Secure Heterogeneous Multi-Robot Collaboration and Docking with Hyperledger Fabric Blockchain

Salma Salimi[†], Paola Torrico Morón[†], Jorge Peña Queralta[†], Tomi Westerlund[†]

[†]Turku Intelligent Embedded and Robotic Systems (TIERS) Lab, University of Turku, Finland.
Emails: [1]{salmas, pctomo, jopequ, tovewe}@utu.fi

***Abstract*—In recent years, multi-robot systems have received increasing attention from both industry and academia. Besides the need of accurate and robust estimation of relative localization, security and trust in the system are essential to enable wider adoption. In this paper, we propose a framework using Hyperledger Fabric for multi-robot collaboration in industrial applications. We rely on blockchain identities for the interaction of ground and aerial robots, and use smart contracts for collaborative decision making. The use of ultra-wideband (UWB) localization for both autonomous navigation and robot collaboration extends our previous work in Fabric-based fleet management. We focus on an inventory management application which uses a ground robot and an aerial robot to inspect a warehouse-like environment and store information about the found objects in the blockchain. We measure the impact of adding the blockchain layer, analyze the transaction commit latency and compare the resource utilization of blockchain-related processes to the already running data processing modules.**

***Index Terms*—Robotics; ROS 2; Blockchain; Hyperledger Fabric; Multi-robot systems; Ultra-wideband (UWB); Inventory management; Fleet management; Distributed ledger technologies.**

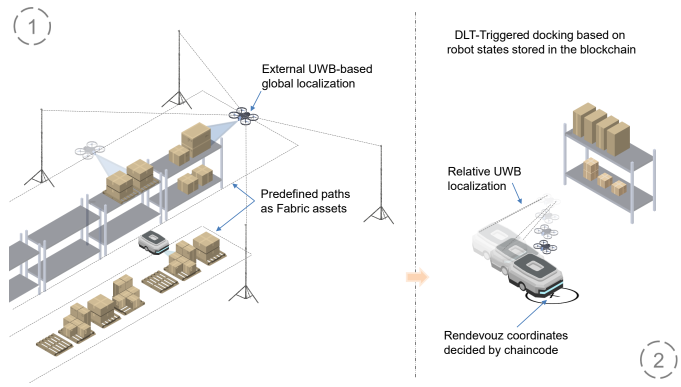


Fig. 1: Conceptual illustration of the proposed application scenario. The blockchain is used for high-level mission commands (predefined robot paths), for triggering multi-robot cooperative actions based on chain-recorded robot states (docking) and switching operation modes (activate/deactivate global/relative localization UWB beacons). Detected objects are also stored in the blockchain.

I. Introduction

Multi-robot systems, including aerial and ground robots, are used more extensively in recent years, gaining robustness and adaptability for their missions [1]. By utilizing heterogeneous multi-robot systems, a wide range of tasks with different characteristics can be autonomously and simultaneously accomplished in the same environment [2]. To make this happen, we need to seamlessly integrate multiple technologies. Integrating multiple technologies, security issues must be considered carefully as they are an intrinsic part of multi-disciplinary field of robotics [3].

Blockchain technology is enabling more secure and trustable distributed systems that multi-robot systems are in essence. Permissioned or private blockchains have special potential in industrial applications, allowing for transparent co-operation between organizations while ensuring data remains private. This is also an important consideration when managing large-scale systems of autonomous robots, which may need to cooperate with each other but also produce data that should remain private to a subset of parties. In this paper, we explore the potential of the Hyperledger Fabric blockchain for managing cooperation between robots in addition to managing individual autonomous robots in a fleet.

Furthermore, multi-robot systems should be able to eliminate false information presented through byzantine entities and/or resist hacking and manipulation of data and path planning systems that could result in disastrous outcomes. Blockchains provide a decentralized, secure and trusted platform that enables coordinating and allocating tasks, path-planning and controlling processes [4], [5], [6].

In distributed systems, blockchain technology can provide security [7], trust [8], data management [9], peer-to-peer transactions and a fault-tolerant middleware [10]. Based on how they handle user credentials, blockchain platforms can be classified into two main categories: (i) public or permissionless, and (ii) private, permissioned, or consortium blockchains [11]. However, we are particularly interested in private blockchains for managing private robot fleets and maintaining data produced by robots, or part of it, private.

Hyperledger Fabric is a private permissioned blockchain framework developed in Go. It has a modular architecture supporting plug-in components, and it also achieves data isolation via channels so that two or more peers can have private communication in the network [3], [12]. With highly modular, configurable, and pluggable features, developers are free to implement the technologies of their choice, and to do so in whatever way they want to (e.g., consensus protocols)[13].

Due to the increased potential for multi-modal and multi-source sensor fusion in multi-robot scenarios, and the flexibility and robust nature of deployment in complex environments, heterogeneous multi-robot systems and algorithms for collaborative autonomy are also getting increasing interest [14].

