Blockchain-Assisted Adaptive Reconfiguration Method for Trusted UAV Network

Lingjun Kong †, Bing Chen *†, and Feng Hu †

The College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
* Correspondence: cb_china@nuaa.edu.cn
† These authors contributed equally to this work.

Abstract: Due to the high mobility of nodes and the complexity of the mission environment, mission-oriented UAV networks are not only subject to frequent topology changes, but also to the risk of being compromised, hijacked and corrupted. As a result, an operating UAV network is essentially a Byzantine distributed system whose physical structure and node trustworthiness change over time. How to implement the global management of UAV networks to achieve a rational allocation of UAV network resources and reconfiguration of trusted networks is a problem worthy of in-depth study. The method proposed in this paper introduces a lightweight storage blockchain in the UAV network through two-stage consensus, firstly performing data consensus on the local state records of the nodes, then performing decision consensus on the data consensus results using algorithms such as fuzzy K-Modes clustering and global trustworthiness assessment, and finally recording the decision consensus results into a new block as the new configuration information of the UAV network. A lightweight storage blockchain-assisted trusted zone routing protocol (BC_TZRP) is designed to dynamically and adaptively build configurable trusted networks in a way that the blockchain continuously adds new blocks. Using QualNet simulation experimental software, an experimental comparison between the classical routing protocol for mobile self-organizing networks and the traditional consensus algorithm for blockchains is conducted. The results show that the approach has significant advantages in terms of packet delivery rate, routing overhead and average end-to-end delay, and can effectively improve the overall working life and fault tolerance of the UAV network.

Keywords: mission-oriented UAV network; lightweight storage blockchain; reconfiguring trusted networks; fuzzy K-modes clustering

1. Introduction

Low-cost, miniaturized, scaled-up and networked UAV clusters have been the focus of research and an important development direction at home and abroad. The qualities of low cost, easy and flexible deployment, and high resistance to destruction have made UAV networks widely used in many fields. In recent years, the domestic consumer-grade UAV market has become increasingly saturated, and industrial-grade UAVs have gradually become the mainstay of the emerging industry. Combined with traditional industries, UAV networks, as gradually maturing aerial platforms, have begun to be widely used in the civilian field. They play an irreplaceable and important role in many special environments, such as security monitoring, emergency disaster mitigation, rescue, exploration and digital cities. In terms of military applications, as an emerging combat force, UAVs have more and more types of missions and their complexity is escalating, single UAVs can no longer meet military needs, and UAV cluster operations will become an important means of modern warfare in the future. Therefore, UAV network-related technologies have become the object of high attention and in-depth research by various military powers.

Since 2000, the U.S. Department of Defense has continued to release the “Unmanned Systems Integrated Roadmap” planning, according to the British International Institute
Electronics 2022, 11, 2549

for Strategic Studies released the “Military Balance 2022” statistics, the U.S. military under 20 kg micro and small UAVs more than 10,000. Russia’s Concept for the Use of Unmanned Aircraft Systems in the Russian Armed Forces by 2025 comprehensively lays out the development of unmanned aircraft equipment and force construction, and according to the plan, the proportion of unmanned combat systems in the Russian military equipment system will exceed 30% in 2025. In the Israeli-Palestinian conflict in May 2021, the Israeli army used a swarm of miniature multi-rotor drones guided by artificial intelligence systems for the first time, and Israel became the first country to integrate artificial intelligence technology to command a swarm of drones in combat. At India’s Army Day parade in 2021, India demonstrated for the first time an offensive drone cluster system consisting of 75 drones that can autonomously identify and shoot down targets [1].

With the boom in UAV technology, anti-UAV technologies [2–5] have emerged. In addition to traditional methods such as communication/navigation system jamming and signal intrusion, research into anti-drone technologies such as electronic warfare, cyber warfare, drone countermeasures and laser weapons has been vigorously pursued. These new anti-drone technologies can perform GPS decoys, interfere with the normal operation of drones, and even counteract drones, such as a drone countermeasure system based on drone clustering proposed by Brust et al. [6] from the University of Luxembourg. Although some progress has been made in drone clustering technology, drones are currently vulnerable to jamming, trapping and strikes due to their own limited resources, the open nature of wireless communications and the imperfection of aerial countermeasures. Mission-oriented UAV networks operate in highly dynamic, complex and unstructured environments where the size of the network, topology and trustworthiness of the network nodes change over time. Improving the fault tolerance of the network and maintaining its trustworthiness during missions is a major challenge for distributed UAV networks with limited resources and no central support.

The UAV network in mission execution faces three major unfavorable objective conditions, which are non-security, complex operation environment, no central support and limited resources of nodes. Therefore, the UAV network not only has to solve the routing reconfiguration brought by topology changes, but also needs to tolerate the internal error nodes generated by the UAV nodes due to their own consumption, being damaged and compromised. Unauthorized access by external nodes must of course also be prevented. The decentralized, de-trusted, and tamper-proof nature of blockchain fits well with the objective nature of mission-based drone networks. In this paper, we propose a stateless permission blockchain system that not only makes the UAV network capable of authentication, but also dynamically isolates untrustworthy nodes by releasing detection scores for all nodes. The continuously updated blockchain provides the UAV network to adaptively perform trusted reorganization of the network while providing a reference for routing. The blockchain-assisted hierarchical Trusted Zone Routing Protocol (BC_TZRP) is essentially a lightweight storage blockchain system based on a zone routing protocol. The system performs data consensus on the local state records of nodes through a two-stage consensus and then implements decision consensus on the data consensus results, and adaptively reconfigures the UAV network based on the decision results.

The main contributions of this work are as follows:

• First, an adaptive configuration method for UAV trusted networks is proposed. The method uses a new block of the blockchain to record the global state assessment of the network nodes at each stage. All UAV nodes reorganize the trusted network based on the updated blockchain and automatically reconfigure the hierarchical logical network based on the newly elected set of delegated authorized nodes to cope with the dynamic nature of the UAV network topology and the time-varying trustworthiness of the network nodes while reducing routing blindness.

• Second, a lightweight storage blockchain with a two-stage consensus is designed. Nodes monitor all forwarding behavior of their neighboring nodes at the network layer, and different behaviors are given corresponding trustworthiness discounts.
Based on the rhythm of the blockchain consensus, each node submits a state transaction including a local trust assessment of neighboring nodes to the upper network. The blockchain system first performs a data consensus on the transactions of the nodes at each stage and then performs a decision consensus on the data consensus results. The decision consensus result is recorded in the solid new block as trusted network configuration information. The blockchain adds only small decision data to each consensus, while the local state transaction dataset at each stage is discarded directly.

Third, propose a blockchain-assisted hierarchical zone trusted routing protocol. A K-modes clustering method is used to periodically elect the zone centers of the UAV network using a list of neighboring nodes, combined with a global trustworthiness assessment of the nodes as a non-numerical node feature vector. The upper layer network reconstructed by the dynamically updated zone centers manages the UAV network and takes on the main operations in stages, thus balancing the consumption of network resources and minimizing the involvement of erroneous nodes in consensus operations. This not only increases the efficiency of the use of UAV network resources, but more importantly, the fault tolerance of the entire network allows for the presence of more than a third of errant nodes.

The rest of the paper is organized as follows. The related work is discussed in Section 2. The system model is in Section 3. A specific description of the recommended scheme (BC_TZRP) in Section 4. In Section 5, We discuss the feasibility of the recommended scheme, including complexity analysis, scalability analysis and security analysis. Section 6, Designing a working scenario of a UAV network in a Byzantine environment and performance analysis of the proposed solution through simulation experiments. Finally, the conclusion is presented in Section 7.

2. Related Work

Mobile self-organizing networks have frequent topology changes due to the mobility of nodes, making it impossible to complete data transmission using static routing protocols. Nodes on the transmission path must be dynamically configured according to specific convergence metrics, such as link state or some vector distance, in order to adapt to network changes. MANET networks are thus dynamically reconfigured networks with a family of a priori routing protocols with optimal link states (OLSR, FSR etc.) [7–9], as well as reactive routing protocols with distance vectors as convergence conditions, such as AODV [10,11], DSR [12], and some hybrid routing protocols have been extensively studied.

According to OLSR (Optimal Link States Routing Protocol), nodes discover neighboring nodes by sending probe packets periodically, broadcast topology information, and select multipoint relays (MPR) to reduce topology control information (TC), and pre-construct routing tables with optimal link states as the convergence condition. FSR (Fisheye State Routing Protocol) uses probing packets of different frequencies to reduce routing overhead based on OLSR, and employs the principle that the closer the fisheye view is, the clearer it is for nodes of different ranges. AODV (Ad hoc on-demand vector routing protocol) builds routes on-demand with vector distance as the convergence condition. AODV (Ad Hoc on-demand vector routing protocol) uses vector distance as a convergence condition to build routes on demand and initiates route discovery and maintenance at the start of transmission, reducing routing overhead but increasing data transmission delay.

