Blockchain for Internet of Things

Data Markets, Learning, and Sustainability

Nguyen, Duc Lam

DOI (link to publication from Publisher):
10.54337/aau478977327

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

BLOCKCHAIN FOR INTERNET OF THINGS: DATA MARKETS, LEARNING, AND SUSTAINABILITY

BY
DUC-LAM NGUYEN

DISSERTATION SUBMITTED 2022

AALBORG UNIVERSITY
DENMARK

Blockchain for Internet of Things: Data Markets, Learning, and Sustainability

Ph.D. Dissertation

Duc-Lam Nguyen

Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7B
DK-9220 Aalborg

Dissertation submitted: May 20, 2022

PhD supervisor: Professor Petar Popovski
Department of Electronic Systems
Aalborg University, Denmark

Assistant PhD supervisor: Assistant Professor Israel Leyva-Mayorga
Department of Electronic Systems
Aalborg University, Denmark

PhD committee: Associate Professor Sokol Kosta (chairman)
Aalborg University, Denmark

Professor Salil Kanhere
University of New South Wales, Australia

Professor Olaf Landsiedel
Kiel University, Germany

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Electronic Systems

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-894-6

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Stibo Complete, 2022

Curriculum Vitae

Duc-Lam Nguyen is a Ph.D. Fellow at Aalborg University. He pursued Master's Degree in Computer Science Department at Seoul National University, South Korea, and a Bachelor's in Telecommunication Department at Hanoi University of Science and Technologies, Vietnam in 2019 and 2015, respectively. His research interest includes building distributed systems, Blockchain, data marketplace over Wireless IoT networks, and applying Blockchain and Machine Learning to enhance the efficiency of distributed IoT monitoring Networks.

He is key Blockchain contributor of the European-funded Project namely INTELLIOT where he contributed his knowledge on the integration of Blockchain and Wireless IoT systems and practically implemented Blockchain and Smart Contract for cross-domain applications. He receives *Best Student Paper Award* for the research about scaling Blockchain in Massive IoT at the IEEE World Forum Internet of Things 2020, travel grant for Hyperledger Climate member from Linux Foundation 2020*, Best Paper Award* for a solution of Blockchain-based CO_2 Emission Trading from VEHITS 2021, and *IEEE Honorable Prize* for Blockchain-based IoT Monitoring solution in IEEE ComSoc Student Competition 2020.

Abstract

"Doverey, no Proverey – Trust, but Verify."
- Ronald Reagan

In this decade, the Internet of Things (IoT) has penetrated many aspects of the physical world to realize different applications. Through IoT networks, the applications can collect, exchange, generate, analyze, and aggregate a vast amount of security-critical and privacy-sensitive data. The data collected from IoT devices can have significant economic value for stakeholders. However, the way the data from the standard IoT system is gathered and collected raises concerns about data integrity, trust, security, transparency, and public availability. On the one hand, in IoT deployments, the measured data is stored in a centralized manner or spread across different parties. These data can be both public and private, which makes it difficult to validate their origin and consistency. Besides, querying and performing operations on the data becomes a challenge due to the incompatibility between different application programming interfaces (APIs). Moreover, given the potential value of IoT data, mechanisms must be designed for a reliable and trustworthy data exchange among participants, which are not necessarily trustworthy. Therefore, there is a need for a scalable, distributed, and trusted system for monitoring and exchanging IoT data. Another related recent development is the emergence of Distributed Ledger Technology (DLT). A distributed ledger is a key enabler of trusted and reliable distributed IoT systems, since a DLT supports immutable and transparent information sharing among the involved parties that are not necessarily trusted.

The objective of this research is to utilize DLTs towards designing innovative decentralized solutions for wireless IoT networks, which can guarantee trust, transparency, and privacy of IoT data collection, storage, and trading. In addition, trusted cooperative frameworks are introduced to allow IoT data to be monitored, accounted, and traded among involved participants in order to maximize the utility of IoT data. The proposed innovative solutions include: i) A collaborative framework to design scalable, trusted, and cost-efficient IoT monitoring systems; ii) A framework to evaluate data trading protocols in distributed marketplaces based on communication and computation overhead; iii) The definition and evaluation of two novel use cases, namely, CO_2 emission trading and shared manufacturing using the developed design and evaluation frameworks. The results highlight the relevance of these use cases and outline steps for future implementations.

