Computing (2022) 104:1589–1613
https://doi.org/10.1007/s00607-022-01064-7

SPECIAL ISSUE ARTICLE

CovaDel: a blockchain-enabled secure and QoS-aware drone delivery framework for COVID-like pandemics

Maninderpal Singh[1] · Gagangeet Singh Aujla[2] · Rasmeet Singh Bali[1] · Ranbir Singh Batth[3] · Amritpal Singh[4] · Sahil Vashisht[5] · Anish Jindal[2]

Received: 30 May 2021 / Accepted: 8 February 2022 / Published online: 14 March 2022
© The Author(s) 2022

Abstract

With increase in the use of contactless deliveries during the times such as COVID-19 pandemic, the emphasis was to minimize the human presence to reduce the spread of virus. In this regard, drones are one promising alternative to be used as delivery agents. However, security and Quality of Service (QoS) are major concerns while making use of drones for deliveries. In order to secure the drone communication system, we propose, CovaDel: a blockchain-based scheme to secure data transactions for the drone delivery use case that works in a phased manner. The proposed scheme make use of decoupled blockchain architecture to overcome the limited resources capabilities of drones to perform blockchain-based computations. Further, to ensure the QoS adherence, we also propose a QoS-aware communication approach that handles collisions and congestion on the basis of firefly algorithm's attractiveness parameter (light intensity) received by the drones. Results obtained using a simulated environment verify the efficacy of the proposed scheme on the basis of gas consumed, transaction time, average network throughput, and delay.

M. Singh, G. S. Aujla, R. S. Bali, S. Vashisht, A. Singh, and A. Jindal. 2020. Blockchain-enabled secure communication for drone delivery: a case study in COVID-like scenarios. In Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond (DroneCom '20) London, UK, 25-30.

✉ Gagangeet Singh Aujla
gagangeet.s.aujla@durham.ac.uk

[1] Chandigarh University, Mohali, India

[2] Durham University, Durham, UK

[3] Lovely Professional University, Phagwara, India

[4] Chitkara University Institute of Engineering and Technology, Chitkara University, Chandigarh, India

[5] Chandigarh Group of Colleges, Jhanjeri, India

Keywords Blockchain · Firefly optimisation · Quality of service · Drone delivery · Security

Mathematics Subject Classification 94A62 · 94A60 · 93A15 · 90B22

1 Introduction

The transportation sector has helped the common life during the pandemic as it brought essential services and goods to the people. With the widespread emphasis of contactless deliveries of goods to the consumers, drones have become one potential solution for providing essential goods and services to the people. Many successful drone-based delivery services and systems are already in work such as, F-Drones, 7-Eleven, Swoop Aero [1], Zipline [2], Matternet, USPS HorseFly to name a few [3]. Drones offer many benefits to act as the promising delivery agents in COVID-like or disaster-hit scenarios, where the reach of humans is limited. These include their optimized designs, good coverage of area, low maintenance cost, and automated control with very less human involvement. In a typical drone delivery use-case, many entities are involved such as buyers, sellers, service providers, drones and their logistic planning. When all these different entities communicate with one another, it is essential to ensure prompt and secure information to the participating entities and deliver products at the right place in order to gain their trust through smooth operations and transactions. However, many security concerns (like, confidentiality, data manipulation, and authentication) and susceptibility of drones to cyber-attacks hinder their widespread rollout for delivering goods and general trust over their use [1]. Therefore, there is an urgent requirement of a streamlined system which can alleviate the security concerns and provide a secure mode of communication and trust between various involved entities.

In this regard, different researchers have tried to address these security concerns (like, confidentiality, data manipulation, event chronology, and authentication) in-part for different scenarios involving drones. For example, [4] proposed a framework to secure the Internet of Drone (IoD) ecosystem by protecting the drones by using multilevel and multi domain strategies. However, this often involves the use of heavy cryptographic computations which should be avoided in drone-based environment to save resources as well as provide speedy operations [5]. Although many research works have focused on the drones-based delivery services, there is still a major research gap concerning security in such systems for critical scenarios. One of the ways to address the security concerns is to use the blockchain technique which is based on a distributed ledger technology. The premise behind the use of blockchain is that the *blocks* in the blockchain are indestructible. It enables addition of data in the form of transactions and periodically update the transactions from various participating entities into blocks [6]. Overall, a block can be defined as the container of information in the form of chronological transitions thereby handling the issues like confidentiality and event chronology. In blockchain, every block has a header part and a trailer part. The header constitutes necessary control and administrative information required for ensuring consistency and verifiability of recorded transactions [7]. The trailer has actual transactions in it. To weave the blocks together, hashing operation is carried out to compute a fixed length output for any given input. The advantage of hashing is that for the same input, the output will be same, and even a minimal change in the input leads to a different output to a large extent.

Even though blockchain is an effective solution to deal with the security related issues, the computations involved in blockchain-based solutions can be computationally expensive for the (computation limited) drones. Blockchain has been found as a suitable solution for the drone delivery or other drone operations as pointed by many studies [8, 9], but very little exploration has been done for the adaptability of blockchain in the context of disaster or pandemic like situations. One of the key reasons for this little exploration is related to the resource constraints associated with the drones. If the complete blockchain including data (trailer) and header is stored on drone, the storage requirements of drones become very high over whelming the drone resources. The structure of conventional blockchain is appropriate for all entities, however, it has certain limitations when adopted in the drones scenario. Motivated from the work in [10], a derivative of blockchain can be used in the drone delivery scenario. In the derived blockchain, the data and header parts can be decoupled from the blockchain to keep the storage requirement on drones to the minimum. The blockchain will contain only the header part of different blocks. Moreover, each drone can have its own block which is appendable in nature. This way every time new data is added to the block of the drone or existing is updated the resulting hash of the block data changes. The hash of the data of drones is embedded into the blockchain of headers. A complete blockchain and derived blockchain can be maintained with the service provider (or drone station) including blocks for all drones but the drones store only their own block. Therefore, in our proposed approach - named as CovaDel, we make use of decoupled blockchain architecture to reduce the computational burden on drones.

When we have large number of resources, optimal utilization becomes essential to have the desired output of the system. So, if a drone-based delivery system is to be scaled commercially it must be designed keeping the optimal resource allocation and utilization in mind. Hence, to use drones for the purpose of delivery a optimized drone scheduler is needed [11, 12]. Specifically, in the case of drones, there are several attributes (status of physical resources, energy levels, and operational capabilities of hardware resources, etc) that must be taken into account before scheduling it for a mission or delivery. For example, if a drone is scheduled for delivery and during its flight any significant incident or delay occurs, then it may have to be divert its route or delivery plans. In such a case, it must have significant resources (like energy level) to sustain its flight considering the diversion or delay. If this is not considered by the scheduler then it may end up in delivery failure or any other untoward accident. Thus, effective scheduling mechanisms work as an enabler for ensuring that the quality of service (QoS) and quality of experience (QoE) parameters are met to make the system adaptable and worthy of deployment. In CovaDel, the scheduler considers all the above highlighted constraints before selecting a drone for delivery. To aid this process, a drone indexing approach is adopted that categorizes and index the drones based on its computational capabilities and physical resource status.

Moreover, the foundation of drone-based system relies primarily on the prompt communication between the involved entities and in this regard, some of the related communication concerns like, network congestion and collisions can have damaging impact on the overall QoS and QoE at the large. The density of the drones in a given

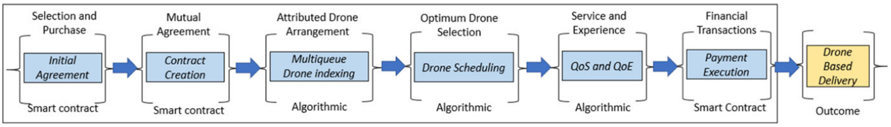

Fig. 1 Schematic diagram for the research approach

area have big impact on the fluctuations of the bandwidth availability. This also impacts the on board resource utilization attributed to handling the network congestion [13]. Further, as the drones use wireless networks that uses shared channels with other coexisting networks like terrestrial and cellular networks resulting into interference, hence leading to collision of data packets. The corrupted data needs to be re-transmitted which is not always feasible due to the real time nature of the environment in which drones operate. Thus, because of this the drone delivery mechanisms should be robust enough to deal with the communication-based issues such as collisions and network congestion. In order to address this, CovaDel proposes a QoS-aware communication amongst the drones on the basis of firefly algorithm's attractiveness parameter. Firefly algorithm is an effective optimization technique that can be applied to swarms of drones [14]. The key parameters for the firefly are attractiveness, randomization and absorption that is influenced from tropical firefly behaviour.

1.1 Research approach and contributions

CovaDel is focused on the delivery of essential goods (or medicines) using drones in pandemic or disaster-like scenarios when the physical moment is restricted (or limited). This article is the extension of our previous work [15] where we have provided the preliminary results and information on the use of blockchain for COVID-like scenarios. The overview of the research approach adopted is presented in the form of a schematic diagram depicted in Fig. 1.

In CovaDel, the block structure used for drone-to-drone, drone-to-ground station communication, and data storage differs from the traditional blockchain approach (Sect. 4). The approach workflow is based on different phases responsible for the functioning of CovaDel. For each phase, a self executing code, i.e., smart contracts, is designed for the interaction between the user and the buyer for placing an order (details given in Sect. 5.1). Furthermore, smart contracts are designed for ensuring the delivery of the items and payment processing between the involved entities (Sects. 5.2 and 5.3) respectively. Drones have different attributes in terms of their flight capabilities, so selecting the right drone for the right delivery is an essential job at hand which is achieved through a multi-queue drone indexing phase (Sect. 5.4). The scheduling of drones is discussed in Sect. 5.5. A good experience is the key for the popularity of any system among masses, hence a QoE enabling mechanism which is possible by ensuring the QoS for the drone communication becomes essential. Thus, a QoS-aware communication approach that handles collisions and congestion situation effectively is proposed in Sect. 6. Some of the significant contributions in this work are listed below.

- A systematic architecture for drone delivery use case has been presented wherein blockchain has been used for secure data transactions. A decoupled blockchain has

been used to overcome the challenge of limited resources capabilities of drones concerning the heavy computational perspective of blockchain process.
- An end-to-end drone delivery mechanism using blockchain and multi-level queuing is designed considering different categories of drones based on their capabilities (such as weight, speed, fitness, battery, etc.).
- A QoS-aware communication approach that handles collisions and congestion is proposed on the basis of light intensity received by the drones.