The above methods of routing reconstruction have a large node search range, are somewhat blind and increase the communication overhead. Some researchers have introduced location information to aid routing, such as the Location Assisted Routing Protocol (LAR) [13] which uses GPS location information to determine transmission nodes, reducing the overhead of route discovery and speeding up network reconfiguration. However, the complex operating environment of UAV networks also increases the likelihood of GPS decoy-like attacks. Due to the limitations of routing overhead and transmission delay, the routing protocols studied above can only be used in small-scale, flat network architectures for MANET and cannot be applied to large-scale, multi-node MANET applications.
To enhance the scalability of MANET networks, routing protocols suitable for hierarchical network architectures have received attention, such as the Zone Routing Protocol (ZRP) [14] and its family [15,16], hierarchical routing protocol (LANMAR) [17,18], etc. ZRP is a hybrid routing protocol that divides the network into small overlapping areas based on a set range of areas (number of hops). Proactive routing protocols are used within areas to reduce transmission delays, and reactive routing protocols are used between areas to generate routes on demand and reduce inter-area routing overhead. Network routes are dynamically reconfigured via Border Broadcast Resolution Protocol (BRP) based on the central node of the partition and the boundary nodes of the overlapping zones. LANMAR marks the nodes in the MANET network whose neighboring nodes exceed a set threshold number as seek points and configures all seek points as upper layer management networks. Inter-regional node communication is performed through the upper layer network. The common feature of ZRP and LANMAR routing protocols is the dynamic selection of special state nodes in the network to form the upper logical network, and the establishment of a hierarchical network architecture with dynamically configured optimal states in the MANET network, which effectively narrows the scope of routing selection and enhances the network scale operation. The routing protocols studied above assume a secure network and do not take into account unauthorized access, or compromised nodes. As special MANETs, the mission environments of UAV networks are mostly complex and non-secure, requiring more secure network reconfiguration methods.

To increase the security of routes, the re-routing adds the evaluation of node reputation. In [19], a reputation-based ant colony optimization method is designed to identify low reputation nodes by modifying the target source routing protocol (DSR) to resist grey hole attacks. Ref. [20] evaluates the reputation of nodes through cooperation between them during the route request process as a reference for route selection. Ref. [21] also evaluates the reputation of nodes, discarding requests from untrustworthy nodes to avoid flooding attacks on routing requests. In [22–24] reputation assessment of node behavior and inclusion of node trustworthiness conditions in existing routing protocols to exclude such nodes when reconfiguring network routes. The above routing protocols consider security issues and circumvent low trustworthiness nodes during network routing configuration, improving the security of the routes, but also increasing the computational overhead of all participating nodes and limiting the network size.

The decentralized, de-trusted and tamper-proof nature of blockchain technology fits well with a distributed MANET network with no management center and no infrastructure. Refs. [25,26] introduces blockchain to MANET. Using a subset of nodes to validate routing actions taken by other nodes and using the distributed consensus mechanism of the blockchain network to accumulate the reputation of each node. Specifically, we use the heterogeneous difficulty of proof-of-work to represent the trustworthiness of verification and design a scoring system to isolate malicious nodes by distributed consensus, and route reconfigurations to directly avoid malicious nodes. However, this direct approach with a traditional blockchain is overwhelming for resource-constrained MANET nodes, with proof-of-work (POW) computation consumption, progressively increasing block storage, and its synchronous update method on the blockchain increasing bandwidth overhead. Ref. [27] creates a distributed management system of trust and rewards in MANET that detects uncooperative nodes via the blockchain. However, this virtual network requires a “mining node” to perform the large consuming calculations such as consensus, which the peer-to-peer lightweight nodes in the UAV network cannot handle.

Most of the above research results reconfigure MANETs at the network layer through the dynamic selection of routes. Furthermore, in insecure environments, malicious nodes still exist in the network and each reconfiguration requires an unnecessary increase in operational overhead. The introduced permissionless blockchain, where any node enters the network indiscriminately, increases communication and storage overhead and the proof-of-work consensus approach is unaffordable for UAV networks with limited node resources. Ref. [27] uses blockchains to build global management systems that reorganize...
the network to exclude uncooperative nodes. Thus, introducing a blockchain to globally manage node trust in distributed UAV networks is a feasible approach. Table 1 shows a performance comparison of the MANET network reconfiguration methods, focusing on security, scalability and the overheads of computation, communication and storage.

Table 1. Performance comparison of network reconfiguration methods.

| References | Security | Scalability | OverHead |
|-----------|----------|-------------|----------|
|           |          |             | Computation | Communication | Storage |
| [7–12]    | Low      | Low         | Low       | -             | Low     |
| [13]      | Low      | Low         | Low       | Low           | Low     |
| [14–18]   | High     | Low         | Middle    | Middle        | Low     |
| [19–24]   | Low      | High        | Middle    | Low           | Low     |
| [25,26]   | High     | Middle      | High      | High          | High    |
| [27]      | High     | High        | High      | High          | High    |

Refs. [7–12] Flat architecture routing: Suitable for small to medium-sized mobile Ad hoc networks. Routing overhead, transmission delay increases with network size; [13] Location-assisted routing: Reduce the overhead of the reconfiguration process by adding node location information and reducing the range of routing nodes selected; [14–18] Hierarchical routing: Good scalability in a secure environment. Scalability is severely affected when malicious nodes interfere, such as DOS attacks; [19–24] Consider node trustworthiness routing: Reconfiguring network routing in a flat network architecture. Considering the node trustworthiness. Security is improved, but computation is increased; [25,26] Blockchain-assisted reputation routing: Accumulating node reputation within a local region improves security, but lacks global reach. High POW consensus computation and high communication overhead; [27] Blockchain Node Reputation Management: Blockchain reputation management platforms avoid uncooperative nodes from participating in network reconfiguration. However, well-resourced mining nodes are requested.

Li, Gang et al. in [28] proposed a blockchain-enhanced data collection framework where UAVs are used to assist in data collection from sensor nodes in large-scale monitoring scenarios in wireless sensor networks. A Merkle tree-based UAV authentication mechanism is designed to ensure the security of data transmission. The scheme determines the authenticity of the collected data by verifying the legitimacy of the UAV node identity. Query verification using Merkle evidence is also time-consuming when the number of drone nodes increases, especially when given a non-existent piece of evidence. Inspired by this, a global authentication blockchain can be established in the UAV network to prevent unauthorized access by malicious external nodes. Islam, Anik et al. [29] proposed a joint learning base data accumulation scheme combining drones and blockchain for remote areas where IoT devices face network scarcity and potential cyber threats. With the consent of the mining nodes, the federated learning is simply stored on the blockchain to obtain models, reducing the storage overhead of the blockchain. Although it does not fit the drone network reconfiguration application scenario, the idea of lightweight storage, and the two-stage consensus is worth addressing.

Inspired by the above research results, the solution proposed in this paper: a blockchain-assisted adaptive reconfiguration method for a trusted hierarchical network of UAVs to build a lightweight authentication blockchain in the UAV network. The local state records of drone nodes are phased for data consensus, and the blockchain system implements decision consensus on the data consensus results. Finally, only the configuration information after the decision is stored on the blockchain. The drone nodes reconfigure the trusted network based on the blockchain. This scheme is based on the protocol architecture of ZRP, which ensures network scalability, serious blockchain to ensure the authenticity of node identity, periodic updates of the upper network to guarantee node trustworthiness and reasonable allocation of resources.
3. System Models

According to the discussion in the previous section, existing mobile self-organizing network reconfiguration methods either construct new network routes based on dynamically changing topology information or combine other attributes of nodes, such as location information, reputation, etc. to reduce path blindness during routing. In terms of computation, communication, storage or security, none of the solutions can satisfy the application scenarios of lightweight UAV networks in the context of Byzantine General problem. The adaptive trusted reconfiguration approach for UAV networks recommended in this paper is achieved by a blockchain system deployed on UAV nodes. The blockchain system is a permissioned blockchain for identity authentication based on vector commitment cryptography, which dynamically evaluates the trustworthiness of nodes by using their own state change records as consensus objects. The block system isolates untrustworthy nodes from the network by dynamically aggregating identity vector commitments, periodically recording the global state of the nodes. The drone nodes adaptively reconfigure the trusted hierarchical architecture network through the continuously updated blockchain and construct secure routes based on the blockchain record identity information.

Figure 1 illustrates the system model of a blockchain-assisted dynamically configurable network.

3.1. Threat Model

The UAV network is an ad hoc mission network, which operates in an unknown and non-secure environment. The exploitability of wireless communication and the mobility of UAV nodes make UAV networks face more challenges than ordinary MANETs. The threats that cause the UAV network to fail to accomplish its mission are not to be ignored.

- Environmental Threats: The UAV network mission execution environment is complex and variable, which may be a distress rescue site or an enemy-occupied area on the battlefield. The interference of unknown terrain topography leads to frequent partitioning of the UAV network and rapidly changing topology information, increasing the overhead of network routing reconfiguration. Multiple anti-drone techniques can not only disrupt communications, but can even cause direct drone damage. UAV networks are thus required to be scalable, adapt to network scaling caused by the environment, and dynamically sense the departure and joining of network nodes.
- Cyber Attacks: The hostile environment and the openness of wireless communication are the security of the UAV network is vulnerable. External malicious nodes can launch a variety of network attacks, such as DOS attacks, black hole attacks, can collude wormhole attacks, etc. These attacks verify to weaken the availability and scalability of the network. Link layer attacks can achieve intrusion hijacking of legitimate nodes.
Internal attacks launched by nodes compromised with legitimate identities can cause more damage to the network. Therefore, in addition to establishing an authentication environment to prevent unauthorized access, the UAV network should also have the ability to detect node trustworthiness in real time and isolate untrustworthy internal nodes from the mission network in a timely manner.

