UAV Optimizations

Decentralization of UAV and Drones

The current state-of-art for UAVs is a centralized system with a server and most probably a supervising controller. However, decentralized solution is a blockchain based implementation where drones act as nodes, and communicate with each other using an immutable ledger. Further, the development of `Smart-Contracts` made it possible to run complex logic that can bring serverless automation.

In this synopsis of multiple blogs, articles and papers, we will explore how Decentralized Solution using Block-Chains can make the currently working solution of Centralized autonomous drones to higher and might I say, seemingly far-fetched futuristic.

Why Decentralize

1. FASTER COMMUNICATION

Their decentralized algorithm requires what they say is significantly lower communications bandwidth, as well as lower computation cost, thanks to the distributed way it makes robots share intel on obstacle-free regions in their immediate vicinity.

Instead of each robot broadcasting to every other robot a complete map of safe space around it, the decentralized algorithm has robots only share maps with their immediate neighbors and also has each calculate where neighbors' maps intersect with their own — sharing only relevant intersected data on to the next neighbor.

So the idea is that, collectively, the team of robots maintains a comprehensive map of safe terrain while reducing the communication data needed to keep the swarm moving. It is like everyone tells their neighbor about what they know, and they propagate only what is needed to be known by their neighbors.

This results in faster communication, and even though decentralized networks generally have extremely high latency, the introduction of 6G has provided a solution for low-latency communication. Further, multi-casting to immediate neighbors rather than broadcasting to all decreases the communication and processing data further.

Plus, most of the drone applications including Traffic Monitoring, crowd control, surveillance, package-delivery etc. should be done on a private or federated blockchain. So latency for these can be made much lesser than the often familiar public Block-Chains by fine-tuning.

2. PRIVACY AND SECURITY

The concern of privacy and security of data as well as the drones themselves can be seen as an added (although very much intentional) advantage of integration of Blockchain protocol.

Security

We will discuss multiple potential attacks and how Blokchain protocol prevents them in UAVs. This is an analysis done by [3]. * `Fabrication` : Data-Fabrication is feeding of false data to the drones, and this type of attack is prevented by the immutable nature of the ledger, and the consensus mechanism to append any data.
Notice this also prevents any data-flooding attacks, to overflow the limited storage capacity of the drones local storage.

- `DDoS` : Pinging with multiple requests to a drone can be a way to consume all bandwidth of the drones and prevent inter agent communications. The trivial way to tackle this problem would be with a dedicated DDoS mitigation service. However, this would consume precious computing resources and time of the system.
  Since BlockChain itself is a decentralized network of nodes, it would be extremely inefficient to attack the network, as the pings would get distributed among all nodes, and flooding with requests would mean to flood all drones with requests, and that is extremely unlikely. Even more so with the proposed 1THz bandwidth of 6G.
- `JAMMING` : This is an attack involving radio blasts to distupt communications by increasing noise. However, in blockchains, the cryptographically-signed, single chain concept (CONFIRM THIS ONCE) prevents any such attack by allowing only relevant data to enter the ledger.

PRIVACY

Data Privacy is maintained by the Block-Chain Protocol. It can use efficient protocols, by the intrinsic modular architecture of the protocol, and inculcate protocols such as the ECC, RSA etc. The Data thus will be encrypted and computationally unfeasible to crack.

Moreover, the Block-Chain network itself would be either Private or Federated, which means no unauthorized node can join the network and clone the data.

However, for security, it would be wise to install a `compromised script`, that would run when a drone is compromised (physically captured / opened without authorization), so that it would erase its entire data and all its credentials and keys irreversibly. This would be more useful for Military Applications for National Security.

3. AUTOMATION USING SMART CONTRACTS

With the addition of Smart Contracts to the Block-Chain protocol, it has become a realizable task to execute some code on the ledger. Now, the complexity of code certainly increases computational, memory and other overheads, but it is not a bottleneck for most cases of automation in drones.

The Paper [2] considers smart-contracts to achieve the following purposes :
* detect status of other drones in the swarm and other IoT devices if monitoring
* it discusses an IoT based agriculture scenario where drones can be used to detect faulty sensors and other equipments by running smart contracts to detect erroneous messages, low battery or otherwise. * this example can be extended to our case of Urban-City where monitoring drones can detect erroneous messages throughout the city and inform / perform repairs. * another implementation of smart contracts could be to synchronize flight of swarm, individual movements, take-off and landing, geo-spacial mapping of fellow-drones of the swarm and other operations that might be needed to run.