Collaboration between different ground or aerial vehicles is necessary to accomplish a mission. It is particularly essential that unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) coordinate during missions in remote areas where humans are in potentially hazardous situations[15]. Robots are capable of performing these tasks safely, as they can collect data from the environment easily, and also perform preprogrammed tasks by seeking around in certain scenarios. It is an interesting challenge to give them autonomous decision and also real-time cooperation and adaption with minimum intervention of human.

Ultra-Wideband (UWB) is a radio frequency technology that is used for indoor localization. Since these devices are more affordable than other indoor localization solutions, UWB technology is attracting increasing attention [16]. Anchors and tags are the two types of nodes used in UWB localization. To obtain the position of tags in most anchor-based scenarios, the locations of the anchors must be known. The coordinates of the fixed anchor nodes are assumed to be known by the system for this calculation. In the presence of enough fixed nodes, depending on the ranging method implemented, the tags can estimate their distances from each anchor and calculate their own location using algorithms such as multilateration [17].

In summary, this paper extends our previous works in blockchain-based fleet management with Hyperledger Fabric [18], leveraging the blockchain identities for collaboration between ground and aerial robots. Specifically, we utilize Fabric smart contracts for logging historical data, and collaborative decision-making. We put this approach into practice with an inventory management application where a ground and an aerial robot inspect a warehouse-like environment and store information about objects found in the blockchain. The smart contracts are then also used to enable docking of the aerial robot in the ground robot when the battery is too low. The robots use anchor-based UWB localization for following predefined trajectory, while the docking smart contract triggers the activation of anchors in the ground robot for more accurate relative localization while docking. These smart contracts can be extended for other collaborative tasks.

The rest of this manuscript is organized as follows. In Section II we introduce related works in blockchain solutions for robot fleets and UWB-based localization for multi-robot systems in GNSS-denied environments. Section III describes the system design, with Section IV focusing on the experimental settings and methodology. Section V reports experimental results and Section VI concludes the work.

II. Background

This section reviews the literature in the areas of blockchain technologies for multi-robot systems first, and then delves into specific applications to aerial and ground robots.

Considering time limitations, execution and autonomy requirements in multi-robot path planning approaches, authors in [19] present a distributed control system which is time-sensitive and is using the permissioned and private blockchain, Hyperledger Fabric platform. As a result, Hyperledger Fabric, which has been used in this paper, reveals less transactional latencies compared to permissionless, public blockchain platforms. It is also more flexible and still features distributed execution of logic that enforces the consensus attainment for the collaborative members by the use of smart contracts. From the perspective of building trust, and to provide a trusted edge collaborative inference environment, the work in [20] introduced a blockchain-based collaborative edge knowledge inference framework, which ensures the reliance of data sharing, avoids knowledge pollution and detects malicious nodes.

A challenge in collaborative multi-robot systems is organizing the communication in data exchanges. To optimize the amount and type of data on the exchange between them, a novel approach for managing the terms of these systems with blockchain technology has been illustrated by [21]. The proof of work systems for estimating the availability of computational resources of different robots have been proposed in this paper, and also smart contracts have been integrated to rank and analyze the quality and validity of each of the robots sensor data.

In a similar direction in terms of managing data transmissions, Guo et al. proposed the integration of edge computing on a blockchain framework to enable large amounts of data exchanges in a spherical multi-robot system[22], instead of overloading each robot's nodes. Additionally, the authors recommended a distributed data processing system that exploits blockchain and edge computing technologies, in order to solve byzantine fault-tolerance problems for entities with limited resources. These works lay the background for more complex, real-world deployments as we are introducing in this paper.

A key use case of blockchain technology in multi-robot systems is also role allocation and driving the collaboration between the robots. Blockchain-based solutions have been shown to increment interaction efficiency, providing more reliable data exchange, consensus in trustless situations, detecting performance issues, identifying intruders, assigning tasks, or deploying distributed solutions and joint missions [4]. This overview of applications shows the potential of blockchain technologies, but also identifies limitations of current solutions that we are addressing in this paper with a permissioned blockchain platform.

In another recent application, Castelló Ferrer et al. use a blockchain as an asynchronous registry of messages by leaders communicating with robots with gregarious attributes to obtain security, limited memory, and resilience to deception [23],. The authors present solutions for the Byzantine Follow Multiple Leaders and Byzantine Loosely Follow Multiple Leaders problems where it has been improved that the proposed method could be utilized in practical scenarios where there are resource limitations. We also study in this paper the impact in terms of resource utilization of the blockchain layer, which we show to be negligible in complex robotic systems.

In terms of blockchain applications to drones, UAVs are a type of aerial vehicle that in addition to being able to operate autonomously, also possess the ability to fly with preplanned flight missions or create their own plan mid-flight [13]. Blockchain technology has been used for protecting the privacy and security of the entire trade process in [24], where a Stackelberg dynamic game based trading scheme has been presented to solve the edge computing resource allocation issue between Edge Computing Stations (ECSs) and Unmanned Aerial Vehicles (UAVs).