- **Selfish Nodes:** Due to their reduced energy, nodes only receive messages and do not forward them out of self-protection. Although such uncooperative zombie nodes do not launch harmful attacks, they interfere with the network routing configuration and generate invalid communications, wasting energy and reducing the overall performance of the network. The system should also have the ability to identify and tag them for isolation.

### 3.2. Network Model

Drone clustering is an important model for UAV network applications to enhance the destructive resistance of UAV networks with node redundancy and provide mission completion rates. The large-scale mission environment and numerous nodes make the scale UAV network has higher requirements for scalability and security. Based on the threats that exist, the mission-oriented UAV network is essentially an asynchronous network with Byzantine General problem at scale, containing different network models containing two phases of mission preparation and mission execution.

- **Centralized network:** The centralized network in the mission preparation phase completes the initialization of the nodes and the network in a secure environment with the registration server as the center. All UAV nodes participating in the mission provide their own unique information, such as IP, NIC address, etc., and submit registration applications to the center. The registration server allocates the corresponding public and private keys of the UAV nodes and generates 48-bit hash values for each UAV node based on the public key as the UAV identity \( ID_i \). The registration center will accumulate all IDs into a fixed-size identity vector commitment \( \vec{ID} \) based on the size of \( ID_i \), and calculate the witness \( W_i \) of the identity existence corresponding to \( ID_i \), and initialize an empty one for storing the untrusted node word vector commitment, and randomly select \( M \) authorized proxy nodes as the consensus committee. The registry creates the Genesis block and deploys the membership information of the consensus committee, the identity vector commitment and the corresponding identity authentication smart contract in it. The blockchain program is deployed on each drone node \( ID_i \), allocating the corresponding key pair \( PK_i, SK_i \), and identity witness \( W_i \), and synchronizing the creation block to complete node initialization, and \( ID_i \) becomes a blockchain network node. The final composition of the UAV network with \( N \) nodes is denoted as \( \{ ID_i \}, i \in \{ 1,2,\ldots,N \} \).

- **Distributed network:** The UAV network in the mission execution phase is an asynchronous distributed network. After the mission starts, the registration server is not involved in the mission and the UAV network is handed over to the blockchain system for management, which is mainly responsible for node identity authentication and node trustworthiness decision during communication. Authorized agent points constitute the upper logical network \( ID_A^j, (j \in [M]) \), which is responsible for consensus computing and inter-zone routing. Combined with the ZRP routing protocol. By detecting the forwarding behavior of nodes, an adaptive reconfigurable network model with hierarchical architecture is constructed based on a continuously updated blockchain.

### 4. Recommended Scheme

The scale of UAVs can effectively improve the destructiveness of the UAV network, which requires the scalability of the UAV network. Poorly controlled routing overheads can lead to a rapid increase in routing overheads required for network maintenance and route discovery until all wireless bandwidth is consumed and the network collapses. The
The purpose of the UAV network is to exchange information and intelligence between nodes and collaborate on tasks, and excessive routing overhead also affects the ability to carry business data streams. In addition to network size, node density also has a serious impact on routing overhead. The UAV network also suffers from the Byzantine problem, where interference from errant nodes also generates inefficient routing overhead. Therefore, to maintain the high availability of the UAV network, the network is dynamically reconfigured based on the status of the nodes at the current stage in terms of location, trustworthiness, etc. The network is reconfigured in such a way that nodes in good condition are selected to form the upper management network, while untrustworthy nodes are isolated from the network. The network is reconfigured in such a way that the most trusted nodes are involved in each phase of the network, thus reducing routing blindness and minimizing routing overhead.

The recommended solution in this paper is the Blockchain-assisted Trusted Zone Routing Protocol (BC_TZRP), a cross-layer routing protocol over the link, network and application layers, including the Neighbor Node Detection Subprotocol (NDP) at the link layer, the Intra-Zone Routing Subprotocol (IARP), the Inter-Zone Routing Subprotocol (EARP) and the Border Broadcast Routing Subprotocol (BRP) at the network layer, and the Blockchain at the application layer. It includes two major functions: firstly, a real-time check of all nodes to generate state record transactions; and secondly, a two-stage consensus of authorized nodes, with the consensus result serving as the basis for reconfiguring the network. The process is that at the link layer nodes detect current neighboring nodes through NDP, and the improved IARP protocol of this method sends topology information to these neighboring nodes in addition to each node monitoring the forwarding behavior of its neighboring nodes, giving different scores for different behaviors, and counting the local trust loss of all its own nodes by consensus cycle, and sending the statistics and the list of neighbors as local state record transactions sent to authorized nodes, authorized node information in the latest block of the blockchain, the specific process is reflected in Algorithm 1; second, two-stage consensus is run on authorized nodes, authorized nodes receive local state transaction records from all nodes and reach data consensus in the form of a common subset, each authorized node invokes the smart contract in the Genesis block to make decisions on data consensus results, so on each authorized node, the Therefore, at each authorized node, the same data result with the same smart contract will reach a consensus decision result, see Algorithm 2. Therefore, each authorized node synchronizes a new block, and the node receiving the new block updates the blockchain and builds a new trusted network based on the configuration information in the latest block. It is worth noting that the local state record transactions at each stage are for decision consensus and do not need to be saved to the blockchain, which only needs to save the result of the decision, a fixed very small block header. Therefore the latest block in the blockchain holds the current global trustworthiness of the network nodes and the configuration information of the upper layer network, a light storage blockchain. The routing protocol, aided by the blockchain, circumvents untrustworthy nodes, enables cross-zone routing through the upper layer network, reduces redundant forwarding and reconstructs low-overhead trusted routes. As shown in Figure 2.
Algorithm 1 Function: AssessmentNeighbours with TrustState\(i\) and message\(i\)

Function: AssessmentNeighbours(TrustState\(i\), message\(i\))

1: #Running in ID\(i\)
2: #t0: Timeout for forwarding delay. t1: Timeout for non-forwarding.
3: let: \(t0 \leftarrow v1(ms), t1 \leftarrow v2(ms)\)
4: #Obtain the list of current neighbor nodes of node ID\(i\)
5: curNeighbours[] ← getNeisFromNDP()
6: # The sending time, which is used to accumulate the forwarding timeout.
7: curtime ← getlocaltime()
8: #Discount\(j\): Trustworthiness discount for one forwarding behavior of neighboring nodes ID\(j\), count: The forwarding times of node ID\(i\) during the cycle.
9: Discount\(j\) ← 0
10: upon received message\(j\)−i from neighbors do
11: # parsing the behavior of neighbor ID\(j\)
12: timeout ← getlocaltime() − curtime
13: #message\(j\)−i: The message returned by neighbor node ID\(j\), which is sent by ID\(i\) to determine whether ID\(j\) has tampering behavior.
14: if timeout < t0 then
15: if message\(j\)−i = message\(i\) then
16: Discount\(j\) ← 0
17: end if
18: if message\(j\)−i ≠ message\(i\) then
19: Discount\(j\) ← −3
20: end if
21: end if
22: if timeout > t0 then
23: if message\(j\)−i = message\(i\) then
24: Discount\(j\) ← −1
25: end if
26: if message\(j\)−i ≠ message\(i\) then
27: Discount\(j\) ← −3
28: end if
29: end if
30: #According to the list of neighbors obtained by NDP protocol, find out the nodes that do not forward data on timeout and define them as selfish nodes.
31: if timeout > t1 and ID\(j\) ∈ curNeighbours then
32: Discount\(j\) ← −2 /*selfish behavior*/
33: end if
34: #Accumulated trustworthiness loss of neighbor nodes.
35: TrustState\(i\).curScore\(j\) = TrustState\(i\).curScore\(j\) + Discount\(j\)
36: Output : {TrustState\(i\)}
Algorithm 2 Local State Transactions Generating for Node ID
t.

