be handled swiftly through the self-executing nature of smart contracts, ensuring efficient task completion.

However, despite these advantages, current research has not yet fully exploited the potential of blockchain smart contracts to enhance collaborative decision-making within UAV swarms in offline conditions. Particularly in complex and dynamic task environments, maintaining coherent decision-making and secure communication offline remains a pressing challenge. Current solutions primarily apply blockchain and smart contracts to basic data exchange and simple task scheduling, without leveraging their potential in complex decision-making, such as facilitating efficient multi-UAV collaboration and flexible adaptation without real-time communication. Therefore, developing a method that fully utilizes blockchain to enhance task continuity and communication security in offline settings is essential. This approach would improve UAV systems' autonomy and collaboration during missions and support their intelligent, autonomous development for more demanding scenarios, equipping UAV systems with adaptive capabilities for dynamic environments and ensuring stable, efficient operation under varied mission conditions.

Therefore, the main contributions of this paper are as follows:

1. **Development of a Blockchain-Based Autonomous Collaboration Algorithm for Offline UAV Swarms**: To improve task execution efficiency of UAV swarms in offline environments, this paper presents an autonomous collaboration mechanism, including algorithms for task allocation, resource scheduling, and collaborative decision-making. These algorithms are embedded within smart contracts to enable autonomous task coordination and efficient execution under offline conditions.
2. **Design of an Automatic Smart Contract Triggering Mechanism for Unforeseen Offline States**: Addressing task interruptions and communication failures in offline UAV swarm environments, this paper introduces an automatic triggering mechanism within smart contracts. This mechanism initiates the autonomous collaboration processes based on predefined conditions and rules, ensuring uninterrupted task execution even in offline states.
3. **Performance and Robustness Validation of the Algorithm in Offline Environments**: Through simulation experiments, this paper validates the algorithm's effectiveness in enhancing task continuity and communication security in offline conditions, demonstrating its applicability across various offline mission scenarios.

The remainder of this paper is organized as follows: Section 2 reviews related research on offline coordination in UAV swarms. Section 3 presents the blockchain-based UAV network architecture, and Section 4 details the design of the proposed autonomous task coordination algorithm based on blockchain smart contracts. Section 5 describes the simulation and performance evaluation of the algorithm. Finally, Section 7 concludes the paper and provides suggestions for future research directions.

2. Related Work

2.1. *Challenges in Offline Communication for UAV Swarms*

Due to their flexibility, mobility, and cost-effectiveness, UAV swarms have seen widespread application in areas such as military operations, disaster response, and logistics [6]. However, their deployment presents significant communication challenges, particularly in establishing high-speed communication links and developing efficient collaborative decision-making algorithms [7]. Achieving decentralized communication among UAVs requires the integration of sensor arrays and advanced routing algorithms to enable autonomous operation without relying on a central base station, thus ensuring quality of service (QoS) [8]. Continuous communication is essential for task coordination, especially in critical scenarios like disaster relief and military operations [9]. Studies have shown that communication failures often lead to mission failure, whereas effective information sharing enhances collaboration and conflict resolution in multi-agent systems, ultimately improving decision-making and operational efficiency [10–12].

Despite advancements in UAV communication and collaboration technology, several challenges persist. UAVs need to establish high-speed links, ensure secure data transmission, and maintain reliable control strategies [13]. Additionally, UAVs are prone to interference and have limited payload capacity, which further complicates integration with existing communication infrastructure [14]. UAV communication networks (FANETs) face numerous issues in security, resource management, and physical constraints, exacerbated by the dynamic nature of UAV operations [15,16]. Secure communication is especially critical; robust encryption and authentication mechanisms are essential to defend against potential attacks [17–19]. In multi-UAV systems, collaborative communication is vital for mission success, yet achieving intelligent UAVs with autonomous coordination capabilities remains technically challenging [20,21].

The challenges become even more pronounced when UAVs lose communication with the ground control center or other UAVs. Electromagnetic interference (EMI) often leads to communication instability in air-to-ground applications. Additionally, given the small size, limited battery life, and high-speed maneuverability of UAVs, maintaining reliable communication in complex environments is even more difficult [22]. Multi-UAV collaborative communication, forming an “aerial network,” is seen as a crucial approach to improving communication reliability [23]. As shown in Table 1, designing network topologies suited for UAV swarms is essential to ensure the safe and efficient operation of UAVs when communication interruptions occur.

Table 1. Drone Network Comparison.

Characteristic	Mesh Network	FANET	MANET	Integrated Reputation Consensus (FANET/MANET)
Self-organization Capability	High	High	High	High
Multi-hop Communication	Supported	Supported	Supported	High
Node Topology	Static or slow-changing	Highly dynamic	Moderate dynamics	FANET High
Latency Requirements	Moderate	Low-latency	Low to moderate	FANET High
Energy Constraints	Unrestricted	Battery-limited	Restricted	Moderate
Application Scenarios	HHome, urban	Military, disaster, monitoring	Military, disaster, transport	FANET High

2.2. *Limitations of Centralized Control Systems in Dynamic Environments*

In dynamic environments, centralized control systems face significant limitations, primarily due to their reliance on pre-programmed decision logic that lacks the flexibility to adapt to unpredictable and complex scenarios. This inherent inflexibility makes centralized systems slow to respond to rapid environmental changes and uncertainties, often resulting in inefficient task execution or even mission failure [24,25]. This challenge is especially critical in Unmanned Aerial Systems (UAS), where real-time responsiveness and high adaptability are essential.

Given these challenges, decentralized and autonomous control methods present a more promising alternative. By distributing decision-making authority across multiple agents, each agent can independently adjust its behavior based on local information without relying on a central command, enhancing the system's flexibility and scalability [26–28]. In the field of Unmanned Aerial Vehicles (UAVs), decentralized control methods have been successfully applied in navigation and formation control [29,30]. For instance, techniques such as virtual spring-damper models and rotational force fields facilitate autonomous navigation and obstacle avoidance in complex environments, allowing UAV swarms to operate effectively even during communication outages [31].

Furthermore, decentralized control also offers advantages in security management. Issues like electromagnetic interference and data encryption are more effectively managed in a distributed framework, underscoring the limitations of centralized control systems.

2.3. *Autonomy and Collaboration in Offline UAV*

The autonomy and collaboration capabilities of offline UAVs are crucial for executing complex multi-UAV tasks with greater efficiency and effectiveness. Integrating game-theoretic methods and multi-agent systems plays a central role in achieving these capabilities. For example, game-theoretic learning frameworks enable dynamic, decentralized task planning, allowing UAVs to autonomously coordinate and allocate tasks efficiently, as demonstrated in multi-UAV inspection tasks [32,33]. Similarly, Multi-Agent Reinforcement Learning (MARL) addresses collaborative task execution challenges in UAV networks, focusing on energy efficiency and trajectory optimization to ensure task completion even in dynamic environments [34]. Distributed or decoupled collaborative autonomy frameworks (D2CAF) further support multi-agent task allocation solutions, enabling communication and task execution among physically disconnected agents [35].

Moreover, integrating ground and aerial robots in disaster response highlights the complementary capabilities of robotic systems, where UAVs provide rapid environmental mapping while ground robots conduct detailed tasks based on gathered data [36]. Autonomous collaborative decision-making systems, which incorporate threat assessment, task planning, and negotiation modules, enhance the capacity of UAV swarms to operate independently and make real-time decisions [37]. Together, these advancements underscore the importance of autonomy and collaboration in offline UAV operations, allowing UAVs to undertake complex tasks with minimal human intervention and maximize task efficiency through intelligent coordination and resource management.

2.4. *Application of Blockchain Technology in UAV*

Blockchain technology is increasingly integrated into UAV systems to address challenges related to security, data integrity, and operational efficiency. One major application is enhancing the security and verifiability of UAV surveillance systems. By leveraging blockchain's decentralized and tamper-resistant properties, UAVs can ensure data reliability and trustworthiness, with smart contracts managing data access and privacy [38]. In the context of Industry 5.0, blockchain frameworks are transforming UAV communication by providing secure and efficient channels essential for human-machine collaboration in industrial applications [39].

Blockchain also supports federated learning initiatives by transferring training tasks to local servers and using new consensus algorithms like Proof-of-Energy to manage UAV participation, thereby safeguarding data and protecting user privacy [40]. Additionally, IoT and blockchain integration improves UAV operational efficiency by increasing data transmission rates and reducing energy consumption [41]. Privacy-preserving blockchain authentication frameworks are under development to secure UAV operations by providing unique digital identities and processing authentication requests through smart contracts, enhancing resilience against unauthorized access and attacks [42].

In multi-UAV networks, blockchain-assisted zero-trust authentication schemes like BAZAM ensure continuous UAV identity verification, preventing data breaches and system failures [43]. Decentralized PKI frameworks, based on blockchain, eliminate single points of failure in UAV communication systems, improving the security and reliability of spatial crowdsourcing services [44]. For UAV swarms, dynamic consensus algorithms like DTPBFT enhance scalability and fault tolerance, enabling efficient management of large-scale UAV networks [45]. Lightweight blockchain solutions are also being developed for UAV ad hoc networks to support secure collaborative computation and efficient task allocation [46]. In smart agriculture, blockchain integration within UAV networks strengthens data integrity and confidentiality, ensuring secure communication and reducing the risk of unauthorized data access [47]. Overall, blockchain technology has proven to be essential for enhancing the security, efficiency, and reliability of UAV systems across various applications.

3. Blockchain-Based UAV Networks

This paper addresses the challenges of autonomous task execution and secure communication for UAV swarms operating in offline environments, focusing on two scenarios: one in which the entire UAV swarm loses connection with the ground station, and another where an individual UAV disconnects from the swarm. To tackle these issues, a blockchain-based network architecture for UAVs is proposed. This architecture integrates FANET (Flying Ad Hoc Network) with a reputation-based consensus algorithm, enabling efficient data transmission and a reliable consensus mechanism. As a result, the UAV swarm gains improved autonomous collaboration and secure communication capabilities even in the absence of ground control. The proposed architecture integrates hardware and software layers to create a highly autonomous and secure UAV system by leveraging blockchain technology and smart contracts.

On the hardware level, each UAV is equipped with an advanced onboard computing unit that manages flight control, data processing, and blockchain operations. Each UAV also includes a wireless communication module supporting the FANET protocol, allowing ad-hoc network connections between drones. Additionally, UAVs are outfitted with various sensors and actuators—including GPS, IMU, and cameras—to enable environmental awareness and task execution. An energy management system further supports sustained operation for high-power consumption tasks.

The system architecture, illustrated in Figure 1, comprises two main layers at the network level: the communication network layer and the blockchain network layer. In the communication network layer, the UAV swarm utilizes FANET's mesh-based multi-hop communication for direct inter-node communication and information sharing. This flexible topology allows the network to sustain connectivity amidst frequent node additions or departures, which supports reliable information exchange for tasks in complex, dynamic environments.

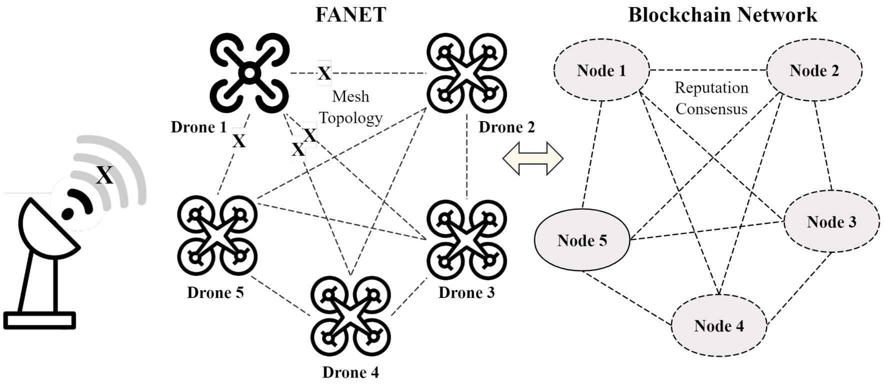


Figure 1. Offline UAV Network Architecture.