2 Related work

The existing literature in this field is presented in this section driven in two research directions focusing on (1) research aspects, and (2) drone delivery use cases.

2.1 Research aspects in existing works

In [16], the authors evaluated a drone delivery case study related to healthcare field. The authors have studied various drone service providers, load capacities of various drones, and their suitability to the drone-based delivery services. Work by the authors in [17] includes the study of the environmental impact of using drone-based delivery services in comparison to motorcycle-based delivery of goods. The study clearly indicated the drone-based delivery holds superior to other means of goods delivery in terms of environmental impact. In [18], the authors have undertaken the study to provide an optimised delivery route planning using mobile ground station for the drones. The trajectory of the ground station is computed in a way that the ground station movement is minimum and the drones are capable of optimize their operations to maximize their coverage area. Kim et al. [19] suggested the use of mixed integer linear programming (MILP) model for planning drone activity to increase the count of drone-based parcel delivery that would take place on building rooftops. However, the major drawback of all these studies is that none of these works considered the security and privacy in drones.

In [20], the authors concluded that the any type of attack in drones can be fatal as it can stop its rotors mid air. Another similar work by authors in [21] on generic drone framework was presented along with its security assessment and resilience for security attacks like battery depletion, eavesdropping, jamming, replay and fabrication. In [22], the authors have presented a complete study of security aspects of IoT and related areas along with security mechanism to deal with the security concerns. All of these studies emphasize on the consideration of security flaws in drones, however, none of the above proposals considered a scenario related to product delivery or emergency delivery using drones. Since the delivery of commercial or non-commercial goods involve personal data or sensitive information related to the customers, it becomes essential to secure these transactions and maintain the integrity of the data. In view of these facts, the researchers in [1] have presented a framework for drone-based delivery of goods. The authors made use of cryptography to ensure confidentiality, authentication and non-repudiation. However, it seems that blockchain is better suited technology that can handle the various challenges concerning drone delivery. On one hand, blockchain can handle the transactions in drone delivery in a secure and tamper proof manner, while, on the other hand, we can utilize the smart contracts to verify and validate the authenticity of the transactions and flag out any potential violations.

Keeping in view of these facts, the authors in [23] presented blockchain, as a distributed ledger, which uses cryptographic methods to secure the shared data. It can also be used to ensure the accuracy of the data stored, as well as to improve the reliability and accountability of unmanned areal vehicles (UAVs). In another work [8], a framework for drone-based delivery empowered by blockchain was proposed to ensure system security. The proposed models uses practical Byzantine fault tolerance consensus mechanism for ensuring the consistence of the blockchain. The system was analysed against various security attacks. Another mechanism for validating the vendor packages was also presented in this work. In [9], the authors described the use of a blockchain and drone combination for delivery in countering COVID-19 to track, identify, handle, and regulate public health emergencies.

2.2 Drone delivery – industrial/commercial use cases

Drone delivery is a massive technology which has proven itself to be an essential part of futuristic transportation systems. Drones helps to redefine the conventional logistic industry, which results in saving of crucial operational time, resources and the cost. This section provides information about the various drone delivery applications categorized into three domains discussed below.

- *Essentials and medicine* Contactless deliveries are booming during the pandemic time and drones are playing an important role in connecting consumer and the service providers by catering to such demands. Drones have become the first choice for delivering the essential goods such as medicine [24], first aid and sensitization during the lockdown time as well as in disaster-hit areas [25]. Drones even help to deliver the transplanted organs in time bound manner which can save one's life [26].
- *e-commerce* The delivery of parcels and goods with the help drone is popular these days. The logistics companies such as DHL [27], Anavia [28] and Amazon [29], utilizes the drones for automated delivery of their products to the consumers. Some use-cases such as [30] exploits the use of drones in the marine deliveries of goods and containers to-and-from vessel to consumers locations and offshore. It is believed that this process helps to reduce the carbon footprints. This is derived due the huge demand of faster, cheaper and ideally green forms of deliveries.
- *Grocery and food delivery* 7 Eleven utilizes the drones for the delivery of grocery items to the customers [31]. To avail shop-to-home delivery, the customers register themselves on the site by providing the their home GPS coordinates for accurate delivery. Dominos, the famous pizza company, also focus on the use of drones to deliver hot pizzas in time bound delivery to its customers [32]. Moreover, a tacocopter (renamed from drone) to deliver tacos in San Fanscisco is designed by star simpson [33].

Table 1 show different drone delivery scenarios adopted by various organizations.

Table 1 Comparison of drone delivery cases

Case	Category	Delivery purpose	No. of drone	Payload
[24]	Essentials	Medicine	Single	Upto 10 kg
[26]	Essentials	Transplanted organs	Single	NA
[27]	e-commerce	Parcel	Single	Upto 2 kg
[28]	e-commerce	Parcel	Single	Upto 65 Kg
[29]	e-commerce	Goods	Single	NA
[30]	e-commerce	Marine delivery	Single & Multi	NA
[31]	Grocery & Food	Grocery and food	Single	Upto 10 Kg
[32]	Grocery & Food	Pizza	Single	Upto 5 Kg
[33]	Grocery & Food	Taco	Single	NA

3 System model

In this section, the various components of the drone delivery system are discussed. In the system model, blockchain is used to ensure a secure communication environment and smart contracts ensure the trust of various entities (buyer, seller, drone, service provider, etc.). Any change in the system state during the drone delivery process is referenced as a transaction on the blockchain. Figure 2 presents the system model for the drone delivery scenario, various components of which are discussed as follows.

3.1 Seller

In the proposed model, the products are sold by an entity called seller (**S**). It provides the details and cost of the products to a cloud-based service provider (**CS**) for product listing on the online platform. **S** continuously updates the product information to **CS** with respect to its availability in the warehouse (**W**).

3.2 Buyer

A buyer (**B**) can use the **CS** platform to buy any product listed there. **B** can browse products and finally place the order on the **CS** platform. However, before the buyer can actually place the order, a smart contract should be prepared by **S** which must be accepted by **B** to avoid any future conflicts. Once the contract is executed, an invoice ($\mathbf{IV}_{\mathbf{O}_k}$) is generated for the kth order ($\mathbf{O}_k$) placed by jth buyer ($\mathbf{B}_j$) with ith seller ($\mathbf{S}_i$).

3.3 Cloud-based service provider

CS offers an online merchandise service wherein it lists the products sold by (**S**). It is responsible for online product listing and the management of the order received from **B** in the proposed architecture. It provides middleware services to the **B** and **S** in a systematic manner. The key roles of **CS** are enlisted below:

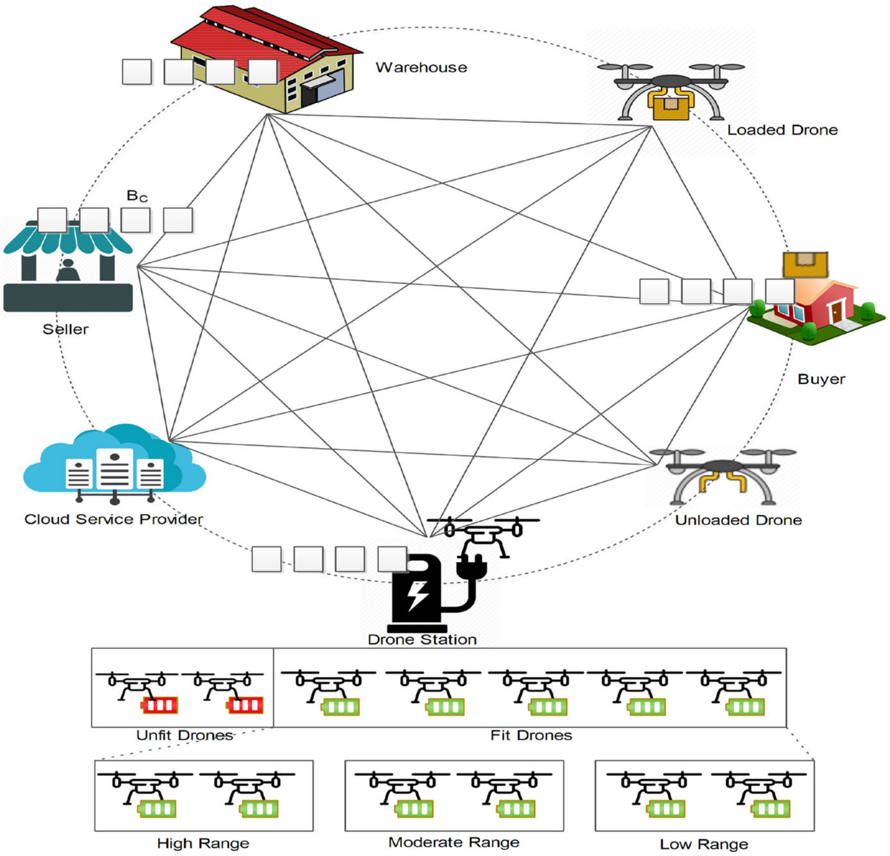

Fig. 2 System model for the drone delivery scenario

- *Product enlisting service* **S** uses the product enlisting service to list their products on the **CS** platform. These products are suggested to **B** on the online platform as per their interests.
- *Browsing service* **B** can use the online platform provided by **CS** to search, browse and order any product of their choice. The j^{th} buyer ($\mathbf{B}_j$) can place the order $\mathbf{O}_k$ with the i^{th} seller $\mathbf{S}_i$.
- *Smart contract* A smart contract (**SC**) is an auto-initialized code which is executed when certain conditions are met. When $\mathbf{B}_j$ wants to buy a product from $\mathbf{S}_i$ through the **CS**, an **SC** is created before $\mathbf{O}_k$ is placed at the **CS**. This **SC** is signed by both $\mathbf{B}_j$ and $\mathbf{S}_i$ to generate a corresponding $\mathbf{O}_k$.
- *Blockchain* The interactions with the blockchain (**BC**) are performed by **CS**. Here, **CS** adds the shipping details $\mathbf{SH}_{\mathbf{O}_k}$ corresponding to $\mathbf{O}_k$ on the **BC**.

3.4 Warehouse

$\mathbf{S}_i$ stores its products at **W** and continuously updates the availability status of the product to the **CS**. When $\mathbf{IV}_{\mathbf{O}_k}$ is generated by $\mathbf{S}_i$ for $\mathbf{O}_k$, the process to prepare the product for shipping (delivery) is initiated at **W**.