1: /*Running in an UAV ID\textsubscript{i}*/
2: // StateData\textsubscript{i}: State Transaction of ID\textsubscript{i} in a cycle.
3: Let StateData\textsubscript{i} ← \{\}
4: // Pre Nei: List of original neighboring nodes.
5: Let Pre Neis ← \{\}
6: // Cur Neis: List of current neighboring nodes.
7: Let Cur Neis ← \{\}
8: // TrustState\textsubscript{i}: Node ID\textsubscript{i}’s neighboring nodes' trust discount accumulation.
9: Let TrustState\textsubscript{i} ← \{\}
10: // bReqireConsensus: Node ID\textsubscript{i} active request consensus flag.
11: Let bReqireConsensus ← false
12: // The number of times node ID\textsubscript{i} sends data in the consensus cycle.
13: Let nCount ← 0
14: // At the beginning of a new consensus cycle, the list of neighbors is taken out and used as a reference for the change of neighbor nodes during the cycle.
15: Pre Neis ← GetNeighboursFromNDP()
while true do
17: // Received Consensus Flag from Authorized Nodes
18: if ConsensusFlag = true then
19: bReqireConsensus ← true
end if
21: // Multicast message packets to neighboring nodes.
22: Multi_cast(message\textsubscript{i}, curNeighbours)
23: // Number of times sent increases.
24: count ← count + 1
25: // Monitor and score the behavior of all neighboring nodes.
26: TrustState\textsubscript{i} ← AssessmentNeighbours(TrustState\textsubscript{i}, message\textsubscript{i})
27: // There are scores less than or equal to 10 in TrustState\textsubscript{i}, requesting to start consensus and reconfigure the network, and isolate the untrustworthy nodes.
28: if ExistUntrustedNeighbour(TrustState\textsubscript{i}) = true then
29: bReqireConsensus ← true
end if
31: // Obtain the latest neighbor node list of node ID\textsubscript{i}.
32: Cur Neis ← GetNeighboursFromNDP()
33: // If the change in the list of node ID\textsubscript{i} neighbor nodes is too large, request consensus to reconfigure the network.
34: if NeisChangingLarge(Cur Neis, Pre Neis) then
35: bReqireConsensus ← true
end if
37: // Send the current state transaction and reaccumulate the state records.
38: if bReqireConsensus = true then
39: TrustState\textsubscript{i}.curScore ← TrustState\textsubscript{i}.curScore/nCount
40: // Send StateData\textsubscript{i} to Delegated Agnet nodes ID\textsubscript{A}
41: StateData\textsubscript{i} ← TrustState\textsubscript{i} + Cur Neis
42: Multi_cast(StateData\textsubscript{i}, ID\textsubscript{A})
43: // Reinitialize the variables related to node local state collection and start a new round of trust assessment.
44: nCount ← 0
45: StateData\textsubscript{i} ← \{\}
46: TrustState\textsubscript{i} ← \{\}
47: Pre Neis ← GetNeighboursFromNDP()
end if
end while
The trusted reconfiguration process of the drone network is also the update of new blocks in the blockchain. The drone trusted network is essentially a logical network with a hierarchical structure embodied in a blockchain. The consensus that needs to be reached by the nodes is mainly the global trustworthiness of the nodes and the composition of the upper layer management network, i.e., the currently authorized nodes. All nodes involved in the task perform monitoring of neighboring nodes’ forwarding behavior (TBM) at the network layer based on the network configuration information of the local blockchain, score the behavior of neighboring nodes in real time (LSA), and establish their own local state transactions and send them to authorized nodes periodically. If the node is a member of the upper layer network, the authorized node, then in addition to performing TBM, has to receive the local state data sent by other nodes to complete the consensus for the current cycle. Unlike traditional blockchains, the local state data constitutes a transaction that is not used to trace the validity of future transactions; its purpose is to determine the current trustworthiness of the network nodes and therefore not to be saved to the blockchain. The recommended consensus algorithm is therefore a two-stage consensus algorithm and a data consensus (LSD_C) on the set of local state transactions, where the consensus result is a common generic subset of the set of local state transactions for all nodes in order of node ID size. The data consensus result is used as input to the decision consensus (GSA_C), which uses the global trust assessment stored in the Genesis block, the smart contract elected by the regional center to obtain the global trust of all nodes and the new authorized node group, and constructs a new block from it. After the blockchain system is updated, the drone nodes are re-enabled to the trusted network based on the configuration information in the new block.

### 4.1. Node Local State Transactions Generation

After system initialization is complete, the Genesis block stores the configuration information of the initial pre-defined upper layer network with a trust value of 10 for each registered node. A neighbor node forwarding behavior monitoring function is added to the intra-zone routing protocol (IARP), where IARP generates a neighbor table using the neighbor information collected by the neighbor node detection protocol (NDP) at the MAC layer, and periodically broadcasts outward the update packets of information such as neighbor table and zone radius at the same time, a forwarding behavior timer is started to record the forwarding behavior of neighboring nodes. The cycle of sending routing
message packets is to monitor the forwarding behavior of neighboring nodes, the behavior is divided into four categories.

1. Normal forwarding, the return packet is received within the threshold time and the data is not tampered with.
2. Delayed forwarding, received outside the threshold time but the data is not tampered with.
3. Data discarded, no return packet is received within a cycle.
4. Forwarding malicious, the data is tampered with.

The definition of the behavior given by the scheme and the criteria for the trust discount are shown in Table 2.

### Table 2. Classification of data forwarding behavior and penalty rules.

| Data Forwarding Behavior                  | Reputation Discount |
|------------------------------------------|---------------------|
| Normal forwarding                        | 0                   |
| Delayed forwarding                       | -1                  |
| not forwarding data                      | -2                  |
| Forwarding of incorrect data             | -3                  |

The pseudo-code 1 of the algorithm indicates the detection of the node’s forwarding behavior to neighboring nodes and the scoring of a given trust discount, as described in lines 14–35, and the output is the cumulative score of the trust discount for one data forwarding behavior of all neighboring nodes of the node.

Each node averages the neighbor trust discount records before the consensus mechanism is initiated and evaluates the local trust of its neighbor nodes based on the stage monitoring record set. Equation (1) indicates the local trust discount, where \( n\text{Count} \) is the number of forwarding actions. \( Discount_{i-j}^k \) is the discount value of node \( ID_i \)'s behavior to its neighbor \( ID_j \) for the \( k\_th \) time in this phase, and \( CurScore_{i-j}^{bc\text{-height}} \) is the average value of the discount in this phase.

\[
CurScore_{i-j}^{bc\text{-height}} = \frac{\sum_{k}^{n\text{Count}} Discount_{i-j}^k}{n\text{Count}}
\]

The final local state transactions are generated with a transaction data structure consisting of node \( ID_i \), a list of neighboring node IDs, and a list of discounted averages of the current phase trust of neighboring nodes. The final local state record takes the form of Formula (2).

\[
StateData_i = \{ID_i, [ID]_i, [ID]_j : CurScore_{i-j}^{bc\text{-height}}, \ldots\}
\]

where \( ID_j \) is a neighbor node of node \( ID_i \), \( [ID]_i \) is the current list of neighbor nodes of node \( ID_i \). When the period ends or the trigger event occurs, node \( ID_i \) sends the record to the upper network \( \{ID\}_j, j \in [M] \) and the consensus process starts. Algorithm 2 then gives the behavior of all nodes during the consensus cycle, with the nodes accumulating local trust loss and can be obtained by comparing the current neighbor table with the list of neighbors at the beginning of this cycle when the structure changes significantly, as shown in lines 20–25, and directly requesting consensus.
4.2. Node Local State Transactions Consensus

The system sets two ways to initiate the consensus process. First, the system sets a security cycle, 20 s for the simulation experiment, and the node cycle sends local state data to the upper network specified by the blockchain while initiating the next round of state data collection. Secondly, it is conditionally triggered to initiate the consensus process when two conditions occur in the cycle. (1) Each reissue of neighbor nodes is incorporated into the monitoring score, and when the cumulative average discount has fallen below the minimum trustworthiness threshold, indicating that there may be untrustworthy nodes in the network, the upper layer network is requested to initiate the consensus process. (2) The list of neighbor node IDs is used as the node’s feature vector, and IARP receives the neighbor node pushed by NDP each time. When the similarity between the two is less than a set threshold, it indicates that the network structure may have changed significantly and triggers the upper layer network to initiate consensus. The consensus cycle is reset after the conditional trigger consensus occurs.

Nodes multicast local state transactions to the upper layer network, \( \{ID\}^j_{M}, j \in [M] \), which consists of a set of authorized nodes specified by configuration information in the latest block. The upper-layer network performs data consensus on the collected Dataset of local state transactions, \( LSD = \{StateData_1, StateData_2, \ldots, StateData_N\} \), through the provably lightweight asynchronous consensus algorithm. The consensus result is a common public subset of \( LSD \).

Referring to the consensus algorithm [30–32] for asynchronous systems, we design the data consensus as a deterministic consensus. The main idea is that each delegated node forms a record set of collected transactions in order of node address size, and marks nodes that have no records. The record set is divided into equal parts according to the number of delegated authorized nodes forming the upper network (information of authorized nodes can be obtained from the blockchain), e.g., if the size of the network is \( N \) and the size of the upper network is \( M \), then each local state record subset size \( B = M/N \). Each authorized node sends its corresponding share of records in the upper network via a reliable broadcast transmission (RBC), e.g., \( ID_x \) and sends the blockchain provides smart contracts for external authentication and data validation, and when two-thirds of the authorized nodes confirm the transaction, a deterministic consensus is reached, which results in an asynchronous common subset \( (ACS) \) of the local state Dataset \( (LSD) \), \( ACS \subseteq LSD \), the result of the data consensus is then used by the authorized nodes for decision consensus to determine the new network configuration.

4.3. Trusted Network Reconfiguration

The ultimate purpose of blockchain introduced into the UAV network is to provide a trust guarantee, identify untrustworthy nodes, isolate them from the task network, and elect the optimal node to configure the upper network to achieve consensus and assist inter-zone routing. The generation of new blocks is the completion of a reconfiguration of the drone network. The conditions for the selection of the optimal node, one is that the node is trusted and the assisted intra-zone routing is to circumvent the selection of untrustworthy nodes; the second is that the current location is distributed in the center of the sub-zone to reduce the redundancy overhead of inter-zone routing.