On the blockchain network layer, the system integrates a distributed ledger, smart contract modules, and a reputation-based consensus mechanism. Each UAV node maintains a partial copy of the blockchain and utilizes smart contracts for automating task allocation, coordination strategies, and resource management. The reputation-based consensus algorithm is tightly integrated with the FANET network, safeguarding data security and consistency throughout task distribution and execution.

This decentralized design empowers the UAV swarm to maintain autonomous coordination and task scheduling via smart contracts, even in the absence of a ground control center.

To enhance collaborative capabilities in resource-limited and dynamic environments, the blockchain network architecture emphasizes efficient resource use and streamlined collaboration. Given the UAV nodes' limited computational power and battery constraints, the blockchain system employs lightweight nodes that store only essential task-related data and block headers. The reputation-based consensus protocol improves efficiency by scoring nodes based on their resource status, reducing unnecessary communication load [48]. Data pruning techniques further reduce storage requirements by deleting non-essential historical data, retaining only critical information. These optimizations substantially decrease storage and computational overhead, enhancing overall task execution efficiency.

3.1. Smart Contract Mechanism Design

The smart contract execution mechanism is crucial for achieving autonomous collaboration within UAV swarms. UAV nodes initiate task execution according to preset consensus rules. To support autonomous collaboration across tasks, the smart contracts offer various templates designed for different mission types, including patrol, search and rescue, supply transport, and attack operations. For example, in search and rescue missions, the contract outlines the UAV's area search patterns, target identification techniques, and coordinated rescue steps upon locating a target. Collaboration strategies also include contingency protocols for UAV disconnection or failure, ensuring mission continuity. Each smart contract template is designed as a minimal functional unit, which can be assembled into sets of contracts as needed. This modular approach supports inter-contract activation and invocation, enhancing both task flexibility and execution efficiency.

As shown in Figure 2, this paper adopts a smart contract deployment strategy that combines full replication and partial replication. In the full replication mode, all UAV nodes store essential smart contracts (e.g., Node 1) to ensure that any node can independently execute core tasks. In partial replication mode, certain contracts are stored only on selected nodes (e.g., Nodes 2, 3, and 4) based on each UAV's resources and assigned roles, conserving storage space. Each smart contract template clearly specifies task objectives, resource requirements, execution procedures, and coordination strategies (see algorithm in Section 4.3).

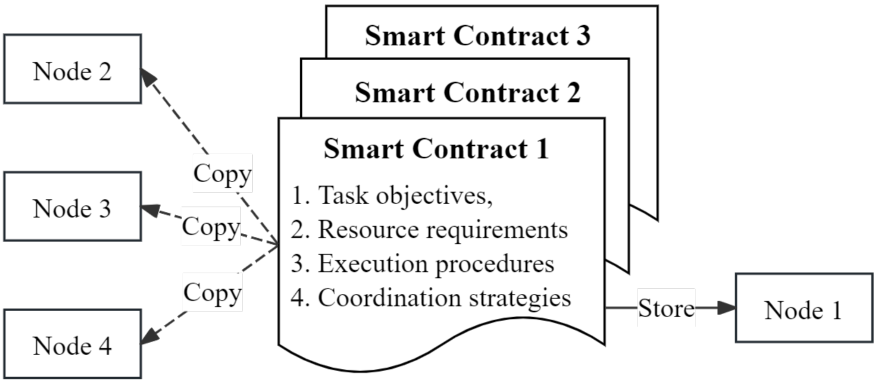


Figure 2. Smart Contract Deployment Methods.

In offline conditions, smart contract execution depends on the UAVs' autonomous decision-making capabilities and the FANET network. FANET allows direct communication between UAVs, so even if some nodes are disconnected, other nodes can maintain network-wide connectivity through multi-hop communication. This environment supports the continued effectiveness of the reputation-based consensus algorithm, ensuring that smart contract execution results are consistently agreed upon across the network.

Moreover, smart contract execution is designed with integrated security and reliability features. During contract execution, UAVs use encryption and authentication mechanisms to ensure only authorized nodes can participate in specific tasks. The system's redundancy and fault tolerance capabilities further enhance the stability of the UAV swarm, enabling it to withstand single or multiple node failures.

4. Design of Autonomous Collaborative Algorithms Based on Smart Contracts

Offline autonomous task execution by drone swarms in real-world applications involves addressing a range of complex requirements and challenges. Operating in highly dynamic and unpredictable environments, drone swarms must exhibit high levels of autonomy and collaboration, particularly when communication with ground control centers is unavailable. In these offline conditions, drones need to manage a variety of tasks effectively, including efficient task allocation, reliable data sharing, stable communication, and robust collaboration mechanisms. To meet these demands, this paper presents an autonomous collaboration algorithm that efficiently addresses task allocation, resource scheduling, and collaborative decision-making complexities. The algorithm is designed to reduce energy consumption, ensure efficient task execution, enhance swarm collaboration in complex environments, and minimize the computational load on smart contracts.

As illustrated in Figure 3, the autonomous collaboration algorithm includes three sub-algorithms, which operate in the following coordinated sequence:

- Step 1: When tasks are registered and added to the task pool, the task allocation algorithm immediately assesses all available drone nodes based on resource status (e.g., battery level, current task load) and selects the highest-scoring node to execute the task.
- Step 2: If the designated node fails during task execution, the resource scheduling algorithm reallocates the task to a suitable alternative node (Step 5).
- Step 3: For tasks requiring multiple nodes, the collaborative decision-making algorithm selects an optimal group of nodes to execute the task jointly (Step 7).
- Step 4: If any node in this group fails, the resource scheduling algorithm replaces it with a new node (Step 6) to maintain uninterrupted task execution (Step 7).

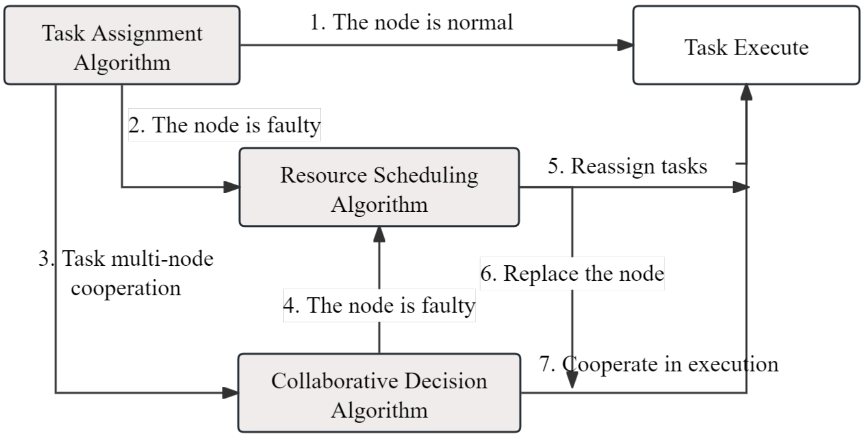

Task Assignment Algorithm
1. The node is normal
Task Execute
2. The node is faulty
Resource Scheduling Algorithm
5. Reassign tasks
3. Task multi-node cooperation
6. Replace the node
4. The node is faulty
7. Cooperate in execution
Collaborative Decision Algorithm

Figure 3. The collaborative process of autonomous collaborative algorithms.

To implement this algorithm effectively, key performance metrics are defined for both drone nodes and task requirements. As show Tables 2 and 3:

1. UAV Node Set: Let there be N UAV nodes, denoted as the set $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$.
2. Task Set: Let there be M tasks, denoted as the set $\mathcal{T} = \{t_1, t_2, \ldots, t_M\}$.

Table 2. Resource status metrics for each UAV node n_i.

Metric	Description	Normalized
E_i	UAV Remaining Energy	E_i^{norm}
L_i	UAV Task Load	L_i^{norm}
C_i	UAV Computational Power	C_i^{norm}
B_i	UAV Communication Bandwidth	B_i^{norm}
ΔT_i	UAV Communication Delay	ΔT_i^{norm}

Table 3. Task requirements for each metric t_j.

Metric	Task Description	Range
P_j	Task Priority	$[P_{\min}, P_{\max}]$
U_j	Task Urgency	$[U_{\min}, U_{\max}]$
C_j^{req}	Required Computational Power	-
B_j^{req}	Task Required Bandwidth	-
E_j^{req}	Task Required Energy	-
T_j^{ddl}	Task Deadline	-

4.1. *Task Allocation Algorithm*

The task allocation algorithm initiates the autonomous collaboration process and is critical to efficient task execution within the UAV swarm. By scoring each node's resource status, the algorithm identifies the most suitable node to execute a given task, optimizing resource utilization and aligning node selection with task requirements.

The task allocation algorithm operates as outlined in Figure 4:

1. Information Collection: Each UAV node collects its resource status information, while the smart contract identifies task requirements.
2. Resource Scoring: Each node's metrics are normalized, and a scoring function calculates its resource status score.
3. Optimal Node Selection: Based on resource scores, an initial set of qualified UAV nodes is selected, from which the optimal node is chosen. To prevent overload, a dynamic load adjustment mechanism recalculates scores based on each node's current load, balancing task allocation. The final optimal node is determined through an optimization model.
4. Task Assignment Execution: The optimal node is then assigned the task.
5. Task Allocation Consensus: Upon successful task allocation, a consensus is recorded on the blockchain to confirm the assignment's validity and the node's acceptance of the task.

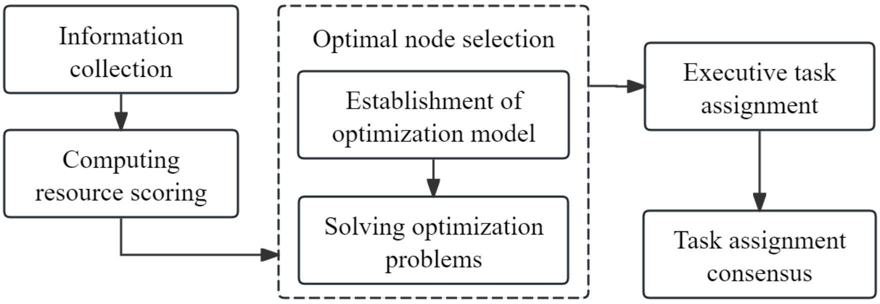

Information collection
Computing resource scoring
Optimal node selection
Establishment of optimization model
Solving optimization problems
Executive task assignment
Task assignment consensus

Figure 4. Design of the Task Allocation Algorithm.

4.1.1. Resource Status Scoring

This paper utilizes a reputation-based consensus algorithm to identify UAV nodes with high scores (indicating strong resource availability) for task execution. The consensus threshold can be adjusted to facilitate rapid selection of high-scoring nodes. A UAV node's score is calculated using five key metrics: remaining energy, task load, computational power, communication bandwidth, and communication delay. Thus, nodes with higher scores indicate greater available resources.