The applications to UAVs are multiple and varied. In a relevant work more related to industrial use cases, and according to the high demand of decentralized solutions for enforcing regulations, satisfying cybersecurity requirements and a need for efficient air traffic management, Alkadi et al. have described several issues with the current UAV traffic management systems and presented a solution that relies on cooperation between the concepts of blockchain smart contracts and mobile crowdsensing [25] .

From the perspective of multi-robot system deployments, Aloaily et.al. introduced a network solution that fulfills the capabilities of UAVs and UGVs, as well as improving connectivity, service availability, and energy efficiency [26]. Additionally, the solution employs a service composition strategy to deliver user-specific services based on their requirements, and also a consensus algorithm is used to verify locally trained models with blockchain to support integrity and authenticity of sensitive services. Compared to previous works, in this paper we have used Fabric smart contracts for collaboration decision-making while logging historical data and leveraged the blockchain identities for multi-robot collaboration by presenting an inventory management application.

Finally, regarding practical considerations for the deployment of autonomous robots in GNSS-denied environments, there is a number of works in the literature that leverage UWB localization for inventory management and other indoor operations in warehouses [27], [28], [29]. These are just a few examples of wider use of UWB technology in GNSS-denied environments [27]. Relative localization between ground and aerial robots enabled by UWB ranging makes possible more efficient collaboration [30]. In comparison to other approaches of cooperative relative localization, wireless ranging provides high performance and low complexity of the system. UWB in particular offers unparalleled performance in unlicensed bands, resilience to multi-path, low interference with other radio technologies and also high time resolution where in this paper we switch between external localization for navigation and relative localization for autonomous docking of the aerial robots on the ground robots [31].

III. System overview

The growing interest in multi-robot systems is driven, in part, by the complexity of tasks and performance requirements. Multiple applications exist in a variety of scenarios, from emergency and rescue missions to surveillance applications and collaborative tracking [2], [32]. Integrating DLTs into such systems can benefit identity management, data sharing security, monitoring and multi-robot inventory management and consensus. Knowledge-sharing-based collaborative inference is often necessary in multi-robot systems to accomplish complex tasks. In order to protect the integrity of such data sharing, it is essential to establish a secure environment. With DLTs, security and privacy can be enhanced while ensuring preservation of data integrity.

A. Fleet management with Hyperledger Fabric

Hyperledger Fabric is private and permissioned blockchain which breaks it from other blockchain systems. Instead of a public and permissionless system that allows anyone to join network, Fabric members join through a trusted Member Service Provider (MSP). In industrial robot fleets, Hyperledger Fabric blockchains can be used to manage identity, secure control interfaces, auditable data flows and create private data channels [18]. Therefore, the main goal of this paper is cooperating of multi-robot systems for a specific purpose using Hyperledger Fabric network for identity management of robots in a fleet, secure data management and robots control. On the other hand, Fabric smart contracts has been used in this paper for storing path and mission parameters or collaboration decision making.

B. Localization and deployment

Autonomous operation in GNSS-denied environments can be achieved with a variety of onboard and external sensors or landmarks. In recent years, UWB positioning systems have become increasingly popular owing to their relative high accuracy at low price [27]. Autonomous robots can operate based on either external, global positioning [33], or relative positioning within multi-robot systems [30]. In this paper, we combine both localization approaches utilizing the blockchain as a channel for deciding when to switch between one mode or another.

In general terms, the core idea of the system we propose is that different modes of operation can be controlled via the Fabric smart contracts. For instance, robots can switch from individual autonomous operation to a behaviour involving collaboration. In our use-case scenario, this is exemplified by changing between external and relative localization methods. However, we present a generic architecture that can be adjusted to meet the needs of other application scenarios, from distribution of computation to role allocation.

C. Architecture

The proposed system architecture is illustrated in Fig. 2. We use the Fabric to ROS 2 connection introduced in [18] to connect the ROS 2 nodes running on the robots to the Fabric blockchain. This is done by implementing ROS 2 nodes in Go using *rclgo*[1]. Compared to our previous work, we delve into a more complex use case with higher number of chaincodes in