4.3.1. Node Global Trustworthiness Assessment

By data consensus, the asynchronous generic subset \( ACS \) of all nodes is obtained, which contains the local state data of the node’s trustworthiness score of all its neighboring nodes, \( StateData_{i} = \{ID_{i}, [ID]_{\nu}, [ID]_{j} : CurScore_{i-j}^{bc-height}, \ldots\} \). The local state assessment \( LSA_{i} \) of a node is obtained by counting the asynchronous generic subset \( ACS \), i.e., the trustworthiness assessment of a node by all neighboring nodes, as \( LSA_{i} = \{ID_{\nu}, [ID]_{\nu}, [ID]_{j} : CurScore_{i-j}^{bc-height}, \ldots\} \). The global trustworthiness discount of nodes can be calculated based on \( LSA \), but the UAV network in the mission is a distributed Byzantine system.
and collusive or selfish behavior is not accurately determined in the above computational method. The global statistical computational method is used to identify colluding nodes, selfish nodes, and to correct for the global trustworthiness discount. The global trust discount of a UAV node ID \( x \) is the expected value of the local discount of all its neighboring nodes \( GDiscount^x \) with a variance of \( \sigma^x \), as shown in the following two equations.

\[
GDiscount^x = \left( \sum_{i=0}^{n} CurScore_{x-i}^{bcheight} \right) / n \tag{3}
\]

\[
\sigma^x = \sqrt{\frac{\sum_{i}^{n} (CurScore_{x-i}^{bcheight} - GDiscount^x)^2}{n}} \tag{4}
\]

where \( i \) is one of all \( n \) neighbors of node \( x \). The rules for the system to set the global trust degree discount are shown in Table 3.

| Reasons for Trust Discount | Global Discount |
|----------------------------|-----------------|
| \( Discount^x_{< 5 \sigma^x} \) | 0 |
| \( Discount^x_{\geq 5 \sigma^x} \) | -0.5 |
| Selfish node \( Discount^x_{error} \) | -1 |

According to the rule, Equation (5) corrects the global trust discount of node \( x \). Equation (6) calculates the current global trustworthiness.

\[
GDiscount^x = GDiscount^x + Discount^x_{error} \tag{5}
\]

\[
GReputation_{bcheight}^x = GReputation_{bcheight}^{x-1} + GDiscount^x \tag{6}
\]

when \( GReputation_{bcheight}^x \) is less than or equal to 0, it means that the node is also untrustworthy and its node ID and its authentication information are accumulated to the wrong node subvector commitment and isolated from the network. Refer to pseudo-code Algorithm 3 for details.
Algorithm 3 Function: Global Trust Assessment

Function: GlobalTrustAssess(\{LSD\}, ID_x)
1: let LSA[N][] ← {}  
2: /*Running in a Delegated Agent UAV ID_x */  
3: if \{StateData\}_x already contains local state transactions for 2/3 of the total number of nodes in the current network.  
4: if Nodes of ReceivedLocalStateTrans(\{StateData\}_x) ≥ 2/3 then  
5: #Extracts the corresponding decentralized transaction message blocks from the local transaction set.  
6: DRBC_x ← BuildMyRBCPacket()  
7: #Multicast its own DRBC_x to the delegate agent nodes.  
8: while 1 do  
9: DRBC_other ← ReceivedDBC formOtherAgents()  
10: #If 2/3 DRBCs are acknowledged, consensus is reached.  
11: ACS ← BuildACSFromStateData(LSD)  
12: #Statistical local assessment of all nodes  
13: #\{LSD\}_i: contains the trustworthiness scores of node ID_i for all neighboring nodes.  
14: #\{LSA\}_i: contains the trustworthiness scores of all neighboring nodes for ID_i itself.  
15: #Computing the assessment of ID_x by all neighboring nodes from \{LSA\}.  
16: GDiscount_x ← (\sum_{i=0}^{n} CurScore_{bcheight}^{x-i} − GDiscount_x) / n  
17: σ_x ← \sqrt{\sum_{i=0}^{n} (CurScore_{bcheight}^{x-i} − GDiscount_x)^2 / n}  
18: #Computing confidence variance of ID_x  
19: if |CurScore_{bcheight}^{x-i} | > (5 × σ^i) then  
20: DisCount_x ← 0.5  
21: end if  
22: if No Record from Neighbour ID_x then  
23: DisCount_x ← 1  
24: end if  
25: GDiscount_x ← (GDiscount_x + Discount_x + Discount_x error)  
26: GReputation_x^{bcheight} ← GReputation_x^{bcheight−1} + GDiscount_x  
27: output: GReputation_x^{bcheight}  

4.3.2. Zone Center Node Elections

The dynamic nature of the UAV network requires the upper layer management network to be time-varying, and the nodes that make up the upper layer network need to be not only trusted but also regionally representative. The upper layer network constructed by the trusted central nodes of each sub-region of the UAV network at each stage minimizes redundant routes. This solution uses clustering algorithms to regionally delineate the UAV network and find the regional centers. The clustering feature is a list of neighboring addresses of the UAV nodes. The feature UAV node i is represented as: \(N_i = [ID_{i1}, ID_{i2}, \ldots, ID_{im}]\), m is the number of its neighboring nodes, and if the number of neighbors is less than m, the missing part is filled with zeros. The UAV network is represented by a feature vector as \(\vec{U} = [\vec{N}_1, \vec{N}_2, \ldots, \vec{N}_n]\), with n being the number of
UAVs in the current mission network. The feature vectors used for clustering are not of numerical type, but are lists used for classification. Therefore a fuzzy K-modes clustering algorithm [33] is used, replacing the mean with the mode as the central node of the zone (cluster) and adapting to the situation of overlapping regions. Clustering makes use of simple matching dissimilarity, i.e., the dissimilarity between two UAV nodes is expressed in terms of the cumulative number of m neighbors of the feature vector, the fewer the mismatches, the closer the two nodes are. The mathematical expression (7) shows the proximity of two UAV nodes.

\[ d(\vec{N}_i, \vec{N}_j) = \sum_{x=1}^{m} \sum_{y=1}^{m} \delta(ID_{ix}, ID_{iy}) \]  

(7)

where \( \delta(ID_{ix}, ID_{iy}) = 0, \) if \( ID_{ix} \neq ID_{iy} ; \) \( \delta(ID_{ix}, ID_{iy}) = 0, \) if \( ID_{ix} = ID_{iy} \)

Let \( U^k = [\vec{N}_{k1}^1, \vec{N}_{k2}^1, \ldots, \vec{N}_{kn}^1] \) be a sub-zone of the UAV network, and the mode in the UAV network denotes the feature vector of the central node of the zone.

Definition 1. The feature vector \( Q = [ID_1, ID_2, \ldots, ID_m] \) is the mode of the UAV network \( U^k \), if it makes the function (8),

\[ D(Q, \vec{N}_i) = \sum_{i=1}^{m} d(\vec{N}_i, Q) \]  

(8)

take the minimum value and \( Q \in U^k \).

Let \( n_{ID_x} \) be the times the neighbor node \( ID_x \) appears in all lists of neighbors, the frequency of \( ID_x \) in the zone \( U^k \):

\[ f(ID = ID_x | U^k) = \frac{n_{ID_x}}{m} \]  

(9)

Theorem 1. Mode update method for k-modes of UAV networks, the function \( D(Q, N_i) \) reaches a minimum when and only when the following inequality holds:

\[ f(ID = ID_x | U^k) \geq f(ID = ID_j | U^k) \]  

(10)

where \( ID_x \neq ID_j, \forall j = (1, 2, \ldots, m) \). The relevant proofs of the theorem refer to Algorithm 4.

Algorithm 4 Proof of updating method for k-Modes of UAV networks

1: while \((m - -)\) do
2: if \( f(ID = ID_x | U^k) \geq f(ID = ID_j | U^k) \) then
3: if \( n_{ID_x} > n_{ID_j}, j \neq x, \forall j = (1, 2, \ldots, m) \) then
4: \( Q \cup ID_x \)
5: end if
6: end if
7: end while
8: \( D(Q, N_i) \) reaches a minimum.

4.3.3. Blockchain Update and Synchronization

The two-stage consensus data consensus of this scheme produces a deterministic consensus result, a common subset of local state transactions ACS, at the delegated agent nodes eventually through the decentralized stable transmission of DRBCs and external provable smart contracts. at this point, each correct delegated agent node is able to unify the algorithm for decision consensus, generate new configuration information and create new blocks. the delegated agent nodes update the local blockchain while multicasting new blocks according to the trusted nodes on the current blockchain. the node that receives the new block compares the height of the local blockchain, and if the difference in height
with the new block is greater than 1, the node requests synchronization of the blockchain from the delegated agent node that sent the block; otherwise, it verifies the new block and updates the local blockchain.