As shown in Table 2, the scoring process consists of two stages. First, normalization adjusts for differences in units and value ranges across metrics. Then, a resource status scoring function, S_i, evaluates each node's suitability by computing a weighted sum of the normalized metrics:

$$S_i = w_1 E_i^{\text{norm}} + w_2 L_i^{\text{norm}} + w_3 C_i^{\text{norm}} + w_4 B_i^{\text{norm}} + w_5 \Delta T_i^{\text{norm}} \tag{1}$$

where the weight coefficients $w_k = \{w_1, w_2, w_3, w_4, w_5\}$ satisfy:

$$\sum_k w_k = 1, \quad w_k \geq 0 \tag{2}$$

The weight coefficients can be dynamically adjusted based on task requirements:

- For urgent tasks, increase w_2 to prioritize nodes with lower load.
- For compute-intensive tasks, increase w_3 to prioritize nodes with higher computational power.
- For tasks requiring high reliability, increase w_5.

4.1.2. Optimal Node Selection

The smart contract selects the UAV node with the highest resource score to execute the task, provided it meets the task's requirements. This method maximizes the efficient use of UAV resources. Once the task is assigned, a consensus is reached within the UAV swarm, ensuring consistent understanding among nodes and minimizing potential miscommunication during execution.

For a given task t_j, the goal is to select a UAV that meets the task requirements. Specifically, we aim to find a node n_{i^*} within the feasible set $\mathcal{N}_j^{\text{feasible}}$ that maximizes the scoring function S_{ij}. Thus, n_{i^*} is the node with the highest score S_{ij} in the feasible set, defined as:

$$n_{i^*} = \arg \max_{n_i \in \mathcal{N}_j^{\text{feasible}}} S_{ij} \tag{3}$$

The feasible node set $\mathcal{N}_j^{\text{feasible}}$, which meets the task requirements, is defined as:

$$\mathcal{N}_j^{\text{feasible}} = \left\{ n_i \,\middle|\, \begin{array}{l} E_i \geq E_j^{\text{req}} \\ C_i \geq C_j^{\text{req}} \\ B_i \geq B_j^{\text{req}} \\ T_{ij}^{\text{est}} \leq T_j^{\text{ddl}} \end{array} \right\} \tag{4}$$

Where, E_j^{req}, C_j^{req}, and B_j^{req} represent the minimum requirements for remaining energy, computational power, and communication bandwidth, respectively, to complete task t_j. T_{ij}^{est} is the estimated time for node n_i to complete the task, and T_j^{ddl} is the task deadline. The estimated completion time T_{ij}^{est} is calculated as:

$$T_{ij}^{\text{est}} = \frac{D_{ij}}{V_i} + \frac{W_j}{C_i} + \Delta T_i + f(E_i) \tag{5}$$

Where, $\frac{D_{ij}}{V_i}$ denotes the estimated flight time for node n_i to reach the task location, $\frac{W_j}{C_i}$ represents the estimated execution time for task t_j, W_j is the workload of t_j, and $f(E_i)$ captures the effect of energy consumption on execution time.

4.1.3. Multi-Objective Constraints for Task Allocation

In multi-task, multi-node scenarios, task allocation becomes an optimization problem aimed at maximizing overall task execution utility. The utility function U is defined as the weighted sum of resource status scores and task priorities for all task-node combinations:

$$U = \sum_{j=1}^{M} \sum_{n_i \in \mathcal{N}_j^{\text{feasible}}} x_{ij} \left(S_i \cdot P_j \cdot U_j \right) \tag{6}$$

where x_{ij} is a decision variable, with $x_{ij} = 1$ if node n_i is assigned task t_j, and $x_{ij} = 0$ otherwise. Here, P_j represents the priority of task t_j, and U_j represents its urgency.

To ensure feasible and efficient task allocation, the following constraints must be satisfied:

- A. Unique Task Assignment: Each task t_j is assigned to only one node n_i.
- B. Node Resource Limits: Node resource limits, including computational power, communication bandwidth, and remaining energy, must not be exceeded.
- C. Task Deadline with Penalty: If the estimated completion time T_{ij}^{est} exceeds the task deadline T_j^{ddl}, a penalty is applied to the node's score S_i, reducing its likelihood of selection. This mechanism ensures timely completion of high-priority tasks.
- D. Decision Variable Restriction: x_{ij} is constrained to $x_{ij} \in \{0, 1\}$.

$$\text{Constraints:} \begin{cases} A: & \sum_{n_i \in \mathcal{N}_j^{\text{feasible}}} x_{ij} = 1, \quad \forall t_j \in \mathcal{T} \\ & \quad \sum_{t_j \in \mathcal{T}} x_{ij} \cdot C_j^{\text{req}} \leq C_i, \quad \forall n_i \in \mathcal{N} \\ B: & \quad \sum_{t_j \in \mathcal{T}} x_{ij} \cdot B_j^{\text{req}} \leq B_i, \quad \forall n_i \in \mathcal{N} \\ & \quad \sum_{t_j \in \mathcal{T}} x_{ij} \cdot E_j^{\text{req}} \leq E_i, \quad \forall n_i \in \mathcal{N} \\ C: & x_{ij} \cdot T_{ij}^{\text{est}} \leq T_j^{\text{ddl}}, \quad \forall n_i \in \mathcal{N}, \forall t_j \in \mathcal{T} \\ D: & x_{ij} \in \{0, 1\} \end{cases} \tag{7}$$

where C_j^{req} is the computational requirement of task t_j, and C_i is the available computational power of node n_i; B_j^{req} is the communication bandwidth requirement of task t_j, and B_i is the available bandwidth of node n_i; E_j^{req} is the energy requirement of task t_j, and E_i is the remaining energy of node n_i. The estimated completion time T_{ij}^{est} for node n_i to execute task t_j must meet the deadline T_j^{ddl}.

In summary, the optimization objective for task allocation is $\max_{x_{ij}} U$. This process constitutes an Integer Linear Programming (ILP) problem, as the decision variable x_{ij} is binary and both the objective function and constraints are linear. By incorporating task priority P_j, urgency U_j, and node resource status S_i, the algorithm prioritizes high-priority, urgent tasks, ensuring they are allocated to the most suitable nodes.

4.2. Resource Scheduling Algorithm

The resource scheduling algorithm manages task execution dynamically. While tasks are initially assigned to specific nodes, a node's resource status can change due to factors like rapid battery depletion, task load increases beyond capacity, or external changes (e.g., signal fluctuations). In such cases, the resource scheduling mechanism assesses real-time node status and reallocates resources as needed to maintain task continuity and stability.

As illustrated in Figure 5, the resource scheduling algorithm follows these steps:

1. Node Resource Change: When a node's resource status changes, the system evaluates its ability to continue the task, triggering adjustments or reallocation if necessary.
2. System State Monitoring: Nodes continuously monitor their resource statuses $E_i, L_i, C_i, B_i, \Delta T_i$ while the system periodically aggregates global parameters such as available node ratio ρ_N, task density λ_T, and averages of computational power $\overline{C}$, bandwidth $\overline{B}$, and delay $\overline{\Delta T}$.
3. Dynamic Threshold Update: Using current system parameters, dynamic thresholds E_{th} and L_{th} are calculated.

doi:10.20944/preprints202411.1684.v1

4. Node Status Assessment: Nodes compare their resource status with dynamic thresholds. If a node's status falls below the threshold, task reassignment is triggered; otherwise, the node continues executing the task.
5. Task Reallocation: For nodes that do not meet requirements ($\delta_i \geq 1$), the system identifies pending tasks $\mathcal{T}_i^{\text{pending}}$ and estimates remaining workload for reassignment.
6. New Node Selection: Pending tasks return to the task pool, where a new node n_k is selected based on resource status score S_k and optimization objectives.
7. Task Migration and Execution: The original node n_i transfers task data to the new node n_k, which begins execution. The system updates task allocation consensus.
8. System Status Update: Resource statuses and task allocations are updated, preparing for the next monitoring and adjustment cycle.

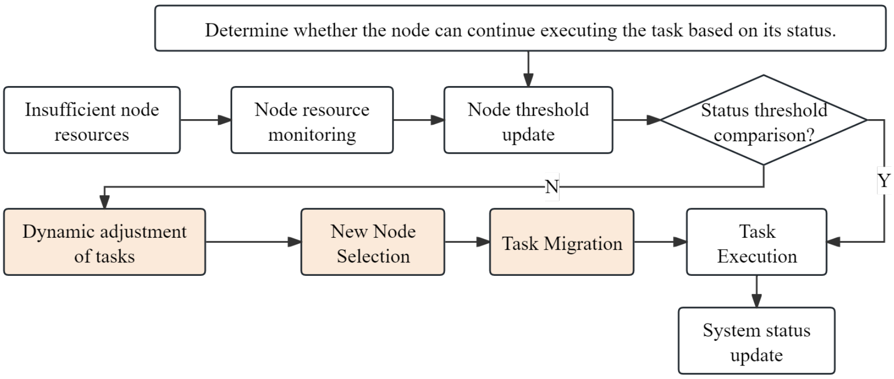

Determine whether the node can continue executing the task based on its status.
Insufficient node resources
Node resource monitoring
Node threshold update
Status threshold comparison?
N
Y
Dynamic adjustment of tasks
New Node Selection
Task Migration
Task Execution
System status update

Figure 5. Resource Scheduling Algorithm Process.

4.2.1. Global Dynamic Threshold Setting

To enable flexible resource management, dynamic thresholding based on the DVRC consensus algorithm adjusts thresholds according to the system's overall state. When task density is high or available nodes are limited, thresholds are lowered to expand the pool of nodes eligible for tasks. Conversely, when task density is low or node availability is high, thresholds are raised to ensure optimal allocation. This dynamic threshold mechanism improves resource scheduling efficiency and reduces computational overhead.

Key parameters used for dynamic threshold setting include available node ratio ρ_N and task density λ_T:

$$\rho_N = \frac{N_{\text{avail}}}{N_{\text{total}}} \quad \text{and} \quad \lambda_T = \frac{M_{\text{pending}}}{N_{\text{avail}}} \tag{8}$$

where N_{avail} represents available UAV nodes (those not overloaded and with sufficient resources), N_{total} is the total number of nodes, and M_{pending} is the number of pending tasks, reflecting resource demand relative to availability.

For adaptive management, thresholds for key resources, such as battery level E_i and task load L_i, adjust dynamically based on system parameters. Average communication delay $\overline{\Delta T}$, computational power $\overline{C}$, and bandwidth $\overline{B}$ are also considered for more accurate UAV node selection.

$$\begin{cases} \overline{C} &= \frac{1}{N_{\text{avail}}} \sum_{n_i \in \mathcal{N}_{\text{avail}}} C_i \\ \overline{B} &= \frac{1}{N_{\text{avail}}} \sum_{n_i \in \mathcal{N}_{\text{avail}}} B_i \\ \overline{\Delta T} &= \frac{1}{N_{\text{avail}}} \sum_{n_i \in \mathcal{N}_{\text{avail}}} \Delta T_i \end{cases} \tag{9}$$

Where, C_i, B_i, and ΔT_i represent the computational power, bandwidth, and delay metrics of node n_i, respectively.

The battery threshold E_{th} is defined as a function of ρ_N and λ_T:

$$\begin{aligned} E_{\text{th}} = E_{\min} &+ w_1(1-\rho_N) + w_2\lambda_T \\ &+ w_3(1-\overline{C}) + w_4(1-\overline{B}) \\ &+ w_5(1-\overline{\Delta T}) \cdot (E_{\max} - E_{\min}) \end{aligned} \tag{10}$$

where $E_{\min}$ and $E_{\max}$ are the minimum and maximum battery levels, respectively, and $w_1, w_2, w_3, w_4, w_5 \geq 0$ are weight coefficients reflecting the influence of available node ratio and task density.