3.5 Drone station

The drones are responsible for the product delivery from **W** to the address provided by **B**. The drones are located in the facility called drone station (**DS**). The flight of i^{th} drone ($\mathbf{D}_i$) begins from **DS** leading towards the shipment pickup from the **W** and then delivering it to $\mathbf{B}_j$ on the address provided in $\mathbf{SH}_{\mathbf{O}_k}$. After successful shipping of the product(s), $\mathbf{D}_i$ moves back to **DS**. At **DS**, the drones might be categorised into following categories based on their properties and states.

- *Unfit drones*($\mathbf{D}_i^U$) When $\mathbf{D}_i$ returns from a delivery, the status of the physical resources is monitored by the **DS**. This status includes battery levels, operations of hardware components, or other physical resources. If the battery status ($\mathbf{BT}_i$) of $\mathbf{D}_i$ is below a threshold required for the shortest possible flight in the future, then $\mathbf{D}_i$ has to charge its battery to be eligible to join the fit list.
- *Fit drones* ($\mathbf{D}_i^F$) $\mathbf{D}_i$ that has all its operational resources in functional order and $\mathbf{BT}_i$ above a threshold value ($\mathbf{BT}_{TH}$), then it is considered to be fit. Algorithm 2 is then used to select a drone for delivery from the fit list.

4 Blockchain architecture

In proposed technique similar to [15], blockchain is used by the participating entities for the storage of the information. To ensure the participation of trusted entities only, private blockchain is used. In the private blockchain, only permitted participants can join in the network. Moreover the allowed participants are only allowed to perform the authorized operations. The structural component of **BC** where the data is actually stored is known as block [34]. Each block is composed of two parts known as header and trailer. The header has the data management and link information, whereas the trailer stores the data.

4.1 Header part

This part has all the information related to the control and management of the blocks. More details on the components of header are as below:

- *Sequence number* The blocks are created as the time passes by. This means that the block ($\mathbf{BL}_i$) is created after $\mathbf{BL}_{i-1}$ and before $\mathbf{BL}_{i+1}$. The block gets stored in chronological order of their creation with the help of control information field known as sequence number ($\mathbf{SQ}_i$).
- *Timestamp* Blockchain has a problem, known as forking, where the consistent state of blockchain with different entities is different because the main fork is split

into different forks. To identify the occurrence of the events, a timestamp ($\mathbf{TS}_i$) of block generation is embedded inside the header.

- *Previous block hash* The biggest advantage that blockchain brings in is the distributed data storage along with the property that transactions once committed are immutable. This is achieved through embedding the hash of the previous block into the new block being made, i.e., in ith block, the hash of the $i-1$th block is stored as previous block hash ($\mathbf{H}_{\mathbf{BL}_{i-1}}$).
- *Current block hash* Based on the difficulty level, the computed current block hash has certain number of leading bits as zero in the target hash value.
- *Nonce* The hash of the current block can only be achieved if we set a value inside the header that forces the output hash to be of specific nature. It is achieved using pseudo random numbers called nonce ($\mathbf{N}$).

4.2 Trailer part

The area where the data is stored inside the block is known as trailer part. The trailer in the proposed blockchain model comprises the following:

- *Smart contract* When **B** places an order, it is converted into **SC** which is signed by **S** and **B** as an agreement.
- *Shipping* If the shipping information is manipulated by an attacker, the product may be delivered to the wrong address. Hence, to keep the shipping information immutable, it is stored on the blockchain.

The structure of **BC** is appropriate for all entities except the drone due to its resource limitations. If the complete blockchain is stored on a drone, it may lead to the over whelming of drone resources. Thus, a decoupled blockchain ($\mathbf{BC^D}$) is used for drones and **GS** communications. The data and header parts are decoupled from the blockchain to keep the storage requirement on drones to the minimum. The $\mathbf{BC^D}$ comprises only the header part of different blocks. Moreover, each drone has its own block which is appendable in nature. This way every time new data is added to the block of the drone, the resulting hash of the data block changes. The hash of the data of drones is embedded into the blockchain of headers, i.e., $\mathbf{BC^D}$. The blockchains are maintained with the **CS**, while the drones store only their own block excluding the data of other drones within the $\mathbf{BC^D}$. The block structure used in $\mathbf{BC^D}$ for drone-to-drone and drone-to-ground station communication, and data storage differs from the **BC** in the following aspects:

- *Header* In the decoupled blockchain, the header contains the control information for $\mathbf{D}_i$. This includes the hash of previous block, drone attributes such as payload capacity ($\mathbf{D}^{max}_{pld}$), ratted battery capacity ($\mathbf{D}^{rtd}_{bat}$), hash of drones registration identity ($\mathbf{H}_{\mathbf{D}_{id}}$), timestamp of block header creation ($\mathbf{T}^{hdr}$), drones public key ($\mathbf{D}^{pb}_i$) and private key ($\mathbf{D}^{pr}_i$), and the merkle root hash (**MRH**) of the drone data ($\mathbf{D}^{MRH}_i$).
- *Data* The data is not stored on the blockchain and only the **MRH** of the data is stored inside the header to ensure integrity. However, the **DS** and **CS** maintain the complete copy of data and header for drone interactions, while the drones store the current mission-related data ($\mathbf{D}^{cur}_{i(d)}$) only.

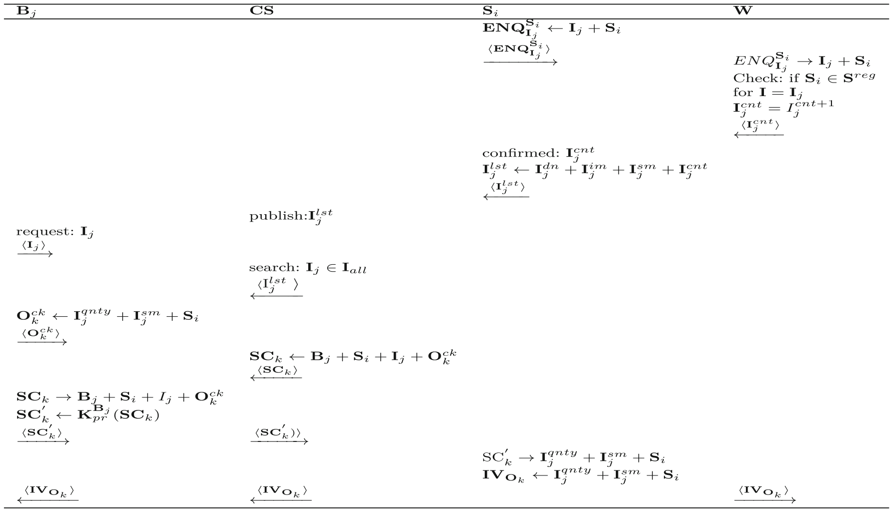
$\mathbf{B}_j$
$\mathbf{CS}$
$\mathbf{S}_i$
$\mathbf{W}$
$\mathbf{ENQ}^{\mathbf{S}_i}_{\mathbf{I}_j} \leftarrow \mathbf{I}_j + \mathbf{S}_i$
$\langle \mathbf{ENQ}^{\mathbf{S}_i}_{\mathbf{I}_j} \rangle$
$ENQ^{\mathbf{S}_i}_{\mathbf{I}_j} \rightarrow \mathbf{I}_j + \mathbf{S}_i$
Check: if $\mathbf{S}_i \in \mathbf{S}^{reg}$
for $\mathbf{I} = \mathbf{I}_j$
$\mathbf{I}^{cnt}_j = I^{cnt+1}_j$
$\langle \mathbf{I}^{cnt}_j \rangle$
confirmed: $\mathbf{I}^{cnt}_j$
$\mathbf{I}^{lst}_j \leftarrow \mathbf{I}^{dn}_j + \mathbf{I}^{im}_j + \mathbf{I}^{sm}_j + \mathbf{I}^{cnt}_j$
$\langle \mathbf{I}^{lst}_j \rangle$
publish:$\mathbf{I}^{lst}_j$
request: $\mathbf{I}_j$
$\langle \mathbf{I}_j \rangle$
search: $\mathbf{I}_j \in \mathbf{I}_{all}$
$\langle \mathbf{I}^{lst}_j \rangle$
$\mathbf{O}^{ck}_k \leftarrow \mathbf{I}^{qnty}_j + \mathbf{I}^{sm}_j + \mathbf{S}_i$
$\langle \mathbf{O}^{ck}_k \rangle$
$\mathbf{SC}_k \leftarrow \mathbf{B}_j + \mathbf{S}_i + \mathbf{I}_j + \mathbf{O}^{ck}_k$
$\langle \mathbf{SC}_k \rangle$
$\mathbf{SC}_k \rightarrow \mathbf{B}_j + \mathbf{S}_i + I_j + \mathbf{O}^{ck}_k$
$\mathbf{SC}'_k \leftarrow \mathbf{K}^{\mathbf{B}_j}_{pr}(\mathbf{SC}_k)$
$\langle \mathbf{SC}'_k \rangle$
$\langle \mathbf{SC}'_k \rangle\rangle$
$\mathrm{SC}'_k \rightarrow \mathbf{I}^{qnty}_j + \mathbf{I}^{sm}_j + \mathbf{S}_i$
$\mathbf{IV}_{\mathbf{O}_k} \leftarrow \mathbf{I}^{qnty}_j + \mathbf{I}^{sm}_j + \mathbf{S}_i$
$\langle \mathbf{IV}_{\mathbf{O}_k} \rangle$
$\langle \mathbf{IV}_{\mathbf{O}_k} \rangle$
$\langle \mathbf{IV}_{\mathbf{O}_k} \rangle$

Fig. 3 Initial agreement and smart contract creation

5 CovaDel: the proposed drone delivery framework

The proposed framework for drone-based delivery scenario consists of several phases that work in tandem to achieve the overall objective. These phases are discussed in the following sections.

5.1 Initial agreement and smart contract creation phase

The first phase deals with establishing an agreement between the **B** and **S**. This agreement is translated in to a smart contract that is deployed on the blockchain. The actions performed in this phase are depicted in Fig. 3.