The following image explains one of the applications of the smart-contracts of drones, in IoT based agriculture and the drones use in detection and/or maintaining of erroneous IoT devices.

4. SPATIOTEMPORAL INDEPENDENCE

5. OTHER PERFORMANCE OPTIMIZATION

Decentralization using Block-Chains usually leads to excellent optimization in work throughput when compared to centralized architectures. It is analysed in [2] over the IoT based agriculture maintainance via drones. The results show significant contrast when shown over large range and capacity.

BLOCKCHAIN ARCHITECTURE

The Blockchain Data Structure

A brief introduction to the block-chain data structure. It is an append-only singly-linked link list with each node having a hash-pointer based tree (known as the merkel-tree), which stores the data. Along with this, there is Hash of the previous block/node, nonce, and other metadata in the block/node's head.

The following image gives a brief explanation of the data structure.

Types of Networks

Blockchain has multiple types of networks based on accessability and control. They are :

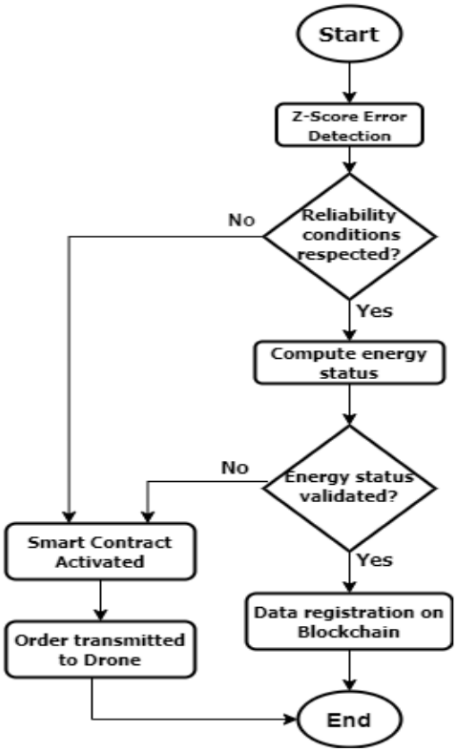
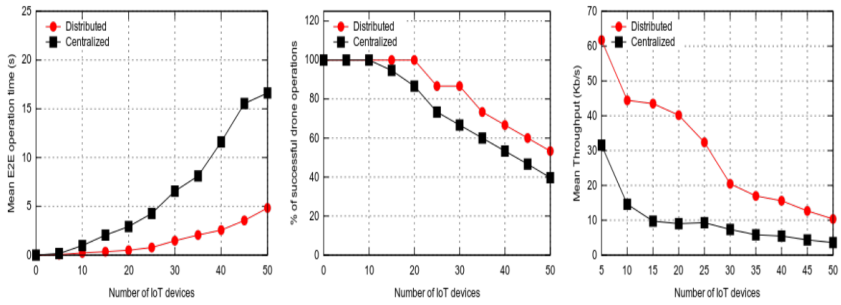


Figure 1: Drones Smart Contract Execution Flowchart (Agriculture Case Example)[2]



(a) Mean E2E operation time (b) Percentage of successfull drone operations (c) Mean throughput

Figure 2: Centralized vs (Blockchain based) Decentralized architecture comparasion