[1]https://github.com/tiiuae/rclgo

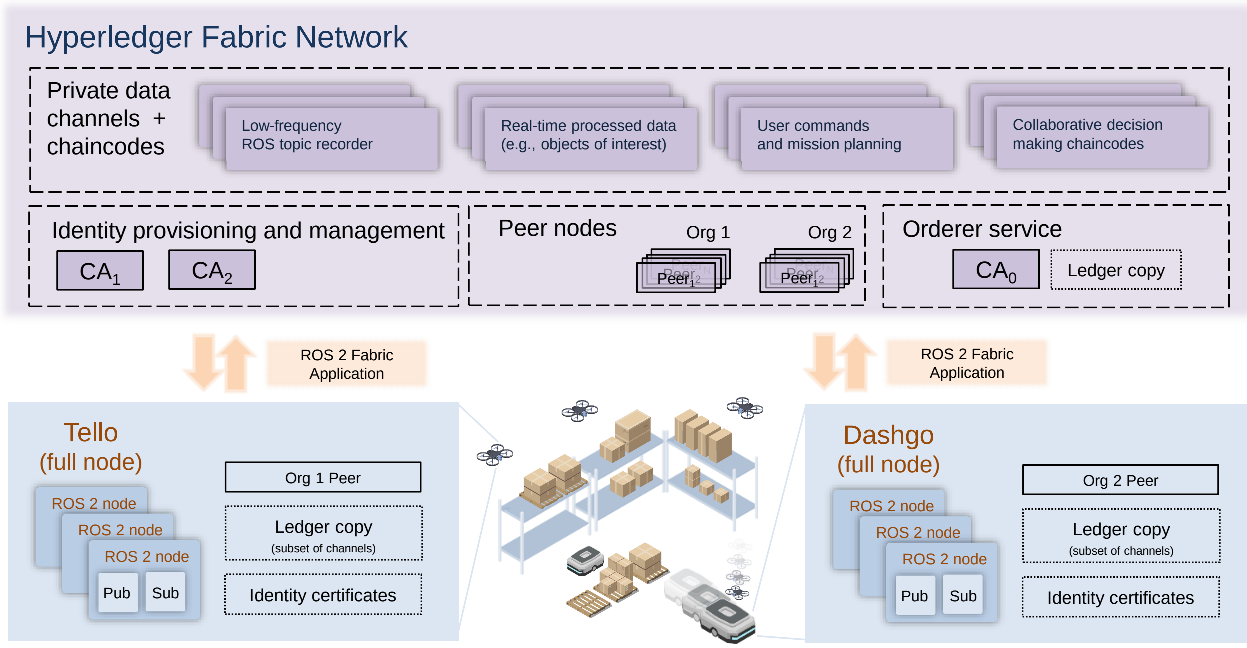


Fig. 2: Architectural diagram of the proposed framework.

this paper, representing a more realistic industrial application. The blockchain layer sits conceptually on top of the data processing and robot control software, driving the overall mission parameters, acting as a data recorder platform, and becoming the means for decision-making regarding real-time robot behaviour. It is worth noting that we do not aim at implementing low-level control through the blockchain layer, owing to the lack of determinism and real-time capabilities in the proposed architecture, but instead drive high-level behaviour: paths to follow, state machines defining the mission flow or decisions in regard to how or when robots are meant to cooperate.

IV. Methodology

This section describes the robots, hardware and software utilized in the experiments.

A. Experimental platforms

The experimental platform of this paper is consist of a commercially available Ryze Tello MAV and a ground robot. The drone is equipped with an UWB module for localization. We also utilize the drone camera for object detection, with the camera feed being available through a websocket connection to a controller computer. The ground robot is an EAI Dashgo platform equipped with an UP HD camera with an OV2735 sensor. An AAEON Up Xtreme with an Intel i7-8665UE processor is used as a companion computer on the Dashgo and RealSense T265 cameras has been used for VIO-based egomotion estimation.

For this experiment, four UWB nodes, Decawave's DWM1001 modules with custom firmware, had been deployed for robot localization and also five extra UWB nodes had been used on the Dashgo platform for more accurate docking of the Tello.

B. Software

The Dashgo robot runs ROS Melodic under Ubuntu 18.04 for the main driver. Localization and object detection are running in ROS 2 Foxy. Fig. 3 shows the different software modules running in different nodes. The fabric applications running onboard the robots are connected to peers running on a separate computer in the network with the same Intel i7 processor.

To forward the data from Melodic topics to Foxy topics, the $ros1\_bridge$ package is used under the same computer. Also, to obtain camera images of the dashgo at a frequency of 30 Hz even though they are forwarded to the object detector at 5 Hz, the usb_cam package has been used which is available in both Melodic and Foxy. The object detector used in the experiments is YOLOX[2], and a selection of objects are part of the categories in the COCO dataset which are used for the purpose of the inventory management.

To implement the different part of the system, the Go programming language (golang) has been used whenever possible to increase the potential for integration between the different parts of the system includes the smart contracts, applications and also ROS 2 nodes. To manage data securely and control robots, a private Hyperledger Fabric network has been set up.