- $\mathbf{S}_i$ sends an enquiry ($\mathbf{ENQ}^{\mathbf{S}_i}_{\mathbf{I}_j}$) to **W** for the available quantity of the item ($\mathbf{I}_j$).
- When **W** receives $\mathbf{ENQ}^{\mathbf{S}_i}_{\mathbf{I}_j}$, it authenticates $\mathbf{S}_i$ from the list of registered sellers ($\mathbf{S}^{reg}$) and the following condition should be true.

$$\mathbf{S}_i \in \mathbf{S}^{reg} \tag{1}$$

  After this process, **W** check the $\mathbf{I}_j$ in the stock and reverts $\mathbf{S}_i$ with available stock inventory count ($\mathbf{I}^{cnt}_j$) for the enquired item.
- Based on $\mathbf{I}^{cnt}_j$, the $\mathbf{S}_i$ lists the items on **CS** for sales. The jth listed item ($\mathbf{I}^{lst}_j$) details on **CS** comprises of item description ($\mathbf{I}^{dn}_j$), images ($\mathbf{I}^{im}_j$), available shipping methods ($\mathbf{I}^{sm}_j$) and the **W** confirmed stock count ($\mathbf{I}^{cnt}_j$). The item listing ($\mathbf{I}^{lst}$) on the **CS** includes the following details.

$$\mathbf{I}^{lst}_j \rightarrow \{\mathbf{I}^{dn}_j + \mathbf{I}^{im}_j + \mathbf{I}^{sm}_j + \mathbf{I}^{cnt}_j\} \tag{2}$$

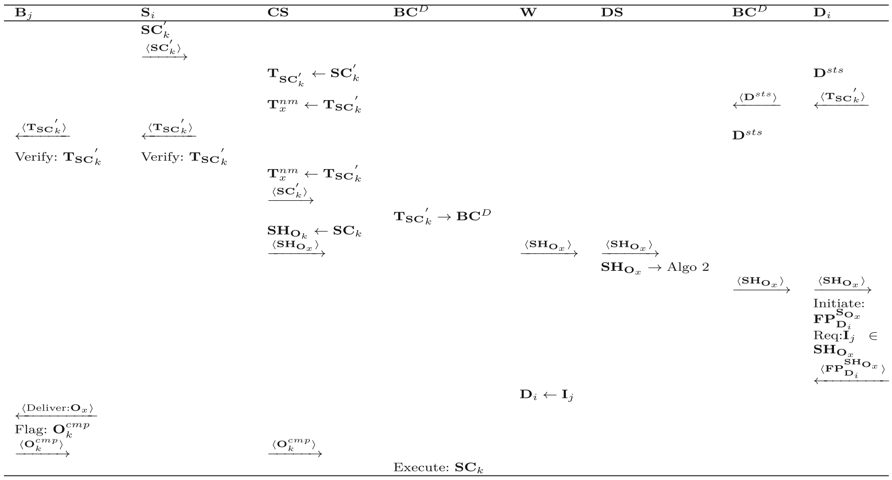
$\mathbf{B}_j$
$\mathbf{S}_i$
$\mathbf{CS}$
$\mathbf{BC}^D$
$\mathbf{W}$
$\mathbf{DS}$
$\mathbf{BC}^D$
$\mathbf{D}_i$
$\mathbf{SC}'_k$
$\langle \mathbf{SC}'_k \rangle$
$\mathbf{T}_{\mathbf{SC}'_k} \leftarrow \mathbf{SC}'_k$
$\mathbf{D}^{sts}$
$\langle \mathbf{D}^{sts} \rangle$
$\langle \mathbf{T}_{\mathbf{SC}'_k} \rangle$
$\mathbf{T}_x^{nm} \leftarrow \mathbf{T}_{\mathbf{SC}'_k}$
$\langle \mathbf{T}_{\mathbf{SC}'_k} \rangle$
$\langle \mathbf{T}_{\mathbf{SC}'_k} \rangle$
$\mathbf{D}^{sts}$
Verify: $\mathbf{T}_{\mathbf{SC}'_k}$
Verify: $\mathbf{T}_{\mathbf{SC}'_k}$
$\mathbf{T}_x^{nm} \leftarrow \mathbf{T}_{\mathbf{SC}'_k}$
$\langle \mathbf{SC}'_k \rangle$
$\mathbf{T}_{\mathbf{SC}'_k} \rightarrow \mathbf{BC}^D$
$\mathbf{SH}_{\mathbf{O}_k} \leftarrow \mathbf{SC}_k$
$\langle \mathbf{SH}_{\mathbf{O}_x} \rangle$
$\langle \mathbf{SH}_{\mathbf{O}_x} \rangle$
$\langle \mathbf{SH}_{\mathbf{O}_x} \rangle$
$\mathbf{SH}_{\mathbf{O}_x} \rightarrow$ Algo 2
$\langle \mathbf{SH}_{\mathbf{O}_x} \rangle$
$\langle \mathbf{SH}_{\mathbf{O}_x} \rangle$
Initiate: $\mathbf{FP}_{\mathbf{D}_i}^{\mathbf{SO}_x}$
Req:$\mathbf{I}_j \in \mathbf{SH}_{\mathbf{O}_x}$
$\langle \mathbf{FP}_{\mathbf{D}_i}^{\mathbf{SH}_{\mathbf{O}_x}} \rangle$
$\mathbf{D}_i \leftarrow \mathbf{I}_j$
$\langle$Deliver:$\mathbf{O}_x \rangle$
Flag: $\mathbf{O}_k^{cmp}$
$\langle \mathbf{O}_k^{cmp} \rangle$
$\langle \mathbf{O}_k^{cmp} \rangle$
Execute: $\mathbf{SC}_k$

Fig. 4 Delivery phase

- $\mathbf{B}_j$ search for required $\mathbf{I}_j$ at **CS** platform and place the request. On the receipt of request, **CS** checks $\mathbf{I}_j$ in the available items and revert $\mathbf{B}_j$ with confirmation.
- Now, $\mathbf{B}_j$ proceeds with the order ($\mathbf{O}_k^{ck}$) wherein it mentions the quantity ($\mathbf{I}_j^{qnty}$), shipping method ($\mathbf{I}_j^{sm}$) and preferred $\mathbf{S}_i$. The generated $\mathbf{O}_k^{ck}$ is confirmed for further processing with **CS**.
- Once **CS** receives $\mathbf{O}_k^{ck}$, it prepares $\mathbf{SC}_k$ (for kth order) comprising information about $\mathbf{B}_j$, $\mathbf{S}_i$, $\mathbf{I}_j$ and $\mathbf{O}_k^{ck}$ and then send it to $\mathbf{B}_j$ for final approval.
- Now, $\mathbf{B}_j$ signs the $\mathbf{SC}_k$ using its private key ($\mathbf{K}_{pr}^{\mathbf{B}_j}$) and send to **CS** as follows:

$$\mathbf{SC}'_k \leftarrow \mathbf{K}_{pr}^{\mathbf{B}_j}(\mathbf{SC}_k) \tag{3}$$

- **CS** send the $\mathbf{SC}'_k$ to $\mathbf{S}_i$ for preparation of invoice($\mathbf{IV}_{\mathbf{O}_k}$). Finally, the $\mathbf{IV}_{\mathbf{O}_k}$ is sent to the $\mathbf{B}_j$ through **CS**.

5.2 Delivery phase

The second phase deals with the delivery of the product from **W** to the buyers address. The actions of the involved entities are as shown in Fig. 4.

- **CS** adds $\mathbf{SC}_k$ into the **BC**. Initially, $\mathbf{SC}_k$ is added to the pool of un-mined transactions ($\mathbf{T}_x^{nm}$) in the form of $\mathbf{T}_{\mathbf{SC}_k}$.

$$\mathbf{T}_x^{nm} \leftarrow \mathbf{T}_{\mathbf{SC}_{k-1}} + \mathbf{T}_{\mathbf{SC}_k} + \mathbf{T}_{\mathbf{SC}_{k+1}} + \cdots \mathbf{T}_{\mathbf{SC}_{k+n}} \tag{4}$$

- **B** and **S** act as miners on the **BC**. The transaction between $\mathbf{B}_j$ and $\mathbf{S}_i$, i.e., $\mathbf{T}_{\mathbf{SC}_k}$ is verified and further sent for mining into the block of **BC**. If either of the involved entities can't verify $\mathbf{T}_{\mathbf{SC}_k}$, it is denied and aborted.

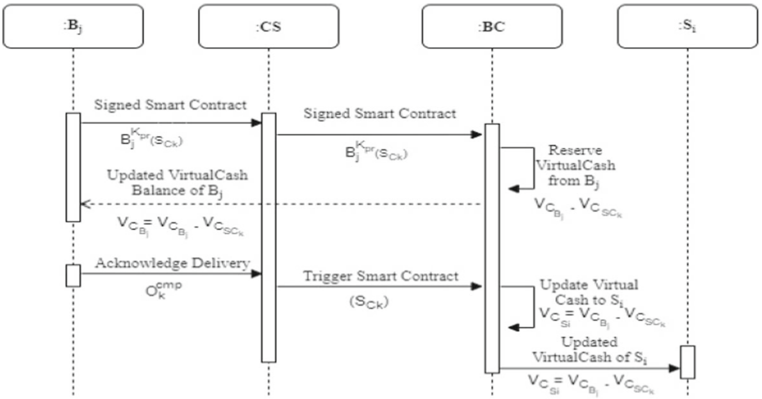
:B_j
:CS
:BC
:S_i
Signed Smart Contract
$B_j^{K_{pr}}(S_{Ck})$
Signed Smart Contract
$B_j^{K_{pr}}(S_{Ck})$
Reserve VirtualCash from B_j
$V_{C_{B_j}} - V_{C_{SC_k}}$
Updated VirtualCash Balance of B_j
$V_{C_{B_j}} = V_{C_{B_j}} - V_{C_{SC_k}}$
Acknowledge Delivery
O_k^{cmp}
Trigger Smart Contract
(S_{Ck})
Update Virtual Cash to S_i
$V_{C_{Si}} = V_{C_{B_j}} - V_{C_{SC_k}}$
Updated VirtualCash of S_i
$V_{C_{Si}} = V_{C_{B_j}} - V_{C_{SC_k}}$