Similarly, the task load threshold L_{th} is defined as:

$$\begin{aligned} L_{\text{th}} = L_{\max} &- w_1(1-\rho_N) + w_2\lambda_T \\ &+ w_3(1-\overline{C}) + w_4(1-\overline{B}) \\ &+ w_5(1-\overline{\Delta T}) \cdot (L_{\max} - L_{\min}) \end{aligned} \tag{11}$$

where $L_{\min}$ and $L_{\max}$ are the minimum and maximum task loads, respectively, with $w_1, w_2, w_3, w_4, w_5 \geq 0$ as weight coefficients for the task load threshold.

Threshold Adjustment Explanation:

1. When ρ_N decreases (indicating fewer available nodes) or λ_T increases, E_{th} is lowered, allowing nodes with lower battery levels to participate and expanding the resource pool.
2. An increase in L_{th} permits nodes with higher loads to continue accepting tasks, meeting high-demand scenarios.

This adaptive thresholding approach enhances resource allocation and optimizes UAV task management in dynamic environments.

4.2.2. Dynamic Task Adjustment Algorithm

During task execution, nodes continuously monitor their resource status. When a node's resources fall below specified thresholds, a smart contract automatically triggers task reallocation, redirecting incomplete tasks to other eligible nodes. This process ensures continuity and system robustness, preventing task failures due to resource depletion at individual nodes.

To determine if UAV nodes meet threshold requirements, a composite status function (including battery status and task load) is defined for each node n_i:

Battery Status Indicator δ_{E_i}:

$$\delta_{E_i} = \begin{cases} 1, & \text{if } E_i \leq E_{\text{th}} \\ 0, & \text{if } E_i > E_{\text{th}} \end{cases} \tag{12}$$

Task Load Status Indicator δ_{L_i}:

$$\delta_{L_i} = \begin{cases} 1, & \text{if } L_i \leq L_{\text{th}} \\ 0, & \text{if } L_i > L_{\text{th}} \end{cases} \tag{13}$$

Composite Status Indicator δ_i:

$$\delta_i = \delta_{E_i} + \delta_{L_i} \tag{14}$$

If $\delta_i \geq 1$, the resource status of node n_i falls below threshold, activating the dynamic task adjustment mechanism. The task adjustment strategy proceeds with the following steps:

1. Identify Uncompleted Tasks: Define $\mathcal{T}_i$ as the set of tasks currently assigned to node n_i. Identify the subset of uncompleted tasks, $\mathcal{T}_i^{\text{pending}}$.
2. Estimate Remaining Workload: For each uncompleted task $t_j \in \mathcal{T}_i^{\text{pending}}$, calculate the remaining workload W_{ij}^{rem}: $W_{ij}^{\text{rem}} = W_j^{\text{total}} - W_{ij}^{\text{completed}}$

3. where W_j^{total} is the total workload of task t_j, and $W_{ij}^{\text{completed}}$ is the workload completed by node n_i.
4. Trigger Task Reallocation: Reassign each uncompleted task $t_j \in \mathcal{T}_i^{\text{pending}}$ to the task pool, making it available for other qualified nodes.
5. Select New Node: Based on the resource status score S_{kj} , select the highest-scoring eligible node n_k for task t_j.

In reallocating tasks, it is crucial to minimize delay and resource consumption. When transferring task status and intermediate data from node n_i to n_k, the estimated task completion time T_{kj}^{est} for n_k combines task execution and data transfer times:

$$\min_{n_k} T_{kj}^{\text{est}} = \min_{n_k} \left(\frac{D_{ij}^{\text{data}}}{B_{ik}} + \frac{W_{ij}^{\text{rem}}}{C_k} \right) \tag{15}$$

where D_{ij}^{data} is the data volume (task status and intermediate results) to transfer, B_{ik} the communication bandwidth between n_i and n_k, W_{ij}^{rem} the remaining workload of task t_j, and C_k the computational power of n_k.

Similarly, to minimize resource consumption during task migration, the total energy consumption E_{total} includes energy for data transfer and task execution:

$$\min_{n_k} E_{\text{total}} = \min_{n_k} \left(P_{ik}^{\text{comm}} \cdot \frac{D_{ij}^{\text{data}}}{B_{ik}} + P_k^{\text{comp}} \cdot \frac{W_{ij}^{\text{rem}}}{C_k} \right) \tag{16}$$

where P_{ik}^{comm} is the communication power, P_k^{comp} the computational power consumption rate of the new node, $\frac{D_{ij}^{\text{data}}}{B_{ik}}$ the data transfer energy, and $\frac{W_{ij}^{\text{rem}}}{C_k}$ the task execution energy.

In practical applications, multiple objectives are optimized simultaneously to select an appropriate new node n_k for task reassignment. A comprehensive objective function, incorporating resource status score S_k (see Section 4.3.1), is formed with each objective combined in a weighted form:

$$\min_{n_k} U(n_k) = \min_{n_k} \left(\lambda_1 T_{kj}^{\text{est}} + \lambda_2 E_{\text{total}} - \lambda_3 S_k \right) \tag{17}$$

where $\lambda_1, \lambda_2, \lambda_3 \geq 0$ are weight coefficients indicating the importance of each objective. This composite objective seeks to minimize time and energy consumption while maximizing the resource status score, achieving optimal task reassignment under task allocation constraints (see equation 7).

4.3. Collaborative Decision-Making Algorithm

Collaborative decision-making is essential for effective multi-node coordination during task execution, especially in complex environments where multiple nodes must work together. While the task allocation phase assigns initial nodes and establishes mechanisms to handle potential failures, dynamic conditions often require multi-node collaboration. This study employs a multi-attribute decision-making (MADM) approach to identify optimal task plans and node combinations, as outlined below:

1. Information Collection: Nodes report their attribute metrics and reputation scores, while the smart contract provides task requirements and candidate plans.
2. Decision Matrix Construction: Attribute metrics for each node are collected, forming a decision matrix for each plan.
3. Weighted Standardization of the Decision Matrix: The decision matrix is standardized, then weighted according to attribute importance.
4. Identifying Ideal Solutions: Calculate the positive ideal solution (PIS) and negative ideal solution (NIS) for each attribute.

5. Relative Closeness Calculation: Compute each node's relative closeness to the ideal solutions, resulting in support scores for each plan.
6. Node Selection with Constraints: Based on support scores and constraints (e.g., resource limits, connectivity), select suitable nodes for task execution.
7. Optimal Plan Selection: Calculate each plan's total score and select the plan with the highest score.
8. Task Execution: Nodes execute tasks according to the assigned plan.
9. Feedback and Reputation Update: Upon task completion, update each node's reputation score.

4.3.1. Problem Definition

The goal of collaborative decision-making is to select the optimal plan o_j^* and the participating node set $\mathcal{N}_j$ for each task t_j, maximizing task execution efficiency. The UAV node set $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$ contains N nodes, and the task set $\mathcal{T} = \{t_1, t_2, \ldots, t_M\}$ contains M tasks. Each task t_j has a set of candidate plans $\mathcal{O}_j = \{o_{j1}, o_{j2}, \ldots, o_{jL}\}$.

4.3.2. Constructing the Initial Decision Matrix

For each candidate plan o_{jl} of task t_j, node performance is evaluated based on its attributes, forming the decision matrix X_{jl}, where rows represent nodes n_i and columns represent K attribute metrics. The matrix is structured as follows:

$$X_{jl} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NK} \end{bmatrix} \tag{18}$$

where each element x_{ik} represents the original value of the k-th attribute metric for node n_i.

4.3.3. Weighted Standardization of the Decision Matrix

To eliminate measurement scale differences, the decision matrix is standardized using a cost-based normalization method. The standardized decision matrix R_{jl} is:

$$R_{jl} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1K} \\ r_{21} & r_{22} & \cdots & r_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & \cdots & r_{NK} \end{bmatrix} \tag{19}$$

where r_{ik} is the standardized value of the k-th attribute for node n_i.

Next, the standardized decision matrix R_{jl} is multiplied by the weight vector $w = \{w_1, w_2, \ldots, w_k\}$, where weights are calculated based on attribute dispersion, resulting in the weighted standardized decision matrix V_{jl}:

$$V_{jl} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1K} \\ v_{21} & v_{22} & \cdots & v_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \cdots & v_{NK} \end{bmatrix} \tag{20}$$

Therefore, the relationship of the three matrices is:

- X_{jl}: Initial decision matrix containing the original attribute metrics of nodes.
- R_{jl}: Standardized decision matrix derived from X_{jl}.
- V_{jl}: Weighted standardized matrix, obtained by multiplying r_{ik} with weight w_k.

4.3.4. Calculating Positive and Negative Ideal Solutions

The ideal solutions identify the best and worst values for each attribute.

Positive Ideal Solution A^+ (Best Case): Composed of the maximum values for each attribute:

$$A^+ = \{v_1^+, v_2^+, \dots, v_K^+\}, \quad v_k^+ = \min_i v_{ik} \tag{21}$$

Negative Ideal Solution A^- (Worst Case): Composed of the minimum values for each attribute:

$$A^- = \{v_1^-, v_2^-, \dots, v_K^-\}, \quad v_k^- = \max_i v_{ik} \tag{22}$$

4.3.5. Calculating Node Relative Closeness

For each node n_i, calculate the Euclidean distances to the positive ideal solution S_i^+ and the negative ideal solution S_i^-. The support score V_{il} for candidate plan o_{jl} is defined as:

$$S_i^+ = \sqrt{\sum_{k=1}^{K} (v_{ik} - v_k^+)^2}, \quad S_i^- = \sqrt{\sum_{k=1}^{K} (v_{ik} - v_k^-)^2} \tag{23}$$

$$V_{il} = \frac{S_i^-}{S_i^+ + S_i^-}, \quad 0 \leq V_{il} \leq 1 \tag{24}$$

A larger V_{il} indicates closer proximity to the positive ideal solution, signifying stronger support for the plan.

4.3.6. Node Selection and Constraints

Select nodes based on support scores and constraints to participate in task execution. Rank nodes by their relative closeness C_i, and select those with higher scores, ensuring the task's required number of collaborative nodes is met: $K_j^{\min} \leq |\mathcal{N}_j| \leq K_j^{\max}$.

Verify that selected nodes meet resource constraints and communication connectivity requirements. Define a connectivity matrix A_{ik}, where $A_{ik} = 1$ if nodes n_i and n_k can communicate; otherwise, $A_{ik} = 0$. A cooperation variable y_{ijk} represents multi-node cooperation relationships:

$$\begin{aligned} y_{ijk} &\leq x_{ij} \quad &, y_{ijk} \leq x_{kj} \\ y_{ijk} &\leq A_{ik} \quad &, y_{ijk} \geq x_{ij} + x_{kj} + A_{ik} - 2 \end{aligned} \tag{25}$$

where $y_{ijk} \in \{0,1\}$ signifies whether nodes n_i and n_k are cooperating in task t_j, with $x_{ij} = 1$ indicating node selection and connectivity.

4.3.7. Optimal Plan Selection

Calculate each candidate plan's score by combining support scores V_{il} and weighted reputation scores S_i for all nodes:

$$V_l = \sum_{i=1}^{N} W_i V_{il} \tag{26}$$

The objective of collaborative decision-making is to select the highest-scoring plan as the optimal choice o_j^*:

$$o_j^* = \arg\max_{o_{jl} \in \mathcal{O}_j} V_l \tag{27}$$

In the optimal plan, each selected node must meet resource and task requirements, ensuring no node is overloaded and that all selected nodes are communicatively connected.

doi:10.20944/preprints202411.1684.v1

By solving this optimization model, the system identifies the optimal set of nodes for task collaboration. If a selected node encounters a failure, the resource scheduling algorithm reallocates the task to maintain continuity, followed by feedback and reputation updates.