To set up the private network, one orderer and two organizations including one peer each and corresponding CAs have

[2]https://github.com/Ar-Ray-code/YOLOX-ROS

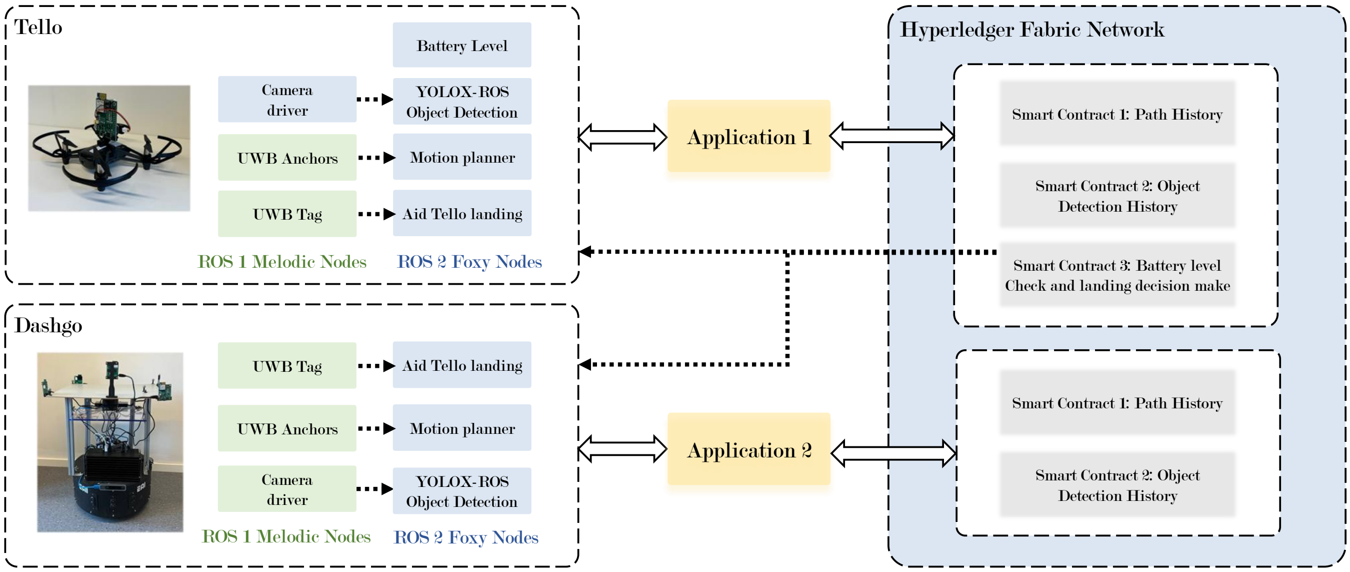


Fig. 3: System implementation diagram.

been created. Then after genesis block generation and certificates creations for each organizations, the docker container has been brought up. Channel genesis block has been created for each channel and then the peers of organizations have been joined to the channel. After packaging and installing the definition of the chaincodes on peers, they have been then approved for organizations. In the last step, the chaincode has been invoked after commitment of definitions.

C. Smart contracts

Five smart contracts have been implemented in the system: two for storing the path tracking of two robots, two for storing the location of the detected objects by both robots and one for updating the battery level of the Tello in the asset and landing decision making(see Fig. 3). One application containing different functions such as creating new assets, read, update and changing the assets has been used for each robot.

A rendezvous point is defined for drone docking. Dasgho and Tello start inspecting the environment following the predefined path given by the operator through DLT where robot paths history and detected objects are stored in. According to the battery level of the Tello, when the battery level decreases below a threshold , the smart contract sends a docking order to the robots. When the ground robot accepts the drone for landing and charging, the position of the rendezvous point is given to the robots by smart contract. After the robots movement to a rendezvous point, the UWB receivers on the Dashgo platform are enabled to have relative localization and accurate docking for Tello drone, so external anchors are not used anymore for localization of the Tello drone.

D. Metrics

In this paper our focus is on measuring and recording the data while using the Hyperledger Fabric blockchain for multi-robot collaboration to check the robots path. We analyze the memory and CPU usage of YOLOX and Fabric during the experiment and also latency of committing the transactions in the blockchain. Finally, we report the distribution of transaction latency in the blockchain layer.

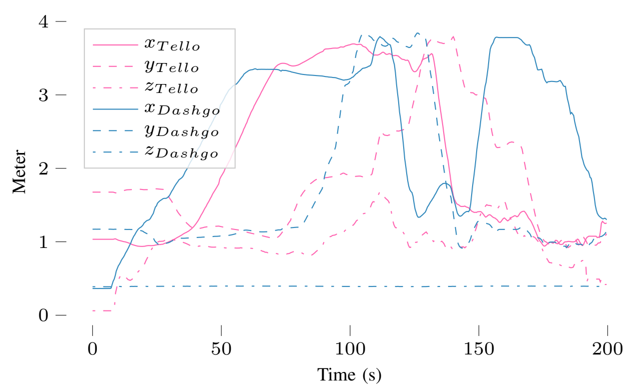
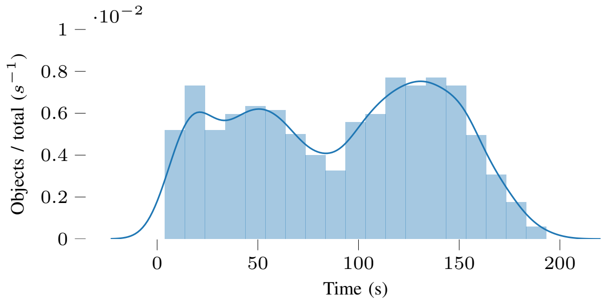


Fig. 4: Trajectory components for each of the robots. At the end of the mission, the robot trajectories converge as the Tello docks on top of the Dashgo.