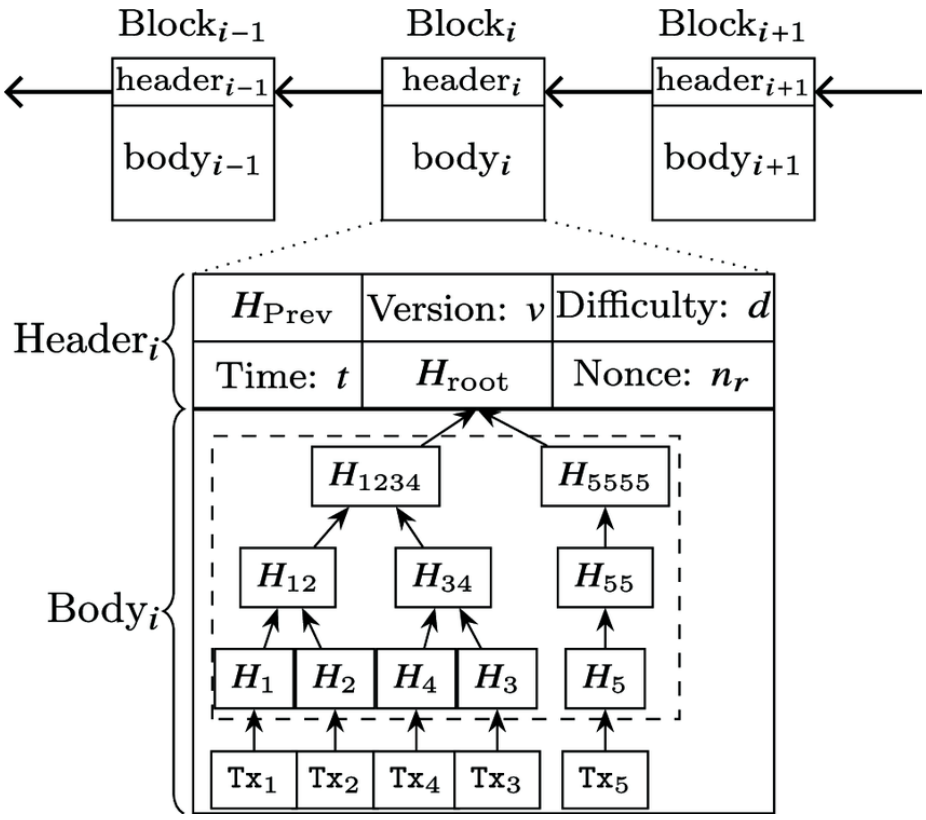


Figure 3: Block-Chain Data Structure

PUBLIC NETWORK - This network has no restrictions, and any person with an internet connection can join the network. In this case, there is advantage of transparency, trust, cutting off any middleman and security. However, the disadvantages include scalability issue, excessive power and memory consumption and lack of transitional speed.
Noticing this, we conclude that transparency is to be avoided in most applications of UAVs, and trust is a non-essential advantage. However, power-consumptions and memory limitations are huge constraints that can drastically affect UAV performance and throughput.

PRIVATE NETWORK - This network is usually used within an organization where only certain members are allowed to join the network. The disadvantages include less secure, single point of failure and problems in achieving trust. However, it has high Transactions per second, and highly scalable.

CONSORTIUM / FEDERATED NETWORK - This is a hybrid of the public and private blockchains. It provides the access control, while preserving the decentralized nature of the network.
The advantage of such a network is its access control, higher scalability and transactions per-second than public network and overall higher throughput. The disadvantages are extremely low, including less transparency and annonymity.

Analysing the networks, we notice that consortium based network is the best option to implement in case of UAVs, for various different scenarios.

Notice another minor optimization. For access control, we can use smart-contract based authentication instead of server side blacklisting/whitelisting mechanism which is centralized.

The 6G based BlockChain Communication Scheme

The proposed scheme is divided into 5 Layers, namely : 1. Data Sensing Layer 2. Block-Chain Layer 3. UAV Layer 4. Application/Control Layer 5. Communication Layer

We will explore the layers one by one :

DATA SENSING LAYER
This Layer consists of receptors to receive data from IoT based devices, and data collected by the UAVs own sensors, which is of different form and from different sources. Notice that this data is represented by the set {S1,S2,...Sn}.
Now, we treat each unit of data as a transaction digitally signed by the receiver node UAV. Hash-Tree of these transactions are computed and the merkel tree is constructed. Notice that time-stamp of the received data is also noted, as it will lead to avoidance of any clashes later on.

Now, all nodes can periodically append a block to the ledger, containing the data it has collected. Repeated data can be deleted by individual nodes via dedicated script that runs whenever a new block is added.

BLOCK-CHAIN LAYER

As already discussed, the blockchain layer will have a Consortium Blockchain Network. The advantages of features like smart-contract would be the elimination of trust on any 3rd parties. Consensus is reached by any of the consensus protocols among PoW, PoS, PoC, delegated-PoS etc. Each of the following consensus protocols have their own advantages and limitations. Due to the versatile nature of the Smart Contracts of Ethereum protocol, we propose to use it.(Discuss after studying all of them)

One of the problems in using Ethereum is the high transaction/storage costs. To overcome such a high storage cost issue in Ethereum, an IPFS protocol is used in the proposed scheme, which is immutable, distributed, and free of cost data storage. It accelerates the data downloading speed with the hosting of data parts to other peers by exploiting minimum network bandwidth[3].

UAV LAYER

In this layer, UAVs are flying to sense data from the D_sl, which it can read/retrieve with extremely low latency as the ledger where D_sl is stored. The UAVs have three components of velocity, the angles (,). These parameters is better to be maintained either using a separate dedicated processing unit, because it will have better performance. However we can make use of multiprocessing capacity of existing hardware and make do.
Now, we can see velocity components [v_x(t),v_y(t),v_z(t), ,]. We can use normal physics and laws of motions, but I think this is not within the scope of discussion.