Finally, the collaborative decision-making algorithm can be expressed as the following optimization problem:

$$\begin{aligned}
&\text{Choose the best solution: } o^* = \arg\max_{o_l \in \mathcal{O}} V_l \\
&\text{Where } V_l = \sum_{i=1}^{N} W_i V_{il} \\
&\text{Node support: } V_{il} = \frac{S_i^-}{S_i^+ + S_i^-} \\
&\text{Voting weight: } W_i = \frac{R_i}{\sum_{j=1}^{N} R_j} \\
&\text{Node selection satisfies:} \\
&\quad K_j^{\min} \le |\mathcal{N}_j| \le K_j^{\max}, \quad \forall j \\
&\quad E_i \ge E_j^{\text{req}} + E_{\text{reserve}}, \quad \forall i \\
&\quad C_i \ge C_j^{\text{req}}, \quad \forall i \\
&\quad B_i \ge B_j^{\text{req}}, \quad \forall i \\
&\quad y_{ijk} \le x_{ij}, \quad y_{ijk} \le x_{kj}, \quad y_{ijk} \le A_{ik}, \quad y_{ijk} \ge x_{ij} + x_{kj} + A_{ik} - 2 \\
&\quad x_{ij}, y_{ijk} \in \{0, 1\}, \quad \forall i, k, j
\end{aligned}$$

5. Simulation and Analysis

5.1. Algorithm Simulation

To evaluate the autonomous collaboration algorithm based on smart contracts, we deployed the algorithm within a blockchain-based smart contract environment. The simulation setup includes 10 UAV nodes and 5 tasks. Additional parameters are generated randomly within specified ranges, all of which comply with predefined constraints. This setup simulates real-time fluctuations in UAV node status and potential failures, as shown in Table 4.

Table 4. Simulation Parameters for UAV Nodes and Tasks

Metric	E_i	L_i	C_i	B_i	P_j	U_j	C_j^{req}	B_j^{req}	E_j^{req}
Parameter Range	[0,100]	[0,50]	[1,5]	[0,20]	[1,5]	[1,5]	[1,5]	[0,10]	[0,20]

As shown in Figure 6, the task allocation algorithm assigns tasks based on each node's reputation score S_i, giving priority to tasks with higher urgency and importance. In the simulation, node $n9$, with the highest reputation score ($S_i = 0.8$), is assigned to task $t4$, which has a priority P_j and urgency U_j of 3. This approach effectively directs high-priority tasks to nodes with the best reputation scores.

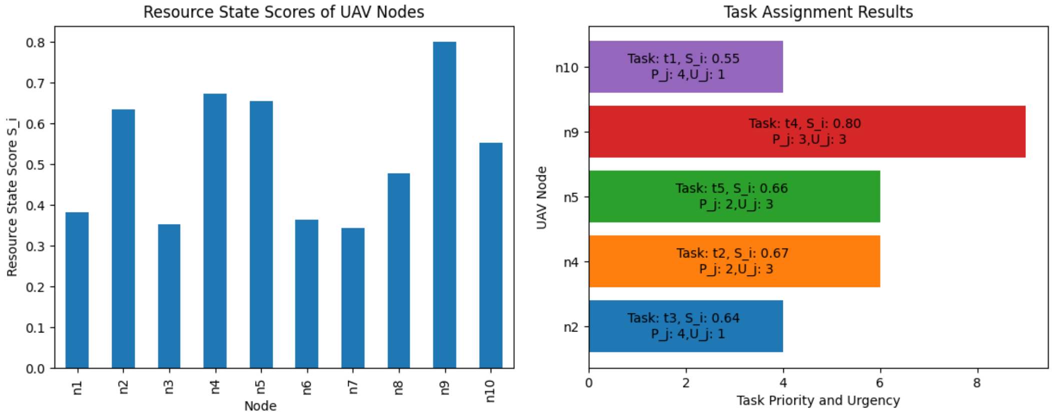


Figure 6. Simulation of task allocation algorithm.

Similarly, as shown in Figure 7, the resource scheduling algorithm continuously monitors each node's resource status after tasks are initially assigned. If a node becomes unable to continue due to a failure or lack of resources, the scheduling algorithm reassigns the task to a more suitable node. For instance, as node $n1$'s reputation score S_k declines and its task load L_i increases, it reaches the minimum resource threshold needed for task $t5$. This triggers the resource scheduling algorithm to reassign the task to node $n3$, which meets the resource constraints and successfully takes over the task during time steps [1-3].

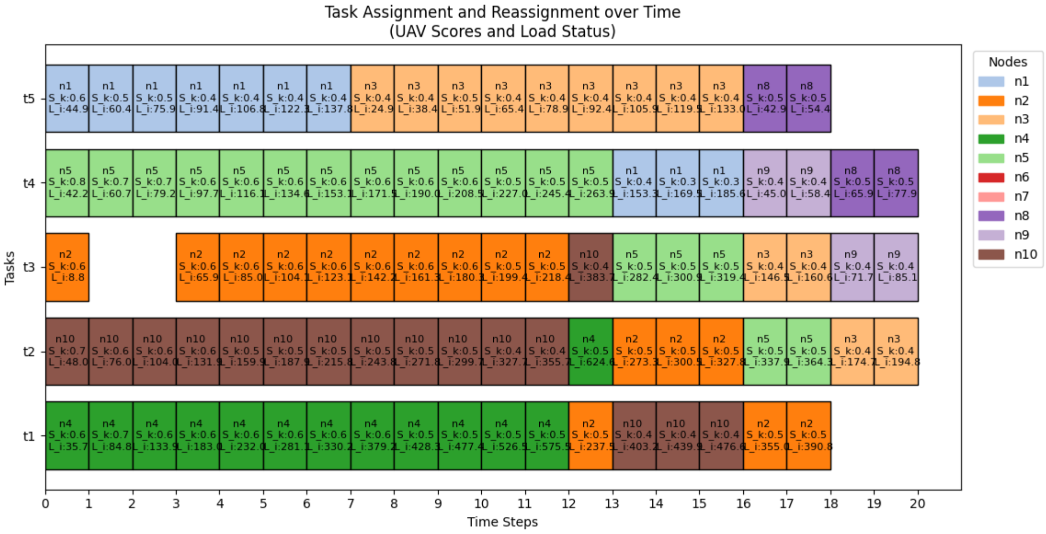


Figure 7. Resource Scheduling Algorithm Simulation.

The collaborative decision-making algorithm is illustrated in Figure 8. When tasks require collaboration from multiple UAVs, the algorithm dynamically forms a collaborative node set based on the current task requirements. To simulate the random occurrence of collaborative tasks, random numbers control their generation in the simulation; thus, only tasks $t2$ and $t5$ trigger collaboration. As time progresses, the number of collaborative nodes for task $t5$ increases. Notably, node $n4$ participates in tasks $t1$, $t2$, and $t5$. Due to variations in task priority and urgency, the resource scheduling algorithm continues to assign node $n4$ to different tasks as it completes each one, showing that node $n4$ has ample resources and remains within its operational threshold.

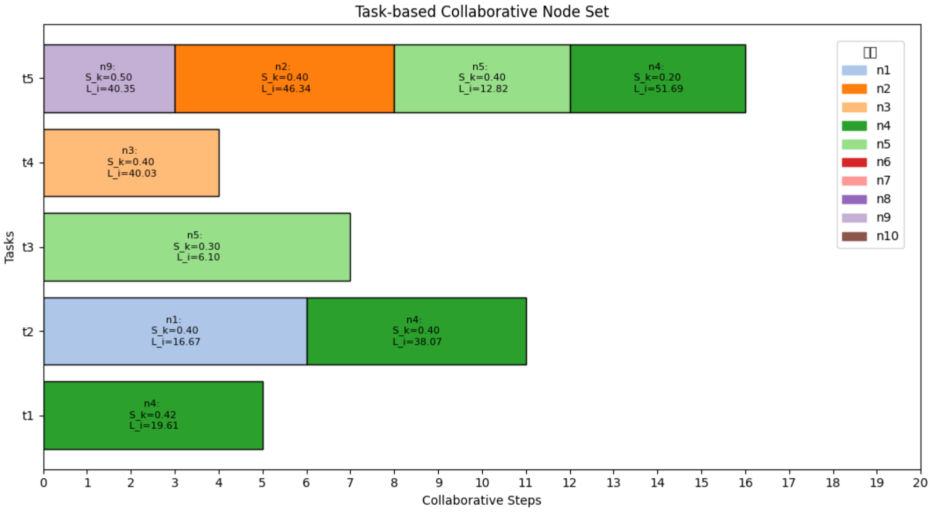


Figure 8. Collaborative decision-making algorithm simulation.

5.2. Algorithm Performance Analysis

To evaluate the reliability of the autonomous collaboration algorithm in task allocation, resource scheduling, and collaborative decision-making, simulations were conducted with 20, 50, and 100 UAV nodes performing 25 tasks. Performance metrics, including task completion rate, remaining UAV energy, and smart contract response time, were measured to assess task execution efficiency, collaboration, and overall system reliability.

As shown in Figure 9, the task completion rate increases significantly with a higher number of nodes. The steep curve illustrates the algorithm's efficiency: fewer inflection points reflect smoother algorithm performance. For example, with 100 nodes, the task completion rate quickly reaches 100%, while with 50 nodes, it achieves 100% after a slightly longer duration. In the case of 20 nodes, the completion rate rose more slowly, reaching only 80% over the same time period. This indicates that the fewer the number of nodes, the task allocation and resource scheduling load exceeds the number of UAVs.

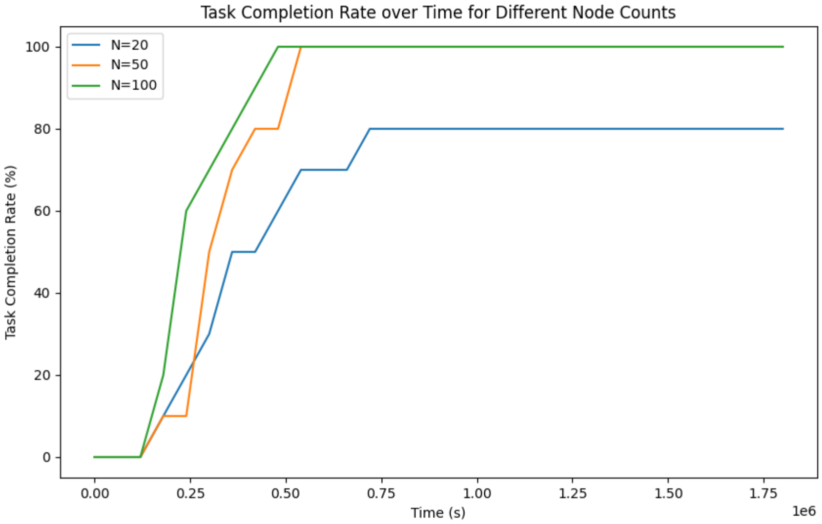


Figure 9. Task Completion Rate over Time for Different Node Counts.

Figure 10 shows that nodes in smaller UAV swarms experience a more rapid decrease in remaining energy during task execution. With an increased number of nodes, the average remaining energy per node is higher, suggesting that a larger node count helps extend the operating time of each node. For heavy task loads, a higher node configuration is recommended to reduce energy consumption per node and extend the system's overall task duration. These results demonstrate that the autonomous collaboration algorithm distributes task load more effectively with a larger node count, improving energy efficiency and task sustainability across the system.