Resumé

"Doverey, no Proverey – Trust, but Verify."
- Ronald Reagan

I dette arti har Internet of Things (IoT) traengt ind i mange aspekter af den fysiske verden for at realisere forskellige anvendelser. Gennem IoT-netvaerk kan applikationerne indsamle, udveksle, generere, analysere og samle en enorm maengde sikkerhedskritiske og privatlivets folsomme data. Beholdende De data, der er indsamlet fra IoT -enheder, kan have en betydelig okonomisk vaerdi for interessenter. Den made, dataene fra standard IoT -systemet indsamles og indsamler, rejser imidlertid bekymring for dataintegritet, tillid, sikkerhed, gennemsigtighed og offentlig tilgaengelighed. Beholdende Pa den ene side, i IoT-implementeringer, er de malte data enten centraliseret eller spredt over forskellige heterogene parter. Disse data kan vaere bade offentlige og private, hvilket gor det vanskeligt at validere deres oprindelse og konsistens. Desuden bliver foresporgsel og udforelse af operationer pa dataene en udfordring pa grund af inkompatibiliteten mellem forskellige applikationsprogrammeringsgraenseflader (API'er). Pa den anden side, i betragtning af den potentielle vaerdi af IoT -data, skal mekanismer vaere designet til en palidelig og palidelig dataudveksling blandt deltagere, som ikke nodvendigvis er palidelige. Derfor er der behov for et skalerbart, distribueret og betroet system til overvagning og udveksling af IoT -data. En anden relateret nyere udvikling er fremkomsten af distribueret hovedbokteknologi (DLT). En distirbuted hovedbok er en nogleaktivering af betroede og palidelige distribuerede IoT -systemer, da en DLT understotter uforanderlig og gennemsigtig informationsdeling blandt de involverede parter, der ikke nodvendigvis er tillid til.

Formalet med denne forskning er at anvende DLT'er til at designe innovative decentrale losninger til tradlose IoT -netvaerk, som kan garantere tillid, gennemsigtighed og privatlivets fred for IoT -dataindsamling, opbevaring og handel. Derudover introduceres betroede kooperative rammer for at tillade, at IoT -data overvages, regnskabes og handles blandt involverede deltagere i raekkefolge maksimerer anvendeligheden af IoT data. De foreslaede innovative losninger inkluderer: Beholdende i) en samarbejdsramme for at designe skalerbare, palidelige og omkostningseffektive IoT-overvagningssystemer; Beholdende ii) en ramme til evaluering af datahandelsprotokoller pa distribuerede markedspladser baseret pa kommunikations- og beregningsomkostninger; Beholdende iii) Definitionen og evalueringen af to nye brugssager, nemlig CO2 emissionshandel og delt fremstilling ved hjaelp af de udviklede design og evalueringsrammer. Resultaterne fremhaever relevansen af disse brugssager og skitserer trin for fremtidige implementeringer.

Acknowledgements

I would like to express my deepest gratitude for my supervisor, Professor Petar Popovski, for his constant support, guidance, and encouragement throughout the course of my doctoral study and research. I have always been inspired by his in-depth interdisciplinary knowledge as well as by his vision and aspiration for high-quality research. From Petar, I have learned how to always keep the bar high and to strive for the best possible achievement. I am honored to have been working under his supervision and I simply cannot thank him enough for his constant advice on many aspects of a Ph.D. student's life and research career. I am grateful for his support and fight for the right of his PhD students. On the other side, I am ready to fight to die for him.

Special thanks goes to Israel Leyva-Mayorga for unlimited help and guidance during my PhD research. Starting PhD is never easy, and I feel lucky enough to have him as a friend, a supervisor at the beginning of my PhD.

I would like to extend my gratitude to Prof. Sokol Kosta, Prof. Salil Kanhere, and Prof. Olaf Landsiedel for serving as committee members in my dissertation defense.