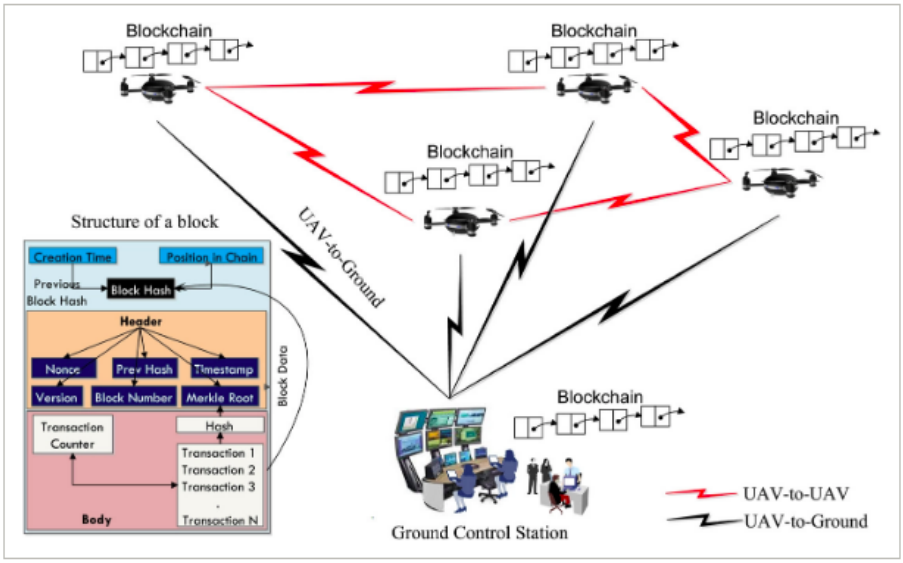


Figure 4: 6G-BlockChain Network with an observer

`APPLICATION AND CONTROL LAYER`
The Application/Control layer can read and process the ledger information. It also allows the Applications, RL Models, and manual controls to be ran from the control centre in case of non-autonomous or semi-autonomous drones.
Other than this, we can also attain data from the ledger from time to time and perform all sorts of operations, analysis etc. on it based on our applications.

`COMMUNICATION LAYER` The communication layer can be devised of either of 5G and 6G, but it is more efficient if we use 6G. It adds the advantages : * Ultra high latency * Extremely high latency * Extremely high reliability * 1 cm 3D position precision * Wide Range (10^7 km)

The most important factor is the latency issue : * for LTE-A latency <= 20ms * for 5G latency <= 5ms * for 6G latency <= 0.1ms

This exceptional low latency is preferred because the blockchain layer, as it is will add a little time delay no-matter how privatized or optimized. This layer is just for inter-drone and control-center transmission of data.

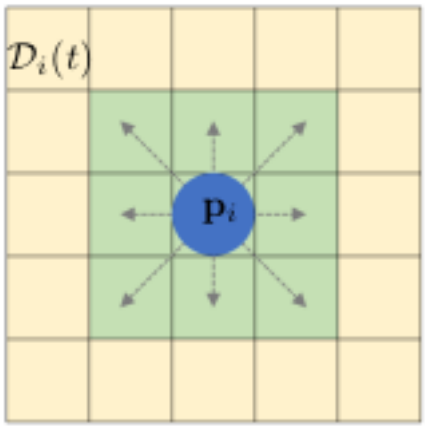
The BlockChain based Autonomous Drone Protocol Scheme

Decentralized Routing Algorithm

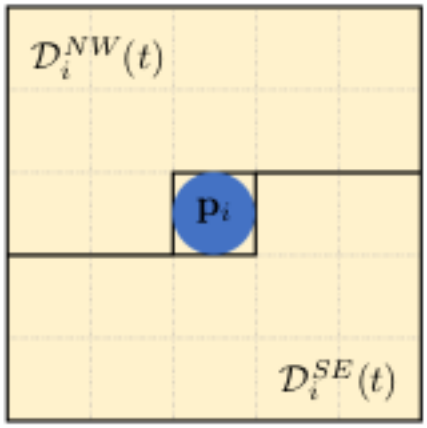
In the first step towards realizing the decentralized motion of drones, we explore a collision-avoidance, multi-agent routing algorithm. In this, we use concepts from Cellular Automaton[5], to devise a path-finding algorithm.

Following is a small walkthrough through the algorithm :

- We first divide space into grids. Then, we define an observable and movable space for 1 timestamp. It is natural to have `observable range > movable range`. We thus define movable space as 1 grid neighborhood of the agent, and the observable space as 2 grid neighborhood of the agent.