After the data consensus is completed, the upper layer network delegates the agent node to calculate the global state data of the nodes and process the global trust discount table and neighbor list table by statistics and clustering to update the trust degree of the network nodes, determine the regional central node, and reconfigure the trusted drone network in the current state. The drone network blockchain adaptively configures the trusted network periodically by continuously adding new blocks. The authorized nodes that make up the upper virtual network run a two-stage consensus process. The adaptivity is reflected in the network system cycle consensus, where any participating node directly requests the upper layer network to initiate consensus if it senses excessive changes in the physical structure or low local trustworthiness of neighboring nodes during the cycle. The process of reconfiguration of the trusted network is the continuous consensus process of the blockchain. The consensus flow is shown in Figure 3.

The delay of new blockchain block increase depends on the time of data consensus to reach an agreement, although the drone network will be frequent network partitioning, an asynchronous consensus algorithm can eventually reach a deterministic consensus. However, the original intention of this scheme design is to maintain the overall trustworthiness of the network by dynamically reconfiguring the network, so in order to meet the needs of practical applications, but the asynchronous consensus timeout, the consensus algorithm of multi-point proof of authorization (DPOA) is automatically started. DPOA is similar to the proof-of-authority consensus algorithm because the nodes participating in consensus in each cycle are the elected best state nodes, so after the asynchronous consensus timeout, the data consensus can be completed as long as there are more than 2 nodes reaching consensus, and the local transaction data set of the node with the smallest agent node ID among them is the final data consensus result. To ensure that the reconfiguration of the UAV network is completed within the specified safe time.

![Figure 3. Two-stage consensus process in an authorized node ID\(^j\).](image-url)
5. Scheme Feasibility Analysis

From the perspective of routing protocols, the BC_TZRP scheme is a method for secure routing of UAV nodes based on the blockchain’s record of node trustworthiness changes. From the perspective of network reconfiguration, the scheme is a blockchain system for adaptively reconfiguring the trusted network using node state changes. This blockchain system evaluates the trustworthiness of neighboring nodes in real time by monitoring the forwarding behavior of nodes through the ZRP protocol and records the changes in the trustworthiness of network nodes with a shared ledger. However, the timeliness of UAV network tasks and the light weight of UAV nodes require the blockchain system to be feasible in terms of operational overhead, scale management and security when reconstructing the trusted network in order to achieve a reasonable allocation of network resources and ensure the completion rate of tasks. Comparing the classical consensus algorithms POW, POS and PBFT, the following content analyzes the blockchain system in terms of computation, time complexity of communication, network scalability and security in conjunction with traditional blockchain application technologies.

5.1. Complexity Analysis

The consensus algorithm is the core of the blockchain and the main overhead of the system resources. The main purpose of this scheme is to dynamically reconfigure the network according to the nodes’ own state changes, including changes in trustworthiness, energy consumption and location, in order to ensure the availability and feasibility of the drone network. However, the drone nodes in the mission are limited in terms of bandwidth, computation, storage and energy resources, and the consensus algorithm must coordinate the complexity of computation, communication and storage to achieve a reasonable allocation of resources, especially energy consumption. The private blockchain created by the scheme in this paper, the main decentralized data asynchronous provable consensus algorithm used. The consensus algorithm uses a smart contract for authentication and transaction verification in the Genesis block to prove the authenticity of the source of the transaction and the integrity of the data with a computational complexity of $O(1)$. The POS and PBFT consensus algorithms do not need to compete for bookkeeping rights, and their computational complexity for verifying transactions is of constant order $O(C)$. The computational complexity of the POW consensus is $O(1/2N^2 + 2/3N)$, which is as high as $O(N \log N)$ even after immediate optimization, while the computational complexity of the POW consensus is mainly in the solution of the Hash problem. In terms of communication complexity, the nodes communicate with each other in a similar way as PBFT, and the consensus is completed when 2/3 of the nodes have confirmed the received transaction. The length of the data that requires consensus is $|m|$, and the required communication complexity is $O(|m|N^3)$. The decentralized asynchronous provable consensus method of this party divides the transaction data based on the number of proxy agents and the size order of their IDs, and uses decentralized stable broadcast control communication (DRBC), if the number of proxy agents is M and the number of nodes in the whole network is N, $M << N$, the communication complexity is $O(M|m|M^2 / N)$. In the POW or POS consensus algorithm, the communication complexity is $O(1)$, but its consensus result is probabilistic and requires multiple consensus rounds to determine, with Bitcoin’s POW requiring six consensus rounds to determine a transaction and POS requiring longer consensus rounds. Table 4 shows the performance comparison of different consensus algorithms in terms of their associated complexity.
Table 4. Consensus methods performance comparison.

| Consensus Algorithm | Complexity          | Deterministic Consensus |
|---------------------|---------------------|-------------------------|
|                     | Computation         | Communication           |                         |
| POW                 | $O(N \log N)$       | $O(1)$                  | No                      |
| POS                 | $O(C)$              | $O(1)$                  | No                      |
| PBFT                | $O(|m|N^3)$         |                         | Yes                     |
| Proposed Scheme     | $O(1)$              | $O(M|m|M^2/N)$           | Yes                     |

Note: POW and POS are probabilistic consensuses, and the determination of transactions requires multiple rounds of consensus, with POW requiring 6 rounds of consensus and POS requiring more. $M$ is the number of proxy agents, $N$ is the number of nodes in the whole network, $M << N$.

5.2. Scalability Analysis

Mission-oriented UAV networks use scale, node redundancy to guarantee mission completion rates in complex environments. Thus the scalability of the network is important. There are many factors that affect the scalability of the network, such as routing overhead, transmission delay, DOS attacks, etc. Storage resources also limit the scalability of the network when using blockchain as the management platform.

First, the routing overhead of establishing and maintaining is too large, which not only occupies too many computing resources, but also squeezes the bandwidth space for normal business data. Moreover, the large network range and many network nodes lead to excessive network transmission delay, which cannot meet the requirements of UAV network timeliness. Thus, the reconfiguration methods of UAV networks with planar architecture, such as OLSR, AODV, etc., do not have good scalability. The recommended scheme is based on ZRP routing protocol improvement, which combines intra-zone routing with a small delay, and inter-zone routing with low routing overhead to build a blockchain-managed hierarchical network architecture with good scalability.

Second, DOS attacks make the network process a large number of invalid requests, which seriously affects the availability of UAV networks, and the larger the network size, the more serious the impact. Thus, it is also a key factor limiting the scalability. BC_TRZP scheme builds the authentication environment of the UAV network, for which the requested nodes can be verified quickly with $O(1)$ complexity by identity vector commitment to stop DOS attacks and improve the scalability of the network.

Finally, the blockchain management platform can effectively provide node trust management, but the continuously growing blockchain ledger is also a challenge to the storage capacity of nodes. Traditional blockchain uses timestamp and hash chain to pack all transactions into blocks, which are uploaded to the chain after consensus. The transactions of the drone network are the nodes’ own state data, and the amount of data is generated all the time and keeps growing. The limited storage capacity of nodes also causes the network to be unscalable, because the larger the network size is, the more transactions there are. This scheme adopts a two-stage consensus approach, where the state data of the cycle is consensual and then processed for decision making. The decision result is only the node trustworthiness of very small size and configuration information such as proxy. This configuration information is used to build new blocks, while a large amount of local state data is discarded after decision consensus, thus guaranteeing the scalability of the UAV network.

5.3. Security Analysis

Security is necessary for UAV networks in insecure environments. The blockchain system in this solution provides the function of node authentication, and each node is assigned public and private keys, a unique identity ID, and an identity witness of the existence network to prevent unauthorized access by external nodes. At the same time, the blockchain keeps updating the trustworthiness of the network nodes and isolates the network from legitimate but untrustworthy nodes to maximize resistance to black
hole attacks, gray hole attacks, impersonation attacks, multiple identity attacks, etc. The statistical analysis of the global trustworthiness of block nodes can also detect selfish nodes and malicious lying nodes and prevent collusion attacks by giving high scores to each other. All of the above enables the BC_TZRP scheme to ensure the trustworthiness of participating network nodes through adaptive reconfiguration of the network, which in turn guarantees the security of the UAV network during the entire mission.

The UAV network uses a fuzzy K-Modes clustering approach to cluster the list of neighbor IDs provided by the Neighbor Node Detection Subprotocol (NDP) to select the trusted central nodes in the partition. These central nodes constitute the upper logical network for consensus operations and for completing inter-partition routing. Through each consensus of the blockchain, the members of the upper logical network are updated periodically. This not only increases the scalability, but also allocates resources rationally, improves the lifetime of the whole network, and ensures the safe use of the UAV network.

6. Simulation Experiments and Effectiveness Analysis

The dynamic reconfiguration of UAV trusted networks relies on the blockchain-assisted Trusted Zone Routing Protocol (BC_TZRP), which is implemented as new blocks are chained during the growing blockchain. Through simulation experiments, the performance of the classical Ad Hoc network routing protocol in terms of data transmission delay, packet delivery rate and routing overhead is verified and compared with that of the classical Ad Hoc network routing protocol when the number of error nodes gradually increases.