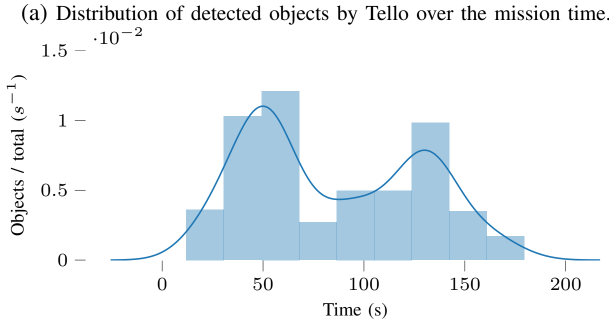
V. Experimental Results

In this section we report the experimental results obtained by data from two robots which are following a predefined path for the purpose of detecting objects and storing them in a Fabric channel chaincode to manage the inventory. For this, UWB nodes, series of shelves and various objects from the COCO dataset categories are deployed in a $40\,m^2$ room. External UWB anchores are installed on the Dashgo platform, so that the Tello be able to have more accurate landing on Dashgo for charging.

Figure 4 shows the trajectories of the robots during the experiment. Towards the mission end, as it can be seen after



(a) Distribution of detected objects by Tello over the mission time.



(b) Distribution of detected objects by Dashgo over the mission time.

Fig. 5: Figure (a) shows the distribution of detected objects by Tello, and (b) shows the distribution of detected objects by Dashgo.

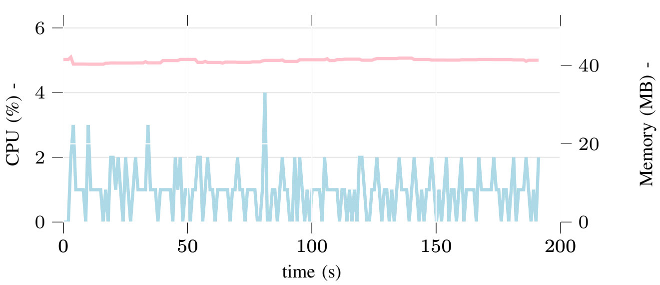


Fig. 6: Go applications activity during the mission where CPU usage is shown in blue and memory in pink (100% CPU / processor core).

more than three minutes, the positions converge when the Tello drone lands on the Dashgo. Figures 5a and 5b show the distribution in time of the objects detected by Tello and Dashgo, respectively.

In Figure 6, the CPU and memory utilization of the Go application during the experiments can be seen, while Figure 7 shows YOLOX resources utilization. We examined how integrating the Fabric network could affect an existing ROS 2 system, in terms of computational resources. Taking these results into account, we can conclude that the addition of Fabric as an additional data sharing channel is negligible, and that the proposed framework has the potential to be implemented on a variety of robotic platforms and application scenarios.

Asset creation is a key parameter affecting the system latency because each transaction is only confirmed when it appears in a block. Therefore, in addition to the resource utilization, we have measured the latency of storing data in

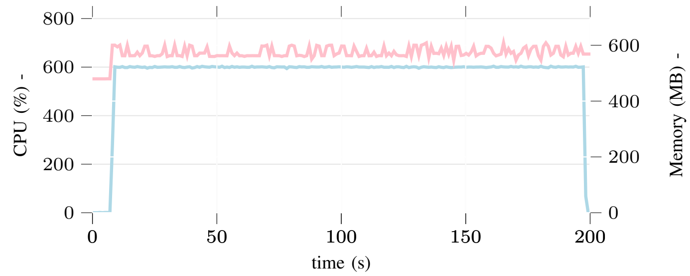


Fig. 7: YOLOX ROS 2 node activity during the mission where CPU usage is shown in blue and memory in pink (100% CPU / processor core).