Fig. 5 Payment phase

- Once the $\mathbf{T}_{\mathbf{SC}_k}$ becomes part of **BC**, the $\mathbf{SH}_{\mathbf{O}_k}$ is extracted from it and provided to **DS**. Now, **DS** selects and schedules $\mathbf{D}_i$ using Algorithm 2.
- **DS** receives an up to date information about drone status through $\mathbf{BC}^D$. The $\mathbf{SH}_{\mathbf{O}_k}$ is added to the block of $\mathbf{D}_i$ in $\mathbf{BC}^D$.
- $\mathbf{D}_i$ reads the information from its block on $\mathbf{BC}^D$ and initiates a flight plan (**FP**) for the execution of $\mathbf{O}_k$ ($\mathbf{FP}_{\mathbf{D}_i}^{\mathbf{SH}_{\mathbf{O}_k}}$).
- $\mathbf{D}_i$ collects the $\mathbf{I}_j$ as per $\mathbf{O}_k$ from **W** and delivers it as per $\mathbf{SH}_{O_k}$ and return to the **DS** where **DS**. Now, the state of $\mathbf{D}_i$ is updated on the $\mathbf{BC}^D$ after delivery.
- $\mathbf{D}_i$ adds its status ($\mathbf{D}^{sts}$) comprising of drone payload capacity ($\mathbf{D}_{pld}^{max}$), rated battery capacity($\mathbf{D}_{bat}^{rtd}$), hashed identity($\mathbf{D}_{id}^{hash}$) and the public private key pair ($\mathbf{K}_{pr}^{\mathbf{D}_i}$, $\mathbf{K}_{pb}^{\mathbf{D}_i}$) on $\mathbf{BC}^D$. This information acts as input to the Algorithm 2 using multi-level queuing technique to schedule $\mathbf{D}_i$ for a delivery.
- When $\mathbf{B}_j$ receive $\mathbf{I}_j$ as per $\mathbf{O}_k$, a flag ($\mathbf{O}_k^{cmp}$) is raised confirming the receipt of $\mathbf{I}_j$. The $\mathbf{SC}_k^{'}$ is executed by $\mathbf{BC}^D$ and the status is notified to $\mathbf{S}_i$.

5.3 Payment phase

The third phase is related to the payment to the $\mathbf{S}_i$ from the $\mathbf{B}_j$'s reserve. The **SC** is auto executed once the order is delivered to $\mathbf{B}_j$ and the corresponding amount is transferred to the $\mathbf{S}_i$'s cash wallet. The interaction between various entities is shown in Fig. 5.

- $\mathbf{B}_j$ agrees to the **SC** made by the **CS** by signing it, i.e., $\mathbf{SC}_k^{'}$ and the same is forwarded to the **BC** through **CS**.
- On receiving $\mathbf{SC}_k^{'}$, the **BC** reserves the amount of virtual cash ($\mathbf{VC}_{\mathbf{B}_j}$) corresponding to the $\mathbf{SC}_k$ from the available virtual cash ($\mathbf{VC}_{\mathbf{B}_j}$) of $\mathbf{B}_j$.
- The updated $\mathbf{VC}_{\mathbf{B}_j}$ is returned to $\mathbf{B}_j$ to eliminate the double spending problem, as $\mathbf{B}_j$ is not able to spend the reserved amount until $\mathbf{O}_k$ either succeeds.
- Once $\mathbf{I}_j$ is delivered by $\mathbf{D}_i$ to $\mathbf{B}_j$, it initiates the order completion flag ($\mathbf{O}_k^{cmp}$).

- On receiving $\mathbf{O}_k^{cmp}$, the $\mathbf{SC}_k$ is triggered on the **BC** that adds $\mathbf{VC}_{\mathbf{SC}_k}$ to the Virtual cash($\mathbf{VC}_{\mathbf{S}_i}$) of $\mathbf{S}_i$.
- The updated $\mathbf{VC}_{\mathbf{S}_i}$ is returned to the $\mathbf{S}_i$ leading to the completion of payment.

$$\mathbf{VC}_{\mathbf{S}_i} = \mathbf{VC}_{\mathbf{S}_i} + \mathbf{VC}_{\mathbf{SC}_k} \tag{5}$$

5.4 Multi-queue drone indexing phase

Initially, the drones are categorized into fit and Un-fit drone queue based on their current battery status. Further, the available drones are segregated into the heavy range queue ($\mathbf{Q}^{HR}$), moderate range queue ($\mathbf{Q}^{MR}$), and low range queue ($\mathbf{Q}^{LR}$) using the proposed multi-queue drone indexing algorithm (Algorithm 1).

Algorithm 1 Multi-queue drone indexing algorithm

```
INPUT: Drones → $\mathbf{D}_i$
OUTPUT: Categorized queues:- $\mathbf{Q}^{HR}$, $\mathbf{Q}^{MR}$, $\mathbf{Q}^{LR}$
INITIALIZE POOL: $\mathbf{D}^f_{pool}$, count = 0
for $i = 1, i \leq n, i++$ do
    if $\mathbf{D}_i == fit$ then
        $count++$
        ADD: $\mathbf{D}_i \rightarrow \mathbf{D}^f_{pool}$
    end if
end for
for $i = 1, i \leq count, i++$ do
    POP: $\mathbf{D}_i \leftarrow \mathbf{D}^f_{pool}$
    if $\mathbf{D}_i^{\varkappa} == \tau^{max} \&\& \mathbf{D}_i^{\mathfrak{R}} == \mathfrak{R}^{max}$ then
        PUSH: $\mathbf{D}_i \rightarrow \mathbf{Q}^{HR}$
    else if $\mathbf{D}_i^{\tau} == \tau^{mod} \&\& \mathbf{D}_i^{\mathfrak{R}} == \mathfrak{R}^{mod}$ then
        PUSH: $\mathbf{D}_i \rightarrow \mathbf{Q}^{MR}$
    else
        PUSH: $\mathbf{D}_i \rightarrow \mathbf{Q}^{LR}$
    end if
end for
```

Various parameters are considered for categorization of the drones like, towing capacity ($\mathbf{D}_i^{\tau}$), battery status ($\mathbf{D}_i^{\mathfrak{B}}$), and maximum coverage range ($\mathbf{D}_i^{\mathfrak{R}}$). The working of this approach is discussed in the below mentioned steps:

- Initially, all the fully charged (fit) drones are pushed into the pool ($\mathbf{D}^f_{pool}$) starting with an initial count = 0.
- Further, the drones are selected for segregation from the $\mathbf{D}^f_{pool}$.
- If $\mathbf{D}_i^{\tau}$ towing capacity of the drone is maximum (τ^{max}), $\mathbf{D}_i^{\mathfrak{R}}$ range is maximum ($\mathfrak{R}^{max}$), then push the $\mathbf{D}_i$ to the $\mathbf{Q}^{HR}$ queue.
- If $\mathbf{D}_i^{\tau}$ towing capacity of the drone is moderate (τ^{mod}), $\mathbf{D}_i^{\mathfrak{R}}$ range is narrow ($\mathfrak{R}^{mod}$), then index the $\mathbf{D}_i$ to the $\mathbf{Q}^{MR}$ queue.
- Otherwise, push the $\mathbf{D}_i$ to the $\mathbf{Q}^{LR}$ queue.

In the algorithm, τ^{max}, and τ^{mod} are variable and can be fixed based on the drone technology. Similarly, $\mathfrak{R}^{max}$, and $\mathfrak{R}^{mod}$ are also variable and can be fixed based on a specific drone model. The proposed approach helps to select a suitable drone for the delivery. The queue manager supervise all the queues and update the status of all the loaded and unloaded drones in the defined queues. Further, the algorithmic complexity is tight upper bound defined as $\theta(n)$.

5.5 Drone scheduling phase

GS is assigned with the task of optimal selection of drones for shipping the product ($\mathbf{O}_k$) to the delivery address. The **CS** provides the list of the requested products $\mathbf{S}_{O_k}$ to the **GS** for further processing. The number of orders ($\mathbf{O}_k$) initiated in a short interval of time are combined into one optimised flight by the proposed scheduling algorithm. The detailed working of the model is discussed below:

- Scheduling algorithm accepts the input as $\mathbf{S}_{O_k}$ from the **CS** comprising of $\mathbf{O}_k$'s weight ($\mathbf{O}_k^{wt}$), dimensions of package ($\mathbf{O}_k^{di}$), destination location coordinates ($\mathbf{O}_k^{dest}$), nature of shipment ($\mathbf{O}_k^{type}$) and delivery timing ($\mathbf{O}_k^{dt}$) and **TP** are the total number of products to ship at various locations.

$$\mathbf{S}_{O_k^{\mathbf{TP}}} \leftarrow \mathbf{CS}_{(\mathbf{O_k^{wt}}, \mathbf{O_k^{di}}, \mathbf{O_k^{dest}}, \mathbf{O_k^{type}}, \mathbf{O_k^{dt}})} \tag{6}$$

- Fetch $\mathbf{Q}^{HR}$, $\mathbf{Q}^{MR}$, $\mathbf{Q}^{LR}$ from Algorithm 1.
- The newly initiated $\mathbf{S}_{O_k}$ pushed into a shipping queue (**SQ**) as $\mathbf{S}_{O_k} \rightarrow \mathbf{SQ}$
- After the shipping requests are pushed into **SQ**, the following steps are performed till all the shipping request in **SQ** are served.
  - The **SQ** is mapped with the $\mathbf{Q}^{HR}$, $\mathbf{Q}^{MR}$, $\mathbf{Q}^{LR}$ for selection of a drone.
  - The number of drones (**N**) from the matched queue are further mapped as per $\mathbf{O}_k$ requirements.
  - The best match ($\mathbf{D}_i$) is the one with minimum difference in $\mathbf{D}_i^{cap}$ and $\mathbf{S}_{O_k}$.
  - The selected $\mathbf{D}_i$ is allocated to $\mathbf{O}_k$ for product delivery.
  - The details of the $\mathbf{S}_{O_k}$ is offloaded to $\mathbf{D}_i$ for flight.
  - After successfully delivery of $\mathbf{O}_k$, it is removed from the **SQ** of the **DS**.
- The approach return the mapped $\mathbf{D}_i \rightarrow \mathbf{O}_k$ to the **W**, which further hands over the $\mathbf{O}_k$ to the $\mathbf{D}_i$ for delivery to the assigned location.