The simulation software uses Qualnet network simulation software, which uses the standard OSI seven-layer model framework and is optimized for wireless mobile communication networks. During the simulation, the behavior of each node of the network is calculated independently to match realistic network operation and provides detailed and varied statistical data analysis functions. This paper designs mission scenarios, the scene size of the simulation experiment is $1000 \times 1000 \text{ m}^2$, the number of UAV nodes is 100, the simulation duration is 210 s, the number of data links is 30, the node movement speed is 0–30 m/s, the dwell time is 30 s, the packet sending interval is 500 ms, the malicious nodes are incremented by 0, 5, 10, 15, 20, 25, 30, the MAC layer protocol is 802.11b, and the wireless transmission range is 400 m. Each test protocol was run three times with different random numbers, and the average of the three runs was used as the basis for flat evaluation. Different random numbers indicate different trajectories of nodes in the network. The error nodes in the experiment mainly consist of compromised legitimate nodes, whose forwarding behavior is dangerous in the sense of deliberately dropping packets or forwarding inconsistent data; selfish nodes, which only actively send data and do not forward it; and failure points.

Packet Delivery Rate: The packet delivery rate is the ratio of packets successfully received by the destination node to those sent by the source node. Figure 4 represents the change in the guaranteed delivery rate of the four routing protocols as the number of error nodes increases.

It is clear that AODV has the highest delivery rate when there are no error nodes, and the other protocols have similar delivery rates, but as error nodes continue to be created, the blockchain-assisted trusted area routing protocols remain largely unchanged, but the delivery rates of OLSR, AODV and ZRP drop sharply and eventually collapse. The reason for this is that BC-TZRP can isolate the largest number of untrustworthy nodes from the network and dynamically reconfigure the upper logical network used for consensus, thus providing high fault tolerance. Moreover, as the number of erroneous nodes increases, the network’s erroneous routing information leads to non-stop maintenance and reconfiguration of road routes under the other three protocols, resulting in a high rate of delivery degradation.
Routing Overhead: The routing overhead is the number of routing control packets sent out by all nodes in the same condition. Figure 5 shows the status of routing overhead for each routing protocol under different error nodes. In the absence of errant nodes, both ZRP and AODV have low routing overhead and OLSR has the highest routing overhead, but as errant nodes appear the overhead of OLSR, AODV and ZRP increases rapidly and OLSR is the first to crash as the routing overhead exhausts the radio bandwidth, but BC_TZRP stays at a low level and shows a continuous downward trend due to the isolation of untrustworthy nodes. For the same reason, erroneous nodes generate a lot of invalid routing information, leading to an increase in the routing overhead of classical routing protocols, whereas the nodes involved in route construction and maintenance in BC_TZRP are trusted, avoiding the interference of invalid routing information, and the routing overhead remains largely unchanged.

The average end-to-end delay refers to the time elapsed from the time the packet leaves the source node until the packet arrives at the destination node. As can be seen in Figure 6, in the absence of error nodes ZRP has the lowest latency, OLSR is at the low end and AODV has a higher latency, again due to interference from error nodes, the average end-to-end latency of all three rises rapidly and when error nodes exceed 25, the network system is in a state of collapse and the latency tends to infinity. BC_TRZP maintains the actual trustworthiness of the network performing the task, and by completing inter-zone routing through the upper layers of the network consisting of the regional centers, the average end-to-end delay was kept at a small level.
The purpose of the BC_TZRP scheme implementation is to reconfigure the new network based on the state of the nodes, through the records of the blockchain. The main component of reconfiguring the network is to isolate malicious nodes from the network. These malicious nodes are not in the new network, there are not involved in routing and data forwarding, so the changes in Figures 5 and 6 are minor. On the other hand, the consensus mechanism is set when the node finds that the neighboring nodes are not trustworthy, it immediately requests the system to initiate consensus to isolate these wrong nodes in time. The behavior of the malicious node set in the experiment is to tamper with the forwarding information, and the discount of its trustworthiness is relatively large, and it is quickly determined as an untrustworthy node and expelled from the task network. So the actual of malicious nodes to do evil is very short and does not cause much harm to the network before being isolated.

The routing protocols of classical mobile Ad Hoc networks do not take into account the security of the routes. Although blockchain network systems provide authentication capabilities for UAV networks that can prevent unauthorized access by external malicious nodes, the complex mission environment has not only selfish and faulty nodes, but also the risk of Byzantine node generation. Experiments in this scenario were set up with varying proportions of erroneous nodes where compromised internal nodes (drones with legitimate identities) would initiate malicious tampering and packet drops. The blockchain-assisted trusted hierarchical routing protocol effectively maintains the trustworthiness of the UAV network during the mission process.

The decentralized and de-trusted blockchain system is the best choice in terms of global trustworthy management and rational use of network resources for distributed drone networks running in complex environments. However, unlike the traditional blockchain, drones as blockchain nodes have limited resources, and the consensus algorithm of the blockchain needs to be redesigned to meet the objective environment of asynchronous, lightweight, and dynamic generation of error nodes. The following content designs experiments for the lightweight storage and energy consumption problems of blockchain.

Blockchain Storage: The blockchain itself is a non-stop growing shared chain database to provide validation of transactions with the tamper-proof nature of historical data, thus requiring high storage capacity of blockchain nodes. An experimental scenario is designed with a network size of 100 nodes, with each node submitting transactions randomly within 5 s, no malicious nodes, and a 20-s cycle consensus. Using the Delegated Proof of Stake (DPOS) consensus algorithm, 21 nodes are designated to take turns to keep score and compare the storage consumption of blockchains with transaction packet sizes of 50 bytes and 100 bytes, respectively. The results of the simulation experiment are shown in Figure 7.
Figure 7. Blockchain Storage Growth Comparison.

Obviously, the BC_TZRP scheme has a small growth rate and is independent of the size of the transaction because it uses a two-stage consensus and the blockchain only keeps the consensus result of the decision, while the traditional blockchain of DPOS has a consensus for each block of the order taking consensus and all the transaction history data is saved to the blockchain, which has a large data growth rate and requires larger storage of blockchain time as the volume of transactions increases.

Energy Consumption: The energy consumption of the drone network is a key issue. The energy of a blockchain system is mainly consumed during the consensus computation in communication and computation overhead. An experimental scenario with 100 drone nodes is designed with ZRP routing concordance compared to a traditional blockchain. The POW consensus algorithm adjusts the number of zeros and transactions in the header of the requested hash to the specific test environment, which can result in a POW calculation time of roughly 20 s. The POS and PBFT consensus algorithms ensure that the consensus time is determined by setting the number of transactions. We provide the two-stage asynchronous consensus algorithm used in the bill. If the preferred asynchronous consensus algorithm fails to reach an agreement within the specified time, the proof-of-authority consensus algorithm is activated to ensure consensus within 20 s. The experiments track the communication and computation overhead of the consensus algorithm under different error nodes and translate the communication and computation overhead into energy consumption. The local state transactions obtained from real-time monitoring of the forwarding behavior of ZRP are subjected to consensus, and the overall network energy consumption is viewed under different experimental scenarios against POW and PBFT consensus algorithms, respectively. The results of the simulation experiment are shown in Figure 8.

Due to the presence of malicious nodes, the traditional blockchain scheme generates invalid routing information during the routing process, increasing the communication and computation overhead. For the POW consensus algorithm, although the communication overhead is small, all nodes consume arithmetic power by participating in the competition for bookkeeping rights, so the consumption is the largest. The POS consensus algorithm does not need to consume arithmetic power, but with the increase in malicious nodes, its invalid routing process also increases the overall network consumption. The PBFT also increases energy consumption due to the increase in malicious nodes and the increase in view switching frequency in the consensus process. The BC_TZRP scheme reconfigures the network periodically to exclude untrustworthy nodes from the network, minimizing the generation of interfering routes, and replacing the agent nodes used for consensus delegation every cycle is a more reasonable resource allocation for the whole network, so the consensus overhead is basically the same in the presence of different numbers of malicious nodes.
Figure 8. Comparison of energy consumption in a consensus process with different numbers of malicious nodes.

7. Conclusions

This solution combines the operational process of the blockchain system with the dynamic update of the UAV network routing, where the evaluation of the data forwarding behavior of nodes to their respective neighboring nodes during intra-zone routing is combined with the list of neighboring nodes to construct node local state transactions. The periodically elected central node phase of the zone is used by the upper layer network to reach consensus on the local state of the network’s prime nodes. The consensus results are used to perform a global trust assessment of the nodes, obtain new zone centers through clustering, generate new blocks, update the blockchain and then reconstruct the upper layer network. The tasking process of the UAV network is also a recording process of the blockchain system for node state migration and the continuous reconfiguration of the UAV network, with the aim of identifying and isolating untrustworthy nodes in a timely manner and maintaining the overall performance of the UAV network during the tasking process, which has also been proven to be effective in simulation experiments.

Author Contributions: Conceptualization, L.K.; Data curation, L.K.; Investigation, F.H.; Project administration, B.C.; Writing—original draft, L.K.; Writing—review & editing, F.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China, under Grant 2019YFB2102002; in part by the National Natural Science Foundation of China, under Grant 62176122, 62001217; in part by A3 Foresight Program of NSFC, under Grant No. 62061146002.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.
References

1. Army Day 2021: Offensive Swarm Drone System Demonstrates Capabilities for the First Time. 2021. Available online: https://timesofindia.indiatimes.com/videos/news/army-day-2021-offensive-swarm-drone-system-demonstrates-capabilities-for-the-first-time/videoshow/80282977.cms (accessed on 15 January 2021).