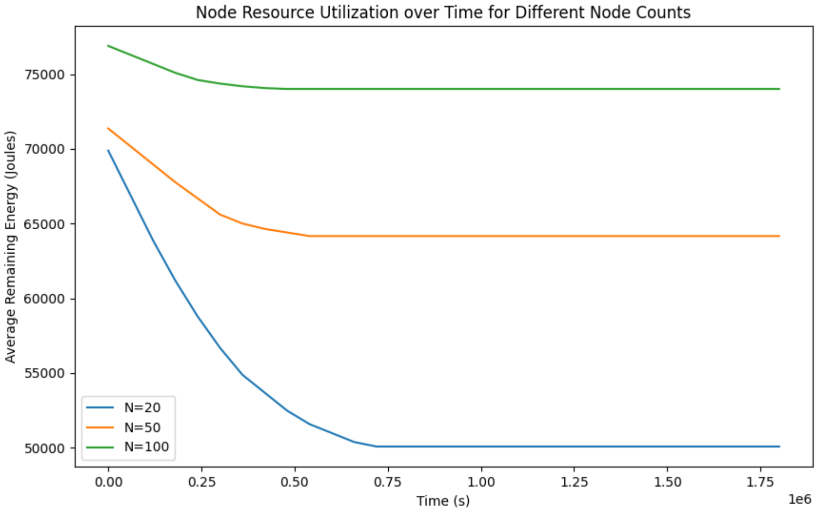


Figure 10. Node Power Utilization over Time for Different Counts..

Figure 11 also illustrates that upon triggering the smart contract, the collaborative algorithm initializes within 0-0.5 seconds. With 100 nodes, the initial delay reaches 24 ms, which is relatively high due to the large node count. After reaching a stable state, response times for all configurations fluctuate around 5 ms, indicating the monitoring phase for UAV status. Overall, while initial load fluctuations may impact the smart contract response time, it stabilizes quickly to maintain a low response time, ensuring the system's real-time responsiveness and efficiency.

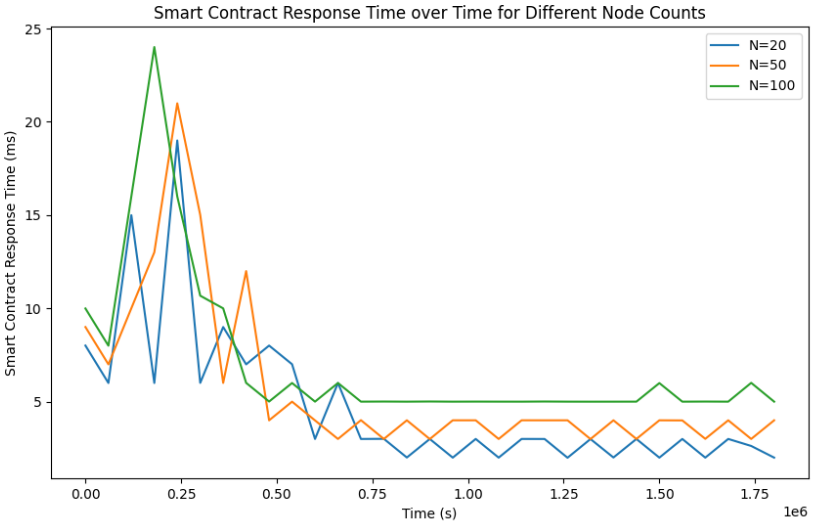


Figure 11. Smart Contract Response Time over Time for Different Node Counts.

5.3. Virtual Simulation and Performance Analysis

To verify the effectiveness of the proposed blockchain-based UAV network architecture and autonomous collaboration algorithms, we built a comprehensive simulation platform. This platform utilizes Microsoft's AirSim + UE simulation software along with the NS3 simulation tool for FANET networking, enabling the modeling of UAV swarm flight, communication, and task execution in complex environments. In this study, we created a 3D simulation environment in AirSim featuring a maritime defense platform to simulate UAV flight behavior under various obstacle conditions. Autonomous navigation, task execution, and collaboration functions were achieved using task-specific plugins and scripts.

For network communication, we enhanced the NS3 simulation tool to support data transmission and inter-node communication based on a reputation-based consensus algorithm. This integrated setup effectively utilizes FANET's communication features to improve data synchronization and consensus efficiency, closely replicating UAV communication processes.

In terms of blockchain and smart contract implementation, we employed the lightweight blockchain framework FISCO BCOS, streamlining the blockchain data structure to retain only essential task information and consensus mechanisms. Smart contracts were designed as modular templates that can be customized for different task types. Finally, we conducted performance testing through virtual simulations, recording metrics such as the relationship between task completion rate and bandwidth/packet loss rate, as well as between consensus time and network latency. The performance of the centralized control system was compared with that of the distributed collaborative system, focusing on task completion rate and network stability.

Figure 12 illustrates the integration of NS3-based communication simulation with the blockchain network for the UAV swarm. Under normal conditions, the nodes are fully interconnected using a mesh topology, with the blockchain embedded in the communication network and operating on a reputation-based consensus algorithm. When Node 0 initiates a consensus request to other nodes, Node 5, for instance, records the consensus data upon receipt and subsequently sends a verification message to confirm that other nodes have recorded the same consensus data, thereby achieving network-wide blockchain consensus. In the event of disconnection from the ground station or similar failures, the UAV swarm restricts the FANET network, blocking external signals and executing only the predefined tasks and algorithms embedded in the smart contracts.

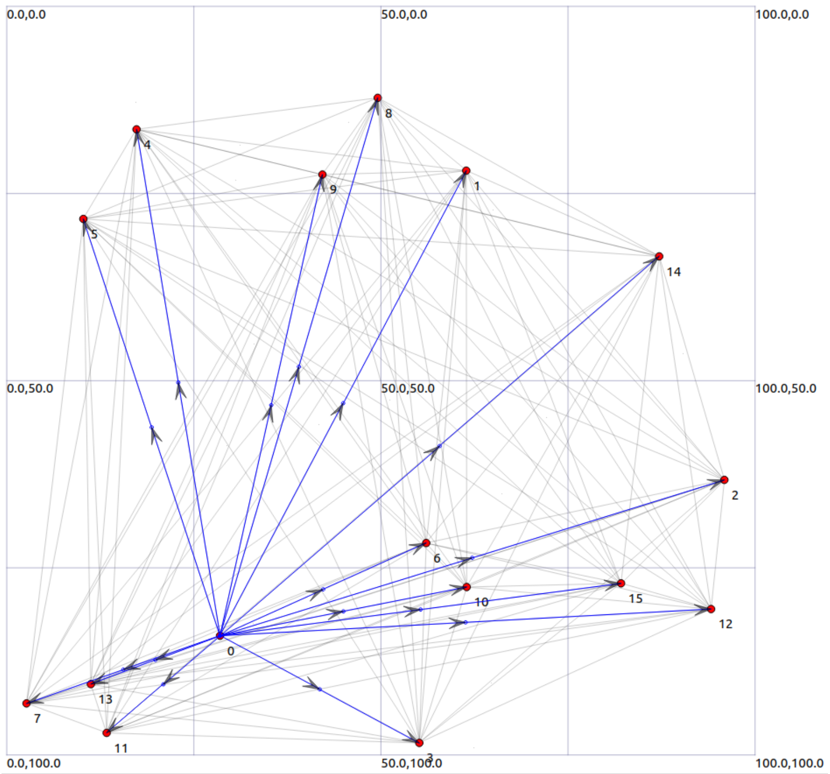
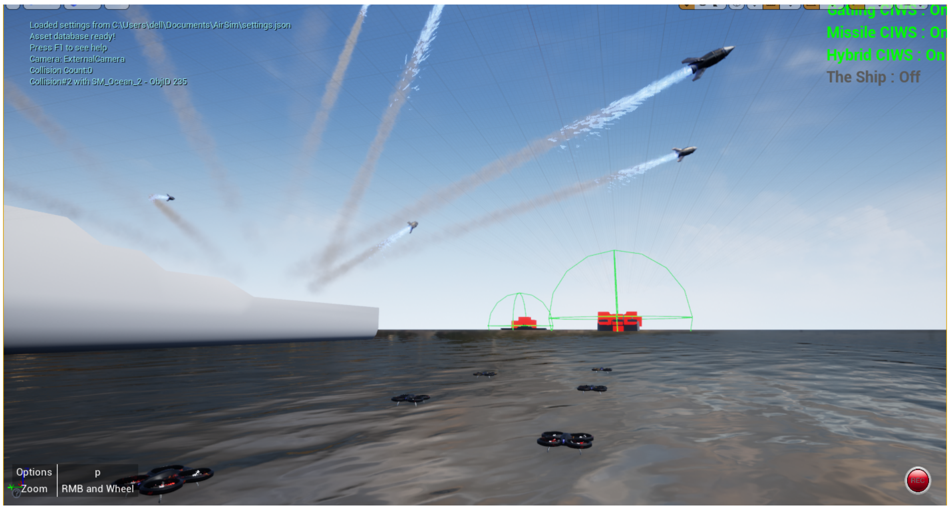


Figure 12. Drone Blockchain Communication Simulation.

Figure 13 presents the AirSim+UE virtual simulation platform. To validate the effectiveness of the proposed autonomous collaboration algorithm, an offline scenario was defined, simulating the loss of ground control. In the event of an attack on the UAV swarm, smart contracts are automatically activated to assess the situation and, based on the autonomous collaboration algorithm, assign tasks to coordinate a phased counterattack. Throughout this process, performance tests were conducted on the virtual simulation platform to evaluate the algorithm's real-world effectiveness.



```
Loaded settings from C:\Users\dell\Documents\AirSim\settings.json
Asset database ready!
Press F1 to see help
Camera: ExternalCamera
Collision Count:0
Collision#2 with SM_Ocean_2 - ObjID 235
```

Gatling CIWS : On
Missile CIWS : On
Hybrid CIWS : On
The Ship : Off

Figure 13. UAV virtual simulation platform based on Airsim.

Figure 14 illustrates the system's task completion rate across different bandwidth conditions, showing an increase in completion rate with higher bandwidth. At lower bandwidths (e.g., 1 Mbps), the task completion rate is approximately 90%, while at 10 Mbps or higher, the rate nears 99%. Bandwidth is a critical factor influencing communication efficiency and data synchronization speed within the UAV swarm. Increased bandwidth allows for larger data transmission volumes, effectively minimizing communication bottlenecks. In this system, efficient communication is essential for blockchain data synchronization and smart contract execution. When bandwidth is ample, data exchanges between nodes become smoother, thereby improving the efficiency of task allocation and resource scheduling, which leads to a higher task completion rate.

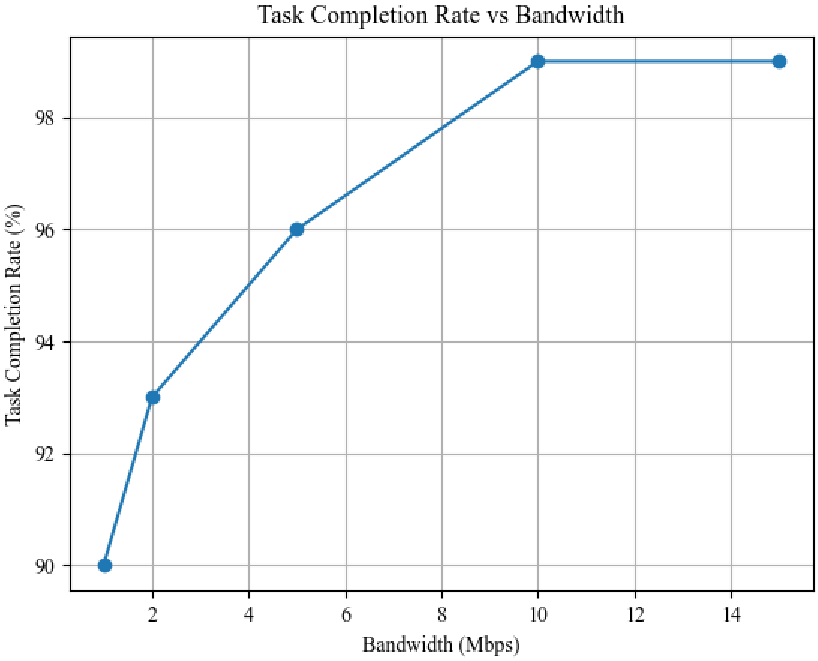


Figure 14. Relationship between task completion rate and bandwidth.