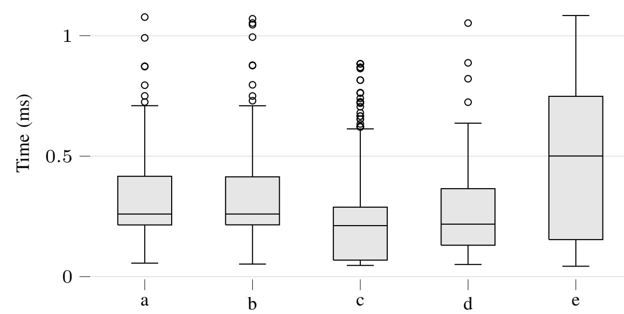


Fig. 8: Distribution of the latency for committing transactions between the robot and the peer node where (a) shows the chaincode recording the Tello battery level, (b) shows the chaincode recording the Tello path history, (c) stores objects detected by the Tello drone, (d) shows the Dashgo path history recorder and (e) shows the storage of objects detected by Dashgo.

the blockchain while application nodes are running on the robots, and peers and orderer are running in another computer in the same network. Figure 8 shows the latency distribution of the five smart contracts where over 200 HZ of ROS 2 data has been stored in average. The main parameters affecting the transaction commit latency are the maximum amount of transactions to be included in a block, and the maximum timeout for a transaction to be included. A block is thus created whenever one or another amount is reached. An initial study on the performance of Fabric with different parameters has already been shown in [18]. In this paper, we analyze the transaction latency distribution to illustrate its relation to the data types and frequencies in each chaincode. Path history and Tello battery are recorded at a constant, pre-defined frequency, and thus the distributions are more similar. The values are not constant because network delays or other transactions still affect the process. The Tello object detection chaincode has slightly lower latency owing to the higher frequency of the data over most of the mission (see Fig. 5a). At the same time, the Dashgo distribution is wider as the recording frequency is more irregular (see Fig. 5b.

VI. Conclusion

In this paper, a framework leveraging blockchain technology for multi-robot collaboration in industrial applications for the purpose of identity management, data sharing security, monitoring and multi-robot inventory management and consensus has been presented. DLTs has been used in order to protect the data sharing in multi-robot collaboration, while providing samples of applications transferring data between ROS 2 nodes and smart contracts in Hyperledger Fabric. On the other hand, for having high performance and low system complexity, robots localization and landing have been done through UWB positioning. According to the experimental results, it has been shown that this framework provides great amount of security and identity management features, and at the same time adding the blockchain layer has minimal impact on the utilization of computational resources.

ACKNOWLEDGMENT

This research work is supported by the Academy of Finland's RoboMesh and AeroPolis projects (grant numbers 336061 and 348480), and in part by the R3Swarms project funded by the Secure Systems Research Center (SSRC), Technology Innovation Institute (TII).

REFERENCES

[1] Yara Rizk, Mariette Awad, and Edward W Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2), 2019.

[2] Jorge Peña Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access*, 8, 2020.

[3] USP Srinivas Aditya, Roshan Singh, Pranav Kumar Singh, and Anshuman Kalla. A survey on blockchain in robotics: Issues, opportunities, challenges and future directions. *Journal of Network and Computer Applications*, 196, 2021.

[4] Ilya Afanasyev, Alexander Kolotov, Ruslan Rezin, Konstantin Danilov, Manuel Mazzara, Subham Chakraborty, Alexey Kashevnik, Andrey Chechulin, Aleksandr Kapitonov, Vladimir Jotsov, et al. Towards blockchain-based multi-agent robotic systems: Analysis, classification and applications. *arXiv preprint arXiv:1907.07433*, 2019.

[5] Volker Strobel, Eduardo Castelló Ferrer, and Marco Dorigo. Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario. 2018.

[6] Volker Strobel, Eduardo Castelló Ferrer, and Marco Dorigo. Blockchain technology secures robot swarms: A comparison of consensus protocols and their resilience to byzantine robots. *Frontiers in Robotics and AI*, 7:54, 2020.

[7] Yongfeng Qian, Yingying Jiang, Jing Chen, Yu Zhang, Jeungeun Song, Ming Zhou, and Matevž Pustišek. Towards decentralized iot security enhancement: A blockchain approach. *Computers & Electrical Engineering*, 72, 2018.

[8] Juah C Song, Mevlut A Demir, John J Prevost, and Paul Rad. Blockchain design for trusted decentralized iot networks. In *2018 13th Annual Conference on System of Systems Engineering (SoSE)*. IEEE, 2018.

[9] Gbadebo Ayoade, Vishal Karande, Latifur Khan, and Kevin Hamlen. Decentralized iot data management using blockchain and trusted execution environment. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, 2018.

[10] Jollen Chen. Devify: Decentralized internet of things software framework for a peer-to-peer and interoperable iot device. *ACM SIGBED Review*, 15(2), 2018.

[11] Jorge Peña Queralta and Tomi Westerlund. Blockchain for mobile edge computing: Consensus mechanisms and scalability. In *Mobile Edge Computing*. Springer, 2021.

[13] Paulo Álvares, Lion Silva, and Naercio Magaia. Blockchain-based solutions for uav-assisted connected vehicle networks in smart cities: A review, open issues, and future perspectives. In *Telecom*, volume 2. Multidisciplinary Digital Publishing Institute, 2021.

[12] Jorge Peña Queralta, Li Qingqing, Zhuo Zou, and Tomi Westerlund. Enhancing autonomy with blockchain and multi-access edge computing in distributed robotic systems. In *FMEC*. IEEE, 2020.