2. Ferreira, R.; Gaspar, J.; Sebastião, P.; Souto, N. A Software Defined Radio Based Anti-UAV Mobile System with Jamming and Spoofing Capabilities. Sensors 2022, 22, 1487. [CrossRef] [PubMed]

3. Cheng, F.; Liang, Z.; Peng, G.; Liu, S.; Li, S.; Ji, M. An Anti-UAV Long-Term Tracking Method with Hybrid Attention Mechanism and Hierarchical Discriminator. Sensors 2022, 22, 3701. [CrossRef] [PubMed]

4. Zhao, J.; Wang, G.; Li, J.; Jin, L.; Fan, N.; Wang, M.; Wang, Y.; Yang, T.; Deng, Y.; Guo, Y.; et al. The 2nd Anti-UAV Workshop & Challenge: Methods and Results. arXiv 2021, arXiv:2108.09909.

5. Jiang, N.; Wang, K.; Peng, X.; Wang, Q.; Xing, J.; Li, G.; Zhao, J.; Guo, G.; Han, Z. Anti-UAV: A Large Multi-Modal Benchmark for UAV Tracking. arXiv 2021, arXiv:2101.08466.

6. Brust, M.R.; Danoy, G.; Stolfi, D.H.; Bouvry, P. Swarm-based counter UAV defense system. Discov. Internet Things 2021, 1, 1–19.

7. Clausen, T.H.; Jacquet, P. Optimized Link State Routing Protocol (OLSR). RFC 2003, 3626, 1–75. [CrossRef]

8. Pei, G.; Gerla, M.; Chen, T. Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks. In Proceedings of the 2000 IEEE International Conference on Communications: Global Convergence Through Communications, ICC 2000, New Orleans, LA, USA, 18–22 June 2000; pp. 70–74. [CrossRef]

9. Hasan, M.A.; Ullah, S.I.; Khan, I.U.; Shah, S.B.H.; Salam, A.; Khan, A.W.U. Unmanned Aerial Vehicles Routing Formation Using Fisheye State Routing for Flying Ad-hoc networks. In Proceedings of the ICFNDS’20: The 4th International Conference on Future Networks and Distributed Systems, St. Petersburg, Russia, 26–27 November 2020; ACM: New York, NY, USA, 2020; pp. 9:1–9:7. [CrossRef]

10. Perkins, C.E.; Belding-Royer, E.M.; Das, S.R. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 2003, 3561, 1–37. [CrossRef]

11. Pandey, P.; Singh, R. Efficient Route Selection Scheme in MANET Using Enhanced AODV Protocol. Wirel. Pers. Commun. 2022, 123, 959–974. [CrossRef]

12. Qin, L.; Kunz, T. Pro-active route maintenance in DSR. ACM SIGMOBILE Mob. Comput. Commun. Rev. 2002, 6, 79–89. [CrossRef]

13. Ko, Y.; Vaidya, N.H. Location-Aided Routing (LAR) in mobile ad hoc networks. Wirel. Netw. 2000, 6, 307–321. [CrossRef]

14. Gasmi, R.; Aliouat, M.; Seba, H. A Stable Link Based Zone Routing Protocol (SL-ZRP) for Internet of Vehicles Environment. Wirel. Pers. Commun. 2020, 112, 1045–1060. [CrossRef]

15. Sinha, P.; Krishnamurthy, S.V.; Dao, S.K. Scalable unidirectional routing with zone routing protocol (ZRP) extensions for mobile ad-hoc networks. In Proceedings of the 2000 IEEE Wireless Communications and Networking Conference, WCNC 2000, Chicago, IL, USA, 23–28 September 2000; pp. 1329–1339. [CrossRef]

16. Lv, X.; Le, X.; Ding, K. An Improved Cluster Routing Algorithm Based on ZRP Protocol. In Proceedings of the Wireless Internet—10th International Conference, WiCON 2017, Tianjin, China, 16–17 December 2017; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Li, C., Mao, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 230, pp. 261–271. [CrossRef]

17. Zhou, B.; Lee, Y.; Gerla, M.; Rango, F.D. Geo-LANMAR: A scalable routing protocol for ad hoc networks with group motion. Wirel. Mob. Comput. 2006, 6, 989–1002. [CrossRef]

18. Lee, Y.; Chen, J.; Hong, X.; Xu, K.; Breyer, T.M.; Gerla, M. Experimental evaluation of LANMAR, a scalable ad-hoc routing protocol. In Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC 2005, New Orleans, LO, USA, 13–17 March 2005; pp. 2032–2037. [CrossRef]

19. Ourouss, K.; Naja, N.; Jamali, A. Defending Against Smart Grayhole Attack Within MANETs: A Reputation-Based Ant Colony Optimization Approach for Secure Route Discovery in DSR Protocol. Wirel. Pers. Commun. 2021, 116, 207–226. [CrossRef]

20. Anand, A.; Rani, R.; Aggarwal, H. A Security Model based on Reputation and Collaboration through Route-Request in Mobile Ad Hoc Networks. KSII Trans. Internet Inf. Syst. 2015, 9, 4701–4719. [CrossRef]

21. Choudhury, P.; Kumar, M.P.; Agarwal, R.K.; De, T. Reputation Based Resistance for Route Request Flooding Attack in MANETs. In Proceedings of the 2010 International Conference on Wireless Networks, ICWN, Las Vegas, NV, USA, 12–15 July 2010; Arabinia, H.R., Clinchy, V.A., Deligiannidis, L., Solo, A.M.G., Eds.; CSREA Press: Las Vegas, NV, USA, 2010; Volume 2, pp. 240–245.

22. Oliviero, F.; Romano, S.P. A Reputation-Based Metric for Secure Routing in Wireless Mesh Networks. In Proceedings of the Global Communications Conference, 2008. GLOBECOM 2008, New Orleans, LA, USA, 30 November–4 December 2008; pp. 1935–1939. [CrossRef]

23. Guillaume, L.; Van de Sype, J.; Schumacher, L.; Di Stasi, G.; Canonico, R. Adding reputation extensions to AODV-NU. In Proceedings of the 2017 17th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT2017), Enschede, The Netherlands, 24–25 November 2010; pp. 1–6.

24. Azza, M.; Hacene, S.B. An Enhanced Reputation-based for Detecting Misbehaving Nodes in MANET. Int. J. Wirel. Microw. Technol. (IJWMT) 2017, 7, 28–37. [CrossRef]

25. Jarjis, A.; Kadir, G. Blockchain Authentication for AODV Routing Protocol. In Proceedings of the 2020 Second International Conference on Blockchain Computing and Applications, BCCA 2020, Antalya, Turkey, 2–5 November 2020; Alsmirat, M.A., Jararweb, Y., Alooqaily, M., Özkasap, Ö., Eds.; pp. 78–85. [CrossRef]
26. Careem, M.A.A.; Dutta, A. Reputation based Routing in MANET using Blockchain. In Proceedings of the 2020 International Conference on COMmunication Systems & NETworkS, COMSNETS 2020, Bengaluru, India, 7–11 January 2020; pp. 1–6. [CrossRef]

27. Goka, S.; Shigeno, H. Distributed management system for trust and reward in mobile ad hoc networks. In Proceedings of the 15th IEEE Annual Consumer Communications & Networking Conference, CCNC 2018, Las Vegas, NV, USA, 12–15 January 2018; pp. 1–6. [CrossRef]

28. Li, G.; He, B.; Wang, Z.; Cheng, X.; Chen, J. Blockchain-enhanced spatiotemporal data aggregation for UAV-assisted wireless sensor networks. IEEE Trans. Ind. Inf. 2021, 18, 4520–4530. [CrossRef]

29. Islam, A.; Al Amin, A.; Shin, S.Y. FBI: A federated learning-based blockchain-embedded data accumulation scheme using drones for Internet of Things. IEEE Wirel. Commun. Lett. 2022, 11, 972–976. [CrossRef]

30. Arnold, R.; Longley, D. Continuity: A deterministic Byzantine fault tolerant asynchronous consensus algorithm. Comput. Netw. 2021, 199, 108431. [CrossRef]

31. Li, K.; Hua, C.; You, X. Distributed Asynchronous Consensus Control for Nonlinear Multiagent Systems Under Switching Topologies. IEEE Trans. Autom. Control 2021, 66, 4327–4333. [CrossRef]

32. Knudsen, H.; Li, J.; Notland, J.S.; Haro, P.H.; Ræder, T.B. High-Performance Asynchronous Byzantine Fault Tolerance Consensus Protocol. In Proceedings of the 2021 IEEE International Conference on Blockchain, Blockchain 2021, Melbourne, Australia, 6–8 December 2021; Xiang, Y., Wang, Z., Wang, H., Niemi, V., Eds.; IEEE: Manhattan, NY, USA, 2021; pp. 476–483. [CrossRef]

33. Kuo, R.; Zheng, Y.R.; Quyen, N.T.P. Metaheuristic-based possibilistic fuzzy k-modes algorithms for categorical data clustering. Inf. Sci. 2021, 557, 1–15. [CrossRef]