Figure 15 shows the impact of network latency on consensus time. As latency increases from 10 ms to 200 ms, consensus time rises slightly but remains relatively stable. At a latency of 10 ms, the average consensus time is around 1 second, while at 200 ms, it is approximately 1.5 seconds. Network latency affects message transmission between nodes, which in turn influences the efficiency of the consensus algorithm. However, the coupling of the reputation-based consensus algorithm with FANET's communication mechanism provides a level of latency tolerance within the system. This enables nodes to exchange critical information efficiently, even under high-latency conditions, thereby minimizing the negative impact of latency on the consensus process. Consequently, consensus time increases only slightly, and overall system performance remains stable.

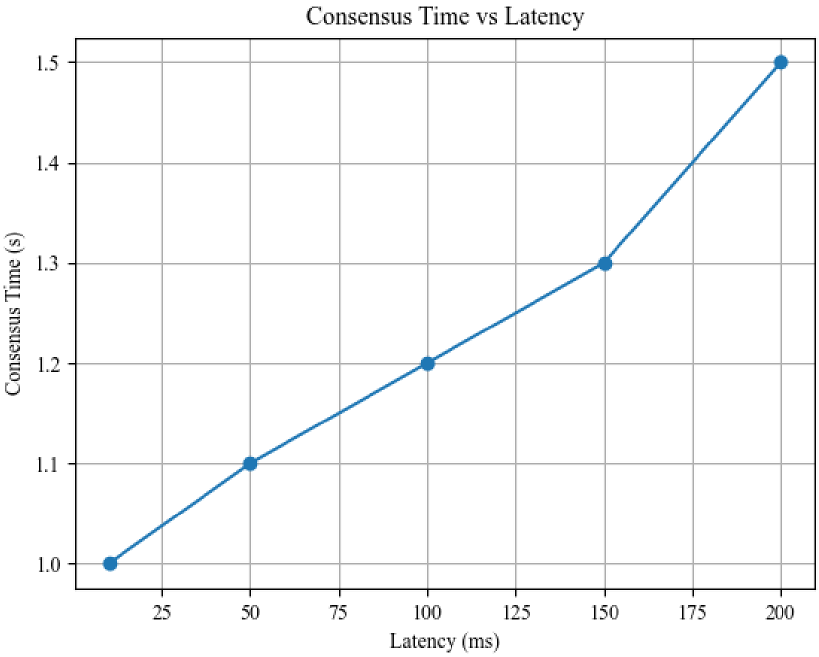


Figure 15. The relationship between consensus time and network latency.

Figure 16 illustrates the system's task completion rate under different packet loss rates. The results indicate a slight decline in task completion rate from 99% to 95% as the packet loss rate increases from 0% to 20%. Even with a high packet loss rate of 20%, the task completion rate remains at a relatively high level. High packet loss can lead to unreliable data transmission, impacting task allocation, status updates, and consensus achievement. However, our system mitigates this issue through several mechanisms:

- Data Redundancy and Retransmission: The communication protocol includes data redundancy and an Automatic Repeat reQuest (ARQ) mechanism, ensuring that critical task data can still be successfully received despite packet loss.
- Fault-Tolerant Consensus Algorithm: The reputation-based consensus algorithm is designed with fault tolerance, enabling consensus even when data from some nodes is missing.
- Multipath Communication in FANET: FANET supports multi-hop and multipath communication, allowing data to reroute through alternative paths if a specific communication link fails.

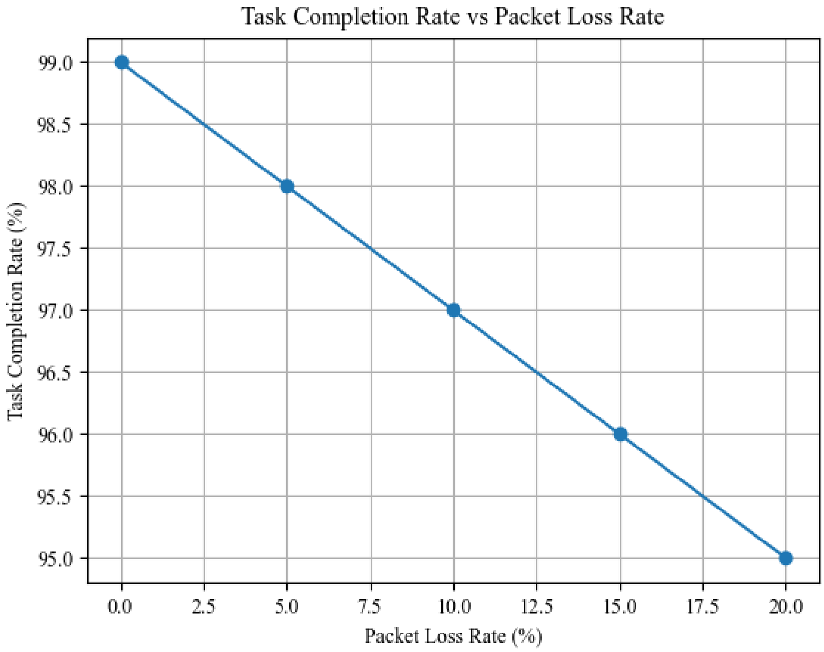


Figure 16. Relationship between task completion rate and packet loss rate.

As shown in Figure 17, the proposed solution is compared to traditional centralized control systems and other distributed collaboration methods, demonstrating clear advantages with a task completion rate of approximately 98%. Traditional centralized control systems depend on continuous communication links and directives from a ground control center, which leads to significant performance degradation when communication is disrupted or restricted. In contrast, the approach in this study leverages blockchain and smart contracts to enable autonomous collaboration and decision-making within the UAV swarm, eliminating the need for ground control.

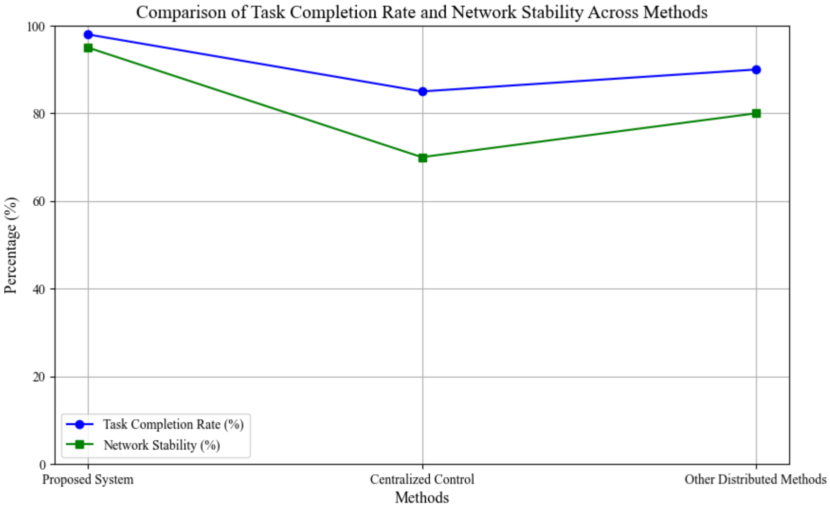


Figure 17. Comparison with existing methods.

Compared to other distributed collaboration methods, the main innovation of this study lies in the deep integration of blockchain technology with the FANET network. By implementing a reputation-based consensus algorithm and a coupled network communication mechanism, this design significantly improves consensus efficiency and network performance while reducing communication overhead. Experimental results indicate that the task completion rate and network stability of this system surpass those of other methods, especially in highly dynamic and complex environments.

6. Discussion

This paper introduces an offline collaborative decision-making framework for UAV swarms based on blockchain and smart contracts, showcasing its potential to significantly enhance task execution efficiency and data security in offline conditions. Compared to traditional centralized control systems, this approach excels in offline scenarios. Traditional methods rely on ground control centers, resulting in substantial performance degradation when communication is interrupted. In contrast, the integration of blockchain and smart contracts in this framework removes the dependence on ground control, allowing UAV swarms to maintain autonomous decision-making and task coordination in offline environments. This leads to higher task completion rates and improves the system's robustness and security.

Current research primarily focuses on applying blockchain for data exchange and basic task scheduling, with limited investigation into complex multi-UAV collaborative decision-making in offline settings. This study fills that gap, offering a new perspective on task execution and communication security for offline UAV swarm operations. The results underscore the extensive applicability of blockchain and smart contracts in enhancing UAV swarm autonomy and intelligence, particularly in scenarios requiring high levels of autonomy and security, such as rescue, reconnaissance, and logistics. Furthermore, with its lightweight design and dynamic resource management strategies, this framework is highly scalable, making it suitable for large-scale UAV swarm applications.

Future work

In future work, To further enhance the performance of this blockchain-based offline UAV collaboration framework, future research focus on the following Automated Smart Contract Generation and Optimization: Researching automated methods for generating and optimizing smart contracts to produce templates suited for various task types and scenarios, simplifying contract design and ensuring execution efficiency.

7. Conclusions

In the context of complex task execution by UAV swarms, existing ground-control-based coordination mechanisms are insufficient for offline environments due to factors like environmental variability and communication interruptions. To address this, we propose an offline UAV collaborative decision-making framework based on blockchain smart contracts. This framework enables UAV swarms to autonomously allocate tasks, collaborate, and secure data communication even in offline states.

The main contributions and innovations of this work are summarized as follows:

1. Blockchain-Based Distributed Communication Network: By integrating blockchain, we developed a distributed, tamper-proof communication network that ensures task data security in offline environments, enhancing UAV information sharing and coordination during task execution.
2. Smart Contract-Driven Collaborative Algorithm*: A smart contract-based collaboration algorithm was designed, encompassing task allocation, resource scheduling, and collaborative decision-making. Smart contracts enable UAVs to autonomously execute predefined collaboration rules offline, facilitating efficient task distribution and resource utilization.
3. Algorithm Performance Validation: Simulation experiments validated the proposed algorithm's effectiveness and reliability. Results across various task scenarios show that the algorithm effectively increases task completion rates as the number of UAV nodes grows, extends node endurance, and maintains low smart contract response times.