The distance of the destination $\mathbf{B}_j$ from the warehouse **W** is computed using the following equation as suggested in [34]:

$$d_{(\mathbf{w} \rightarrow \mathbf{B}_j)} = \left| \frac{d_{\mathbf{W}}}{d} \right| \times d + \left| \frac{d_{\mathbf{B}_j}}{d} \right| \times d + n_{(\mathbf{W} \rightarrow \mathbf{B})} \times d \tag{7}$$

The algorithmic complexity of this algorithm is calculated as $O(TP + N)$.

Algorithm 2 Drone Scheduling Algorithm

```
INPUT: $\mathbf{S}_{O_k}, \mathbf{Q}^{HR}, \mathbf{Q}^{MR}, \mathbf{Q}^{LR}$    ▷ Refer Algorithm 1
OUTPUT: $\mathbf{D}_i \rightarrow \mathbf{O}_k$
SHIPPING QUEUE: $\mathbf{SQ} \leftarrow \mathbf{S}_{O_k}$
for $k = 1, k \leq \mathfrak{T}\mathfrak{P}, k++$ do
    if $\mathbf{S}_{O_k} \varepsilon \mathbf{Q}^{HR}$ then
        SHORTLISTED DRONE POOL: $\mathbf{D}_{sl}^{f} \leftarrow \mathbf{Q}^{HR}$
    else if $\mathbf{S}_{O_k} \varepsilon \mathbf{Q}^{MR}$ then
        SHORTLISTED DRONE POOL: $\mathbf{D}_{sl}^{f} \leftarrow \mathbf{Q}^{MR}$
    else
        SHORTLISTED DRONE POOL: $\mathbf{D}_{sl}^{f} \leftarrow \mathbf{Q}^{LR}$
    end if
end for
DRONES IN $\mathbf{D}_{sl}^{f}$ : $\mathbf{N} \leftarrow count(\mathbf{D}_{sl}^{f})$
for $i = 1, i \leq \mathbf{N}, i++$ do
    FIND: min $\mathbf{D}_i \leftarrow \mathbf{D}_i^{cap} - \mathbf{S}_{O_k}$
end for
MAPPING: $\mathbf{D}_i \rightarrow \mathbf{O}_k$
OFFLOAD: $\mathbf{S}_{O_k} \rightarrow \mathbf{D}_i$
REMOVE: $\mathbf{O}_k \notin \mathbf{SQ}$
RETURN $\leftarrow \mathbf{D}_i$
```

6 QoS-aware communication approach

CovaDel is strictly dependent on timely data/information from the participating entities. At a given time, multiple drones may be out for delivery (or other missions) so they can communicate among themselves and to the **GS** to pass the vital information (like, drone density or critical incident in the region) that can aid the drone-based delivery process. The density of the drones flying in a given region can lead towards the bandwidth fluctuations thereby limiting availability of the communication resources. This can further end up in network collisions and thereby network congestion. Now, the collisions and congestion in the network can eventually effect the QoS metrics and therefore degrade the QoE for the drone delivery scenario. The proposed QoS-aware communication approach helps to establish a connection between any two drones for sharing prompt information while adhering to the QoS metrics (end-to-end delay and network throughput). The aims of this approach are concerned with the collision avoidance and congestion control in order to to provide adequate QoS for the underlying communications related to drone delivery process. This approach adopts accurate positioning beacons for collision avoidance among drones and then apply congestion-avoidance policy to coordinate the communication.

This approach uses the properties of fire-fly optimization algorithm [35] and proposes an efficient mechanism based on light-intensity formation (an attractiveness parameter) to enhance QoS in drone communications. Firefly algorithm considers the concept of flashing light (produced through the process of bio-luminescence) generated by fire flies and correlate it to signaling systems. The fundamental functions related to the fire fly's flashing light, also known as attractiveness functions include, (a) to attract mating partners, and (b) to attract prey. Additionally, the flash light is also used as a signal for protective warning. The rate, pattern and related time are key aspects of flashing light that form a part of signalling system between fire flies.

The flashing light concept has been idealized and associated with objective functions related to optimization to formulate a firefly algorithm in [35]. This algorithm follows three key rules, (a) all fireflies are unisex and they can attract each other irrespective of their sex (ideally fit in case of drones), (b) attractiveness is correlated to brightness and both decrease with an increase in the distance, and (c) the brightness can vary with respect to the representation of the objective function. Further, this algorithm relies on two key concepts, (a) variation in the light intensity, and (b) formulation of attractiveness. Looking in to these two factors, the attractiveness of a firefly can be decided on the basis of its brightness (higher or lighter light intensity) and thereby linked to an objective function. We have applied this concept for drone communications and defined two objectives related to collision avoidance and congestion control based on the light intensity (brightness) that varies with an increase in the distance between the drones in a given area.

Inspired form the above concept, a communication approach has been designed wherein we have used the time slot mechanism proposed in [14] to form a connection between drones. Based on this, the proposed approach establish collision avoidance strategy through accurate position beacons as suggested in [36] and thereafter use the congestion control policy to realize the communication. The positioning of the beacons is vital and the effect of localization of drones is critical for drone delivery scenario. As drones have high mobility, the anchor points for the localization tend to change and finding appropriate anchor points is crucial. The proposed technique uses the light intensity parameter for optimization of the drone communications, that relies on the effective beacon positioning and hence provides effectiveness in areas with no fixed infrastructure to deploy the congestion avoidance policies. On the other side of the coin, the beacon position impacts the efficiency of the congestion avoidance which is susceptible to fail if the beacons are not chosen effectively.

In this approach, the average-light intensity received by the drones positioned at location L_i and L_j becomes the base of the proposed collision avoidance approach. In this regard, we define the average-light intensity $(\updownarrow^{(r)}_{A,i})$ received by the drones as below [14].

$$\updownarrow^{(r)}_{(i,j)} = \alpha^{(t)}_{i} + \alpha^{(t)}_{(i,0)} e^{-\eta\beta^{2}} \left(L_i - L_j\right) + \gamma \tag{8}$$

where, the attraction value between two drones concerning $irmth$ is denoted by $\alpha^{(t)}_{i}$, initial attraction value is represented by $\alpha^{(t)}_{i,0}$, γ denotes the density of a drones in its neighbourhood, η represents the rate of change with respect to the present route, and inverse of probability of connectivity is represented by β.

$\updownarrow^{(R)}_{A,i}$ is computed for all the drones in the region, where γ act as the varying factor to enforce the varied number of inter-connected drones. Based on location-awareness, this model identifies collision possibilities as follows.

$$\updownarrow^{(r)}_{i,j} \leq \left(\updownarrow^{(r)}_{i,j}\right)_{(THR)} = \min\left(\overline{\updownarrow^{(r)}_{i}}\right) \forall D \tag{9}$$

where, THR represents a threshold and D represents a set of drones.

The work pattern of the original firefly algorithm tends to increase the light intensity (attractiveness or brightness) that doesn't fit with the requirement of the proposed model. The requirement of this approach is concerned with the lower light intensity as higher intensity (brightness) depicts the possibility of a collision. Thus, we used the modified firefly algorithm from [14] that operates in a reverse pattern thereby triggering the collision avoidance mechanism within the defined time cycle in an efficient manner.

Once the collision possibilities are identified, the next task relates to the congestion control in the network. For this purpose, for the incoming traffic, the intensity mechanism (of firefly algorithm) is used as the base to decrease (or increase) the congestion window by using lower (or higher) streaks for any given connection as proposed in [14]. Let us say, the traffic rate between any two drones is ($\mathcal{V}_{i,j}$), then the congestion-control mechanism works for both drones in a simultaneous manner while considering (or managing) their transmissions (τ) and receptions. The congestion-control evaluations can be expressed as below.

$$\mathcal{C}_{i,j}^{(r)} = \mathcal{V}_{i,j}^{(t)} + \mathcal{V}_o e^{-v\theta^2} \Delta C + C_(cp) \tag{10}$$

where, v denotes the velocity of the drone, θ represents the current heading of a drone, ΔC is the possibility of a connection between the drones and ranges between 0 and 1 for any incoming drones, and $C_(cp)$ represents the number of channels for the connected components in average connectivity in a network.

In the above defined congestion-control evaluation, if at an instance there is an increase in the value of $\mathcal{C}_i^{(r)}$, then the network may end up in congestion. Thus, the proposed model need to adjust traffic based on the following condition.

$$\mathcal{C}_i^{(r)} \leq \left(\mathcal{C}_i^{(r)}\right)_{THR} \tag{11}$$

where, THR represents the threshold that is based on the average rate that is sustainable over a defined number of channels.

Based on the above evaluations, the Algorithm 3 represents the workflow of the QoS-aware drone communications. Here, the first step involves satisfying the collision avoidance criteria. The mechanism continuously checks the possibilities of collisions over the defined way-points thereby averting any possible termination in the transmission. The mechanism ensures that a drone must satisfy the defined light-intensity constraints keeping in view of the neighbouring drone. If in a case, these conditions are not satisfied or doesn't hold TRUE, then an instant feedback is triggered so that the timing control can consider it while providing a time slot for next drone communication.

Once the collision-avoidance possibilities are satisfied, the next step includes the fulfilment of congestion control evaluations. The algorithm checks the possible collisions over the way-points in a continuous manner by managing the size of the congestion window. This helps to prevent any unintentional breakage in the transmissions and traffic overloading scenario. Once both the criterion are satisfied, we

Algorithm 3 QoS-aware Communication Algorithm

```
Input: Drone network components and entire system model
Output: Transmission route
Initialize the networks
while (Transmission==continue) do
    Share beacons and find location
    i=1
    while (i≤$|D|$) $< parallel >$ do
        Calculate $\updownarrow_{i,j}^{(r)}$
        Input metrics from neighboring drones
        if ($\updownarrow_{i,j}^{(r)} > \left(\updownarrow_{i,j}^{(r)})\right)_{(TH)}$ then
            Collision possibilities=true
            Update incidence and adjacency matrices
        else
            Continue with the present state
            if collision avoidance criteria satisfied then
                Calculate $\mathcal{C}_{i,j}^{(r)}$
                Input metrics from neighboring drones
                if ($\mathcal{C}_{i,j}^{(r)} > \left(\mathcal{C}_{i,j}^{(r)})\right)_{(TH)}$ then
                    congestion_window=congestion_window-1
                else
                    congestion_window=congestion_window+1
                end if
            end if
            if collision avoidance and congestion control criterion are satisfied then
                Initialize traffic and set timing diagram
                Check for timing diagram and available slots
                add connected channels to route matrix route[]
            else
                remove connected channels from route matrix (route[])
            end if
        end if
        i=i+1
    end while
end while
Operate for all channels and recheck conditions
```

proceed to check for route while maintaining conditions related to the QoS requirements. The overall algorithmic complexity of this algorithm is $O(D * \tau)$.