- Now, all our analysis will revolve around this space. First we divide the region into two parts, the `Agent-Detection Regions`, which include Region `North-West` and `South-East`. This is linked with a time-synchronized function, which at any time `t`, returns same value (either `N-E` or `S-W`). Let us call this function `direction-detection`.



- Now, we define 3 rules :
  `RULE 1 :` Don't move into an occupied cell
  `RULE 2 :` Don't move into a cell if it falls in another agent's movable region, and the agent is in high priority region. Here, high priority region is the `North-West` or `South-East`, depending on the output of the direction-detection function.
  `RULE 3 :` Move to a cell such that the `Chebyshev distance` to the destination is shortened the most.
  `RULE 4 :` Iterate through all possible points, the drone can move at, and move the drone to the one which minimizes the `Euclidean Distance`. If

no such point exists, don't move the agent.

- Now, the algorithm is as follows :
  - For each timestep :
  - update direction-detection function
  - check if RULE 1 and RULE 2 satisfies
  - if RULE 1 and 2 satisfies, move according to RULE 3
  - otherwise move according to RULE 4
  - increment timestamp

The implementation of the Agent Class, where all the code is implemented, highlighting the decentralized-nature of the algorithm is provided in this link.

Notice the following constraints : 1. No. of Obstacles(cells) + No. of agents << 1/4 (Total No. of Cells). This is done, to avoid deadlock due to congestion. 2. Agent can cover only one cell at a given timestamp.

Now, we discuss the analysis of the simulation, for different values of N_p (no. of agents) and N_q (no. of blocks of obstacles).
Here, we have shown results for the agent with observable moore distance = 2.

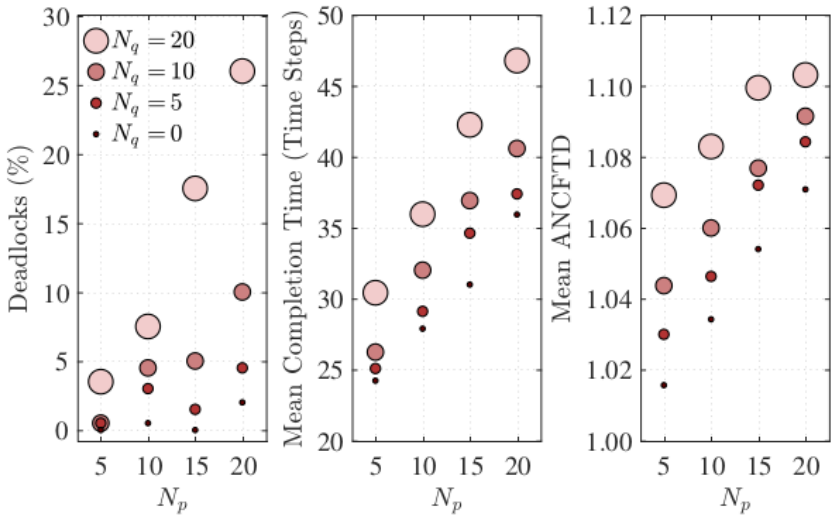


Figure 5: D_observable = 2

Observations : * As Np+Nq increases, Deadlocks increase. However, notice deadlocks are relatively too small for 5-10 drones, which is a realistic scenario for swarm deployment. * Mean Completion time varies linearly with the number of Agents. It also varies exponentially/hyperbolically (not confirmed) with increase in number of obstacles.

BIBLIOGRAPHY

1. Tech Crunch Blog - MIT creates a control algorithm for drone swarms | Natasha Lomas | HYPER-LINK
2. Blockchain-based IoT Platform for Autonomous Drone Operations Management. The Second Workshop on DroneCom in Conjunction with ACM MobiCom 2020, Sep 2020, London, United Kingdom | Samir Dawaliby, Arezki Aberkane, Abbas Bradai | PAPER-LINK
3. Blockchain-assisted secure UAV communication in 6G environment: Architecture, opportunities, and challenges | Rajesh Gupta, Anuja Nair, Sudeep Tanwar, Neeraj Kumar | PAPER-LINK
4. Cellular Automata based Decentralized Cooperative Collision Avoidance Control for Multiple Mobile Robots | Erick J. Rodríguez-Seda and Catalina K. Rico | PAPER-LINK
5. Elementary Cellular Automaton and Its Applications in Computer Science | Varul Srivastava | BLOG-LINK
6. Public-vs-Private-vs-Protected Blockchain | Toshendra Kr. Sharma | Article Link