Author Contributions: Linchao Zhang: Conceptualization, methodology, writing,project administration, funding acquisition ; LeiHang: formal analysis, investigation, resources, data curation; Keke Zu: visualization, supervision; Yi Wang: software, validation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly funded by the Municipal Government of Quzhou (No. 2023D029), the Quzhou Science and Technology Bureau Soft Science Project (No. 2023r006), the Chenguang Plan of the Shanghai Municipal Education Commission (No. 21CGB08), and the Science and Technology Innovation Plan of the Shanghai Science and Technology Commission (No. 23YF1431200).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mohsan, S.A.H.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the unmanned aerial vehicles (UAVs): A comprehensive review. *Drones* **2022**, *6*, 147.
2. Sebbane, Y.B. *Intelligent autonomy of UAVs: advanced missions and future use*; CRC Press, 2018.
3. Li, J.; Yue, Q.; Huang, Z.; Xie, X.; Yang, Q. Vulnerability Analysis of UAV Swarm Network with Emergency Tasks. *Electronics* **2024**, *13*, 2005.
4. Alladi, T.; Chamola, V.; Sahu, N.; Guizani, M. Applications of blockchain in unmanned aerial vehicles: A review. *Vehicular Communications* **2020**, *23*, 100249.
5. Peddibhotla, U.; Kumar, R.; Sobin, C.; Kumar, P.; Javeed, D.; Islam, N. Securing Agricultural Communications: Blockchain Integration in UAV Networks for Smart Farming. 2024 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2024, pp. 1443–1449.
6. Telli, K.; Kraa, O.; Himeur, Y.; Ouamane, A.; Boumehraz, M.; Atalla, S.; Mansoor, W. A comprehensive review of recent research trends on unmanned aerial vehicles (uavs). *Systems* **2023**, *11*, 400.
7. Javaid, S.; Saeed, N.; Qadir, Z.; Fahim, H.; He, B.; Song, H.; Bilal, M. Communication and control in collaborative UAVs: Recent advances and future trends. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 5719–5739.
8. Javed, F.; Khan, H.Z.; Anjum, R. Communication capacity maximization in drone swarms. *Drone Systems and Applications* **2023**, *11*, 1–12.
9. Vassiliou, M.S.; Alberts, D.S.; Shah, S. Mission success: Assured communications and agile organizations. 2016 IEEE International Carnahan Conference on Security Technology (ICCST). IEEE, 2016, pp. 1–7.
10. Vassiliou, M.S.; Alberts, D.S.; Shah, S. Mission success: Assured communications and agile organizations. 2016 IEEE International Carnahan Conference on Security Technology (ICCST). IEEE, 2016, pp. 1–7.
11. Degbor, D.; Bedi, A.; Zhang, S.; Chabot, E. Communication-Facilitated Coordination in Agent Team Rescue Mission. 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE). IEEE, 2023, pp. 19–22.
12. Zadeh, R.B.; Zaslavsky, A.; Loke, S.W.; MahmoudZadeh, S. A multiagent mission coordination system for continuous situational awareness of bushfires. *IEEE Transactions on Automation Science and Engineering* **2022**, *20*, 1275–1291.
13. Javaid, S.; Saeed, N.; Qadir, Z.; Fahim, H.; He, B.; Song, H.; Bilal, M. Communication and control in collaborative UAVs: Recent advances and future trends. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 5719–5739.
14. Thasnim, J.; ElGhanam, E.; Hassan, M.S.; Osman, A. Towards Efficient Collaborative Delivery Solutions Using Drones and Ground Transportation: A Review. 2024 International Telecommunications Conference (ITC-Egypt). IEEE, 2024, pp. 727–732.
15. Pasandideh, F.; Costa, J.P.J.d.; Kunst, R.; Hardjawana, W.; de Freitas, E.P. A systematic literature review of flying ad hoc networks: State-of-the-art, challenges, and perspectives. *Journal of Field Robotics* **2023**, *40*, 955–979.
16. Nguyen, H.; Nguyen, M.; Truong, L.H.; Le, T.; Tran, T. UAV communication networks: benefits, research challenges and opening issues. *ICSES Transactions on Computer Networks and Communications* **2021**, *7*, 30–35.
17. Hassija, V.; Chamola, V.; Agrawal, A.; Goyal, A.; Luong, N.C.; Niyato, D.; Yu, F.R.; Guizani, M. Fast, reliable, and secure drone communication: A comprehensive survey. *IEEE Communications Surveys & Tutorials* **2021**, *23*, 2802–2832.

18. Thasnim, J.; ElGhanam, E.; Hassan, M.S.; Osman, A. Towards Efficient Collaborative Delivery Solutions Using Drones and Ground Transportation: A Review. 2024 International Telecommunications Conference (ITC-Egypt). IEEE, 2024, pp. 727–732.
19. Koulouris, C.; Dimitrios, P.; Al-Darraji, I.; Tsaramirsis, G.; Khadidos, A.O.; Khadidos, A.O.; Papageorgas, P. A Survey Study and Comparison of Drones Communication Systems. International Conference on Flexible Electronics for Electric Vehicles. Springer, 2022, pp. 351–361.
20. Pasandideh, F.; Costa, J.P.J.d.; Kunst, R.; Hardjawana, W.; de Freitas, E.P. A systematic literature review of flying ad hoc networks: State-of-the-art, challenges, and perspectives. *Journal of Field Robotics* **2023**, *40*, 955–979.
21. Nguyen, H.; Nguyen, M.; Truong, L.H.; Le, T.; Tran, T. UAV communication networks: benefits, research challenges and opening issues. *ICSES Transactions on Computer Networks and Communications* **2021**, *7*, 30–35.
22. Chai, S.; Yang, Z.; Huang, J.; Li, X.; Zhao, Y.; Zhou, D. Study on Cooperative Air-to-Ground Surveillance Planning and Controlling for Unmanned Aerial Vehicles. 2022 22nd International Conference on Control, Automation and Systems (ICCAS). IEEE, 2022, pp. 905–910.
23. Haas, Z.J.; Zheng, Z. Engineering a Network in the Sky. 2021 International Conference on Information Networking (ICOIN). IEEE, 2021, pp. 237–242.
24. Abdallaoui, S.; Ikaouassen, H.; Kribèche, A.; Chaibet, A.; Aglzim, E.H. Advancing autonomous vehicle control systems: An in-depth overview of decision-making and manoeuvre execution state of the art. *The Journal of Engineering* **2023**, *2023*, e12333.
25. Andruseac, G.C.; Tătulea, L.A. CENTRALIZED CONTROL OR DISTRIBUTED CONTROL–DO WE NEED A PARADIGM SHIFT? *INTERNATIONAL SCIENTIFIC CONFERINCE" STRATEGIESXXI"* **2022**, *18*, 162–170.
26. Schmid, S. Distributed Agent-Based Control in Dynamic Environments. 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS). IEEE, 2023, pp. 969–970.
27. Schmid, S. Distributed Agent-Based Control in Dynamic Environments. 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS). IEEE, 2023, pp. 969–970.
28. Saravanos, A.D.; Aoyama, Y.; Zhu, H.; Theodorou, E.A. Distributed differential dynamic programming architectures for large-scale multiagent control. *IEEE Transactions on Robotics* **2023**.
29. Miyazaki, R.; YASUTA, Y.; Xiao, H.; Tomita, K.; Kamimura, A. Decentralized Multi-UAV Formation Control and Navigation over a Self-Organizing Coordination Network. 2023 IEEE/SICE International Symposium on System Integration (SII). IEEE, 2023, pp. 1–6.
30. Wu, W.; Zhang, X.; Miao, Y. Starling-behavior-inspired flocking control of fixed-wing unmanned aerial vehicle swarm in complex environments with dynamic obstacles. *Biomimetics* **2022**, *7*, 214.
31. Puzicha, A.; Buchholz, P. Dynamic Mission Control for Decentralized Mobile Robot Swarms. 2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2022, pp. 257–263.
32. Pascarella, D.; Castrillo, V.; Iudice, I.; Pigliasco, G.; Vozella, A. Game-theoretic learning for the coordination of drone teams in autonomous cooperative inspection. Journal of Physics: Conference Series. IOP Publishing, 2024, Vol. 2716, p. 012058.
33. Castrillo, V.U.; Pascarella, D.; Pigliasco, G.; Iudice, I.; Vozella, A. Learning-in-Games Approach for the Mission Planning of Autonomous Multi-Drone Spatio-Temporal Sensing. *IEEE Access* **2024**.
34. Li, Y.; Li, C.; Chen, J.; Roinou, C. Energy-Aware Multi-Agent Reinforcement Learning for Collaborative Execution in Mission-Oriented Drone Networks. 2022 International Conference on Computer Communications and Networks (ICCCN). IEEE, 2022, pp. 1–9.
35. Furfaro, T.C. A distributed framework for embedded collaborative autonomy. OCEANS 2018 MTS/IEEE Charleston. IEEE, 2018, pp. 1–6.
36. Rajapakshe, S.; Wickramasinghe, D.; Gurusinghe, S.; Ishtaweera, D.; Silva, B.; Jayasekara, P.; Panitz, N.; Flick, P.; Kottege, N. Collaborative Ground-Aerial Multi-Robot System for Disaster Response Missions with a Low-Cost Drone Add-On for Off-the-Shelf Drones. 2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, 2023, pp. 75–80.
37. Wang, W.; Huang, B.; Wu, Y. Design of a Multi-unmanned Aerial Vehicle Cooperative Hunting System Based on MADDPG. International Conference on Man-Machine-Environment System Engineering. Springer, 2024, pp. 480–486.
38. Arshad, U.; Faheem, Y.; Shaheen, R. Futuristic Blockchain-based Secure and Verifiable Drone Surveillance System: Chain in the Sky. *IPSI Transactions on Internet Research* **2024**.

39. Gowda, V.D.; Kauser, S.H.; Shaik, N.; Prasad, K.; Bhardwaj, R. Advancing Drone Communication Frameworks With Blockchain in Industry 5.0 Applications. In *Drone Applications for Industry 5.0*; IGI Global, 2024; pp. 111–129.
40. Jing, J.; Yang, Y.; Li, M.; Li, B.; Zhang, J.; Ma, J. Blockchain-based federal learning program for drone safety. International Conference on Computer Network Security and Software Engineering (CNSSE 2024). SPIE, 2024, Vol. 13175, pp. 421–430.
41. Mehta, S.; Kundra, D. Improving Drone Operational Efficiency through IoT-Blockchain Integration. 2023 4th International Conference on Intelligent Technologies (CONIT). IEEE, 2024, pp. 1–5.
42. Saroopa, P.; Poongodi, K.; Saranya, V.; Subaash, S. Securing Drones with Blockchain-Enabled Privacy-Preserving Authentication. 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS). IEEE, 2024, Vol. 1, pp. 1–6.
43. Xie, M.; Chang, Z.; Alfarraj, O.; Yu, K.; Chen, T.; Li, H. BAZAM: A Blockchain-Assisted Zero-Trust Authentication in Multi-UAV Wireless Networks. *arXiv preprint arXiv:2407.00630* **2024**.
44. Akram, J.; Anaissi, A. Decentralized pki framework for data integrity in spatial crowdsourcing drone services. 2024 IEEE International Conference on Web Services (ICWS). IEEE, 2024, pp. 643–653.
45. Han, P.; Wu, X.; Sui, A. DTPBFT: A dynamic and highly trusted blockchain consensus algorithm for UAV swarm. *Computer Networks* **2024**, p. 110602.
46. Xiong, R.; Xiao, Q.; Wang, Z.; Xu, Z.; Shan, F. Leveraging lightweight blockchain for secure collaborative computing in UAV Ad-Hoc Networks. *Computer Networks* **2024**, *251*, 110612.
47. Peddibhotla, U.; Kumar, R.; Sobin, C.; Kumar, P.; Javeed, D.; Islam, N. Securing Agricultural Communications: Blockchain Integration in UAV Networks for Smart Farming. 2024 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2024, pp. 1443–1449.
48. Sarfaraz, A.; Chakrabortty, R.K.; Essam, D.L. Reputation based proof of cooperation: an efficient and scalable consensus algorithm for supply chain applications. *Journal of Ambient Intelligence and Humanized Computing* **2023**, *14*, 7795–7811.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.