7 Experimental evaluation and discussion

The proposed scheme is evaluated on the basis of three directions, (1) blockchain validation for smart contracts, (2) numerical validation, and (3) QoS validation based on simulation experiments.

7.1 Blockchain validation using ethereum-based test network

The smart contacts are used in the proposed framework to ensure the integrity of the system resulting in trust of the involved entities. The experimental setup for validation is elaborated below.

- *Environment setup* The logic used in the proposed scheme is translated in the form of smart contacts, which are executed through remix IDE [37] in combination with

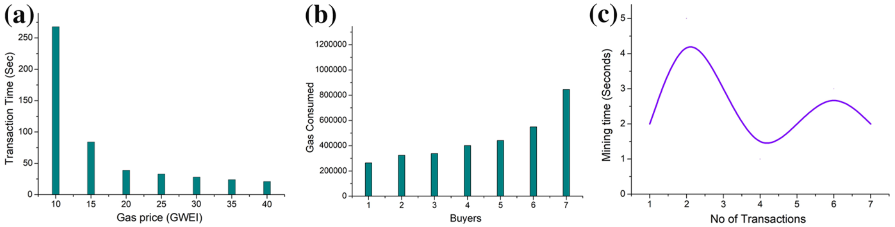

Fig. 6 **a** Gas price versus transaction time, **b** Gas consumed versus number of buyers, and **c** No of transactions versus mining time

web3 scripting environment. The performance is evaluated on an Ethereum-based test network [38]. Metamask [39] account has been setup to access the Ropsten test network wherein the smart contacts are deployed. This environment is setup on a personal machine with a fixed hardware configuration configuration (Intel i7 7700HQ, 16GB RAM, PC4-19200 DDR4 2400MHz, Overclocked to 3.8GHz, Nvidia GeForce GTX 1060 (4GB)).

The effectiveness of the smart contracts, which is regarded as mileage according to Ethereum terminology, is evaluated through the price paid for the gas for deployment of the smart contracts. The first tested smart contact is for the interaction between the buyer and the seller. The results obtained are depicted in Fig. 6a, suggesting the effect of the paid gas price on the transaction throughput. With more cost paid for gas per unit transaction, the transaction throughput increases. The improvement ratio in transaction though is more around the lower end, whereas moving towards expensive transactions does not improve the throughput proportionately. Hence, to meet the requirements of the QoS, the gas amount has to be chosen accordingly to control the deployability of the proposed approach. Next, we validated the effect of the consecutive number of buyers in the proposed framework on the amount of gas consumption for smart contract execution. The findings are depicted in Fig. 6b, representing the trend that with an increase in the number of transactions, the gas requirement and hence the cost of deployment tends to increase in a controlled manner. In the end, the model is validated for the mining time with respect to the number of transactions that are simultaneously committed and the results are depicted in Fig. 6c. The results suggests the non linear variation of mining time which can be attributed to the miners. The possible explanations for the variation include the nonce whose computation is probabilistic in nature. Sometime nonce is computed very quickly, whereas sometimes it takes a longer set of combinations to be tried by the miners before a successful resultant. These validations give valuable insights into the resource consumption and the load of computation related to CovaDel.

7.2 Numerical validation based on computation and communication costs

The proposed model witness number of interactions between different parties. So, we compute the computational and communication costs related to these parties.

7.2.1 Computational cost

The computational cost associated for each phase is calculated as below.

- *Initial agreement and smart contract phase* In the initial agreement phase, four entities $\mathbf{B}_j$, $\mathbf{S}_i$, **CS** and **W** are involved. $\mathbf{B}_j$ perform 5 append operations taking 0.6 ms in total and one digital signature computation which includes the encryption of $\mathbf{SC}_k$ using the private key of buyer costing 1.2 ms. **CS** performs one search operation requiring at least 2 ms which tends to increase with the larger database and 4 append operations requiring 0.45 ms. $\mathbf{S}_i$ requires 3 append operations to generate the invoice requiring 0.3ms. Lastly, the **W** performs one search operation requiring 2 ms. Hence, the total computational cost in this phase is 6.55 ms.
- *Delivery phase* In this phase, $\mathbf{B}_j$ performs one comparison operation to check the transaction from $\mathbf{T}_{\mathbf{SC'}_k}$ requiring 0.5ms. When the order gets delivered, $\mathbf{B}_j$ needs to generate $\mathbf{O}_k^{cmp}$ to indicate the receipt of product which requires 0.2 ms. $\mathbf{S}_i$ needs to do one comparison operation in order to validate the $\mathbf{T}_{\mathbf{SC'}_k}$ which is to be added to the blockchain requiring 0.5 ms. **CS** adds the $\mathbf{T}_{\mathbf{SC'}_k}$ of $\mathbf{SC}_k$ into blockchain for which it first adds it into the $\mathbf{T}_x^{nm}$ requiring 0.2 ms. After verification by $\mathbf{B}_j$ and $\mathbf{S}_i$, the $\mathbf{T}_{\mathbf{SC'}_k}$ is finally mined into the block requiring 2 ms and added to the $\mathbf{BC}^D$. Then, the extraction of $\mathbf{SH}_{\mathbf{O}_x}$ from the smart contract requires 0.3 ms. The smart contract is added to the blockchain requiring 1 ms. After delivery the smart contract is executed by the blockchain requiring 2 ms. **DS** executes the drone scheduling algorithm for which 5 comparison operations are performed requiring 2.5 ms, 6 addition operations requiring 1.8 ms, and searching requiring 0.5 ms. $\mathbf{D}_i$ collects the information from $\mathbf{BC}^D$ and then processes the flight plan requiring 2 ms. It also updates its status on the $\mathbf{BC}^D$ which needs 0.4 ms. Cumulatively, the computational cost of the delivery phase accounts to 12.9 ms.
- *Payment phase* The final phase of payment processing requires the **BC** to perform one subtraction operation requiring 0.3 ms and one addition requiring 0.3 ms. The **CS** needs to perform the trigger on $\mathbf{SC}_k$ requiring 0.2 ms. Hence, the total cumulative computational cost of the payment phase is 0.8ms.

This brings the total computational cost of all the three phases to 20.25 ms.

7.2.2 Communication cost

The communication cost for different phases is calculated as below:

- *Initial agreement and smart contract phase* The first phase requires the communication between $\mathbf{B}_j$ and **CS** for $\mathbf{O}_k^{ck}$ which is of 39 bits with 32 bits for $\mathbf{S}_i$, 4 bits for $\mathbf{I}_j^{qnty}$ and 3 bits for $\mathbf{I}_j^{sm}$. When the **CS** prepares the $\mathbf{SC}_k$, it comprising 32 bits $\mathbf{B}_j$, 32 bits $\mathbf{S}_i$, 32 bits $\mathbf{I}_j$ and 39 bits $\mathbf{O}_k^{ck}$. Hence, 135 bits are transmitted by the **CS**. $\mathbf{S}_i$ prepares the $\mathbf{IV}_{\mathbf{O}_x}$ comprising of $\mathbf{I}_j^{qnty}$ of 4 bits, $\mathbf{I}_j^{sm}$ of 3 bits and 32 bits $\mathbf{S}_i$. Hence, these 39 bits are transmitted by $\mathbf{S}_i$. Hence, the total communication cost involved in this phase is 213 bits.
- *Delivery phase* Here, **CS** sends 135 bit $\mathbf{T}_{\mathbf{SC'}_k}$ to the $\mathbf{B}_j$ and $\mathbf{S}_i$ for verification of transaction. $\mathbf{S}_i$ and $\mathbf{B}_j$ flags the validity of $\mathbf{T}_{\mathbf{SC'}_k}$ via 1 bit response each. Afterwards, the 135 bit $\mathbf{T}_{\mathbf{SC'}_k}$ is sent to $\mathbf{BC}^D$. Next, **CS** extracts $\mathbf{SH}_{\mathbf{O}_x}$ from $\mathbf{SC}_k$ which

is of 64 bits. This 64 bits $\mathbf{SH}_{\mathbf{O}_x}$ is sent to $\mathbf{W}$ and $\mathbf{DS}$. $\mathbf{DS}$ sends the 64 bits $\mathbf{SH}_{\mathbf{O}_x}$ to $\mathbf{D}_i$ chosen drone drone scheduling algorithm. Finally, the $\mathbf{B}_j$ sends the $\mathbf{O}_k^{cmp}$ to $\mathbf{CS}$ which is of 9 bit including 8 bits for hash of $\mathbf{SC}_k$ and 1 bit for indicating the completion. These 9 bits $\mathbf{O}_k^{cmp}$ is further sent by $\mathbf{CS}$ to $\mathbf{BC}$ which accounts to 354 bits.

– *Payment phase* This phase comprises of $\mathbf{BC}$ reserving $\mathbf{VC}$ corresponding to $\mathbf{SC}_k$ from $\mathbf{B}_j$'s $\mathbf{VC}$ and sends 32-bit balance ($\mathbf{VC}_{\mathbf{B}_j}$) to $\mathbf{B}_j$. On delivery acknowledgement of 9 bit, $\mathbf{BC}$ adds $\mathbf{VC}_{\mathbf{S}_i}$ to $\mathbf{S}_i$ for which the updated $\mathbf{VC}_{\mathbf{S}_i}$ of 32 bits is sent to $\mathbf{S}_i$. The total communication cost becomes 73 bits.

The communication cost of the proposed scheme is 640 bits per item.

7.3 Simulation study for QoS validation

To understand the performance of the proposed scheme, the validation is performed through a simulation study performed using Network Simulator (NS-2). The length of RTS, CTS and ACK packets considered for the evaluations is 170, 120, and 120 bits, respectively. The physical layer header and default packet lengths are same as 802.11b (pause time 2s). The performance results are evaluated on the basis of network throughput and end-to-end delay. To guarantee the QoS satisfaction, a maximum resource utilization must be achieved with an increase in the average transmission rate. The network throughput increases when the achieved utilization rate nears the permissible rate. However, the average network throughput decreases when we increase the number of users. The proposed scheme aims to handle the above challenge by sustaining the average network throughput to its best, even when the number of drones are increased. Fig. 7a shows the variation of the average network throughput with respect to an increase in the number of drones (10, 15, 20) and the number of users (100, 200, 300) requesting delivery.