[14] Jorge Peña Queralta, Li Qingqing, Tuan Nguyen Gia, Hong-Linh Truong, and Tomi Westerlund. End-to-end design for self-reconfigurable heterogeneous robotic swarms. In *DCOSS*. IEEE, 2020.

[15] Mahmoud Hussein, Réda Nouacer, Federico Corradi, Yassine Ouhammou, Eugenio Villar, Carlo Tieri, and Rodrigo Castiñeira. Key technologies for safe and autonomous drones. *Microprocessors and Microsystems*, 87, 2021.

[16] Matteo Ridolfi, Abdil Kaya, Rafael Berkvens, Maarten Weyn, Wout Joseph, and Eli De Poorter. Self-calibration and collaborative localization for uwb positioning systems: A survey and future research directions. *ACM Computing Surveys (CSUR)*, 54(4), 2021.

[17] Matteo Ridolfi, Samuel Van de Velde, Heidi Steendam, and Eli De Poorter. Analysis of the scalability of uwb indoor localization solutions for high user densities. *Sensors*, 18(6), 2018.

[18] Salma Salimi, Jorge Peña Queralta, and Tomi Westerlund. Hyperledger fabric blockchain and ros 2 integration for autonomous mobile robots. *arXiv preprints*, 2022.

[19] Amr Mokhtar, Noel Murphy, and Jennifer Bruton. Blockchain-based multi-robot path planning. In *WF-IoT*. IEEE, 2019.

[20] Jianan Li, Jun Wu, Jianhua Li, Ali Kashif Bashir, Md Jalil Piran, and Ashiq Anjum. Blockchain-based trust edge knowledge inference of multi-robot systems for collaborative tasks. *IEEE Communications Magazine*, 59(7), 2021.

[21] Jorge Peña Queralta and Tomi Westerlund. Blockchain-powered collaboration in heterogeneous swarms of robots. *arXiv preprint arXiv:1912.01711*, 2019.

[22] Shuxiang Guo, Sheng Cao, Jian Guo, and Jigang Xu. Study on distributed data processing system for decentralized spherical multi-robot based on edge computing and blockchain. In *ICMA*. IEEE, 2020.

[23] Eduardo Castelló Ferrer, Ernesto Jiménez, Jose Luis Lopez-Presa, and Javier Martín-Rueda. Following leaders in byzantine multirobot systems by using blockchain technology. *IEEE Transactions on Robotics*, 2021.

[24] Haitao Xu, Wentao Huang, Yunhui Zhou, Dongmei Yang, Ming Li, and Zhu Han. Edge computing resource allocation for unmanned aerial vehicle assisted mobile network with blockchain applications. *IEEE Transactions on Wireless Communications*, 20(5), 2021.

[25] Ruba Alkadi and Abdulhadi Shoufan. Unmanned aerial vehicles traffic management solution using crowd-sensing and blockchain. *arXiv preprint arXiv:2110.14979*, 2021.

[26] Moayad Aloqaily, Ismaeel Al Ridhawi, and Mohsen Guizani. Energy-aware blockchain and federated learning-supported vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[27] Yu Xianjia, Li Qingqing, Jorge Peña Queralta, Jukka Heikkonen, and Tomi Westerlund. Applications of uwb networks and positioning to autonomous robots and industrial systems. In *MECO*. IEEE, 2021.

[28] Nicola Macoir, Jan Bauwens, Bart Jooris, Ben Van Herbruggen, Jen Rossey, Jeroen Hoebeke, and Eli De Poorter. Uwb localization with battery-powered wireless backbone for drone-based inventory management. *Sensors*, 19(3), 2019.

[29] Yu Xianjia, Li Qingqing, Jorge Peña Queralta, Jukka Heikkonen, and Tomi Westerlund. Cooperative uwb-based localization for outdoors positioning and navigation of uavs aided by ground robots. In *2021 IEEE International Conference on Autonomous Systems*. IEEE, 2021.

[30] Jorge Peña Queralta, Li Qingqing, Fabrizio Schiano, and Tomi Westerlund. Vio-uwb-based collaborative localization and dense scene reconstruction within heterogeneous multi-robot systems. In *IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2022.

[31] Wang Shule, Carmen Martínez Almansa, Jorge Peña Queralta, Zhuo Zou, and Tomi Westerlund. Uwb-based localization for multi-uav systems and collaborative heterogeneous multi-robot systems. *Procedia Computer Science*, 175, 2020.

[32] Ángel Madridano, Abdulla Al-Kaff, David Martín, and Arturo de la Escalera. Trajectory planning for multi-robot systems: Methods and applications. *Expert Systems with Applications*, 173, 2021.

[33] Jorge Peña Queralta, Carmen Martínez Almansa, Fabrizio Schiano, Dario Floreano, and Tomi Westerlund. Uwb-based system for uav localization in gnss-denied environments: Characterization and dataset. In *IROS*. IEEE, 2020.