The end-to-end transmission delay is dependant on many factors and it is not possible to control all these factors. But, the major contributors to the overall delay can be considered as processing and queuing tasks. If both of these tasks are controlled effectively, then it is possible to reduce the overall end-to-end delay. The proposed approach provides a very simple processing model supported by a multi-level queuing scheme for drone scheduling and product delivery. The proposed congestion and collision-aware transmission scheme helps to reduce the overall end-to-end delay in the network. Fig. 7b shows the variation of average delay with respect to an increase in the number of drones (10, 15, 20) and the number of users (100, 200, 300). It depicts that the delay increases with an increase in the number of users in contrast to the number of drones. If we increase the number of drones and reduce the number of users, then a lower delay is observed.

7.4 Complexity comparison in contrast with existing similar works

Various existing proposals have explored the problems related to drones but most of them have not considered the drone delivery case. Here, the proposed work is compared

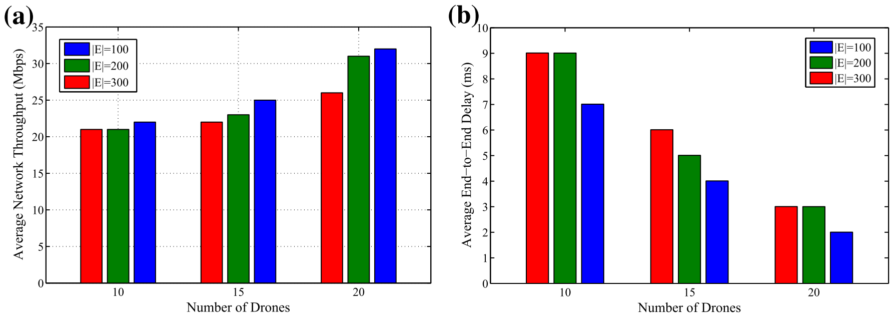

Fig. 7 **a** Average network throughput, and **b** Average end-to-end delay

Table 2 Comparison with similar existing works

	[40]	[41]	[42]	Proposed
Drone indexing	$\theta(k * n)$	–	–	$\theta(n)$
Drone scheduling	O(TP+n)	O(n^3)	O(TP+n)	O(TP+n)
QoS	–	–	–	O(D*τ)

with the existing works in terms of the complexities of the underlying approaches. The findings are summarized in the Table 2.

8 Conclusion

The proposed framework leverages the advantages of blockchain and smart contracts resulting into more transparent delivery operations COVID-19 like situations. Here, a derived blockchain is proposed for the part of the model where the normal blockchain will not be very suitable candidate still preserving the heritage of immutability of blockchain. In this framework, a virtual currency-based transactions are presented eliminating the reliance on third parties for payment processing hence eliminating the extra burden in terms of monetary cost. One of the key contribution comes from the drone scheduling algorithm which is backed by multi level queuing model ensuring optimal allocation of drones for various delivery jobs with minimal overhead. Incorporation of the QoS aware drone to drone communication approach has brought forward advantage of meeting the most challenging service levels in such scenarios. The proposed framework have been verified and validated for its reliability and performance in a simulated environment. The results indicate the performance of the system is effective. Further, the model is evaluated theoretically for the communication and computation costs which also favours the model.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Seo S-H, Won J, Bertino E, Kang Y, Choi D (2016) A security framework for a drone delivery service. Association for Computing Machinery, New York
2. Rosen JW (2020) "Zipline's ambitious medical drone delivery in africa,". [Online]. Available: https://www.technologyreview.com/2017/06/08/151339/blood-from-the-sky-ziplines-ambitious-medical-drone-delivery-in-africa/
3. "Eleven teams with flirtey for first ever faa-approved drone delivery to customer's home - friday, july 22, 2016," Jul 2016. [Online]. Available: https://corp.7-eleven.com/corp-press-releases/07-22-2016-7-eleven-teams-with-flirtey-for-first-ever-faa-approved-drone-delivery-to-customer-s-home
4. Lagkas T, Argyriou V, Bibi S, Sarigiannidis P (2018) Uav iot framework views and challenges: towards protecting drones as things. Sensors 18(11):4015
5. Ozmen MO, Yavuz AA (2018) Dronecrypt-an efficient cryptographic framework for small aerial drones. In: IEEE military communications conference, pp 1–6
6. Jindal A, Aujla GSS, Kumar N, Villari M (2019) Guardian: blockchain-based secure demand response management in smart grid system. In: IEEE transactions on services computing, pp 1–1
7. Chaudhary R, Jindal A, Aujla GS, Aggarwal S, Kumar N, Choo KKR (2019) Best: blockchain-based secure energy trading in SDN-enabled intelligent transportation system. Comput Secur 85:288–299
8. Ferrag MA, Maglaras L (2019) Deliverycoin: an ids and blockchain-based delivery framework for drone-delivered services. Computers 8(3):58
9. Alsamhi SH, Lee B, Guizani M, Kumar N, Qiao Y, Liu X (2021) Blockchain for decentralized multi-drone to combat covid-19 and future pandemics: framework and proposed solutions. Trans Emerg Telecommun Technol 32(9):e4255
10. Singh M, Aujla GS, Bali RS (2020) Odob: one drone one block-based lightweight blockchain architecture for internet of drones. In: IEEE conference on computer communications workshops (INFOCOM WKSHPS)
11. Boysen N, Briskorn D, Fedtke S, Schwerdfeger S (2018) Drone delivery from trucks: drone scheduling for given truck routes. Networks 72(4):506–527
12. Mbiadou Saleu RG, Deroussi L, Feillet D, Grangeon N, Quilliot A (2018) An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. Networks 72(4):459–474
13. Manrique P, Johnson D, Johnson N (2017) Using competition to control congestion in autonomous drone systems. Electronics (Switzerland) 6:06
14. Vashisht S, Jain S (2019) An energy-efficient and location-aware medium access control for quality of service enhancement in unmanned aerial vehicular networks. Comput Electr Eng 75:202–217
15. Singh M, Aujla GS, Bali RS, Vashisht S, Singh A, Jindal A (2020) Blockchain-enabled secure communication for drone delivery: a case study in covid-like scenarios. Association for Computing Machinery, New York
16. Scott J, Scott C (2017) Drone delivery models for healthcare. In: Proceedings of the 50th Hawaii international conference on system sciences
17. Park J, Kim S, Suh K (2018) A comparative analysis of the environmental benefits of drone-based delivery services in urban and rural areas. Sustainability 10:888, 03
18. Chang YS, Lee HJ (2018) Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst Appl 104:307–317
19. Kim J, Moon H, Jung H (2020) Dronebased parcel delivery using the rooftops of city buildings: model and solution. Appl Sci 10(12):4362
20. Hooper M, Tian Y, Zhou R, Cao B, Lauf AP, Watkins L, Robinson WH, Alexis W (2016) Securing commercial wifi-based uavs from common security attacks. In: IEEE military communications conference. IEEE, pp 1213–1218
21. Lützen TH (2017) A security framework for unmanned aerial vehicles and practical exploitation analysis
22. Russell B, Van Duren D (2018) Practical internet of things security: design a security framework for an Internet connected ecosystem. Packt Publishing Ltd

23. Alladi T, Chamola V, Sahu N, Guizani M (2020) Applications of blockchain in unmanned aerial vehicles: a review. Veh Commun 23:100249
24. https://www.europeanpharmaceuticalreview.com/article/103799/medicine-delivery-drone-safety-andquality/, Medicine delivery by drone, Tech. Rep
25. https://www.eaglehawk.io/drone-enabled-disinfectantspraying, Drone-enabled disinfectant spraying, Tech. Rep
26. https://www.webmd.com/a-to-z-guides/news/20190429/world-first-drone-delivers-kidney-fortransplant, Drone deliver kidney for transplant, Tech. Rep
27. https://www.dhl.com/discover/business/business-ethics/parcelcopter-dronetechnology, Dhl delivery by drones, Tech. Rep., (2016)
28. https://anavia.eu/en/, Drone delivery, Tech. Rep
29. https://www.amazon.com/AmazonPrime-Air/b?ie=UTF8&node=8037720011, Prime delivery by drones, Tech. Rep
30. https://www.fdrones.com, Maritime delivery by drones, Tech. Rep., (2016)
31. https://techcrunch.com/2016/12/20/7-eleven-delivers-77-packages-via-drone-in-first-month-of-routineservice/, 7-eleven delivery by drone, Tech. Rep
32. https://qz.com/838254/dominos-is-delivering-pizza-with-autonomous-drones-to-customers-in-newzealand/, Dominos pizza delivery, Tech. Rep., (2016)
33. https://tacocopter.com, Flying robots deliver tacos to your location, Tech. Rep
34. Singh G, Singh A, Singh M, Sharma S, Kumar N, Choo KR (2020) Blocked: blockchain-based secure data processing framework in edge envisioned v2x environment. In: IEEE transactions on vehicular technology, pp 1–1
35. Yang X-S (2010) Firefly algorithm, levy flights and global optimization. In: Research and development in intelligent systems XXVI. Springer, pp 209–218
36. Tuba E, Tuba M, Beko M (2018) Two stage wireless sensor node localization using firefly algorithm. In: Yang X-S, Nagar AK, Joshi A (eds) Smart trends in systems, security and sustainability. Springer, Singapore, pp 113–120
37. Ethereum ide. [Online]. Available: https://remix.ethereum.org/
38. Zhang K, Jacobsen H-a (2018) Towards dependable, scalable, and pervasive distributed ledgers with blockchains, 07, pp 1337–1346
39. Metamask. [Online]. Available: https://metamask.io/
40. Li Q, Meng S, Sang X, Zhang H, Wang S, Bashir AK, Yu K, Tariq U (2021) Dynamic scheduling algorithm in cyber mimic defense architecture of volunteer computing. ACM Trans Internet Technol 21(3):1–33
41. Huang H, Savkin AV, Huang C (2020) Scheduling of a parcel delivery system consisting of an aerial drone interacting with public transportation vehicles. Sensors 20(7):2045
42. Shavarani SM, Mosallaeipour S, Golabi M, İzbirak G (2019) A congested capacitated multi-level fuzzy facility location problem: an efficient drone delivery system. Comput Oper Res 108:57–68

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.