I would like to thank all the Connectivity members, especially Beatriz, Shashi, and collaborators for the discussions and encouragement. Warm thanks to Broering Arne for help me a lot during the time I stay in Siemens, Munich. My sincerest thanks to all the coauthors of my publications. I would like to also thank all my wonderful friends at Aalborg University, especially C1-103 (then changed to C1-110) team, with Anders, Radek, and Igor for for making my Ph.D student life enjoyable every minutes.

I am grateful for the financial support from European Research Council (Horizon 2020 ERC Consolidator Grant Nr.648382) Willow and IntellIoT (grant agreement No. 957218) which have provided me with the necessary resources to carry on research and finish this dissertation.

Last, but not least, I would like to thank my parents, Mr. Nguyen Van Thanh and Mrs. Vu Thi Lien, my wife Thu-Hang Tran, and my parent-in-law Mr. Tran Van Huyen, and Mrs. Nguyen Thi Toan, for their continued understanding and encouragement. Their unconditional love and support have given me the strength to chase my dreams and aspirations. To them, I dedicate this dissertation.

Duc Lam Nguyen
Aalborg University, May 20, 2022

Contents

Thesis Details

Thesis Title:	Blockchain for Internet of Things: Data Markets, Learning, and Sustainability
Ph.D. Student:	DUC-LAM NGUYEN
Supervisors:	Professor Petar Popovski, Aalborg University
Co-supervisor:	Assistant Professor Israel Leyva-Mayorga, Aalborg University

The main body of this thesis consist of the following papers.

[A] **Duc Lam Nguyen**, Israel Leyva-Mayorga, and Petar Popovski. "Witness-based Approach for Scaling Distributed Ledgers to Massive IoT Scenarios." In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, p. 9221269. IEEE, 2020. ⋆Best Student Paper Award⋆

[B] **Duc Lam Nguyen**, Anders E. Kalor, Israel Leyva-Mayorga, and Petar Popovski. "Trusted wireless monitoring based on distributed ledgers over NB-IoT connectivity." *IEEE Communications Magazine 58*, no. 6 (2020): 77-83.

[C] **Duc Lam Nguyen**, Israel Leyva-Mayorga, Amari N. Lewis, and Petar Popovski. "Modeling and analysis of data trading on blockchain-based market in iot networks." *IEEE Internet of Things Journal*, no. 8 (2021): 6487-6497.

[D] **Duc Lam Nguyen**, Shashi Raj Pandey, Soret Beatriz, Arne Broering, and Petar Popovski. "A Marketplace for Trading AI Models based on Blockchain and Incentives for IoT Data." *IEEE Internet of Things Journal (Submitted) (2022)*.

[E] **Duc Lam Nguyen**; Lewis, A.; Leyva-Mayorga, I.; Regan, A. and Popovski, P. (2021). B-ETS: A Trusted Blockchain-based Emissions Trading System for Vehicle-to-Vehicle Networks. In Proceedings of the *7th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, ISBN 978-989-758-513-5; ISSN 2184-495X, pages 171-179. DOI: 10.5220/0010460501710179. ⋆Best Paper Award⋆

[F] **Duc Lam Nguyen**, Broering Arne, Massimo Pizzol, and Petar Popovski, "Analysis of Distributed Ledger Technologies in Industrial Manufacturing", submitted to *Nature Scientific Reports* (2022).

In addition to the main papers, the following publications have also been made.

[1] Soret, Beatriz, **Duc Lam Nguyen**, Jan Seeger, Arne Bröring, Chaouki Ben Issaid, Sumudu Samarakoon, Anis El Gabli, Vivek Kulkarni, Mehdi Bennis, and Petar Popovski. "Learning, Computing, and Trustworthiness in Intelligent IoT Environments: Performance-Energy Tradeoffs." *IEEE Transactions on Green Communications and Networking (2021)*.

[2] Danzi, Pietro, Anders E. Kalor, Rene B. Sorensen, Alexander K. Hagelskjær, **Duc Lam Nguyen**, Cedomir Stefanovic, and Petar Popovski. "Communication aspects of the integration of wireless IoT devices with distributed ledger technology." *IEEE Network* 34, no. 1 (2020): 47-53.

[3] Carrillo, Dick, **Duc Lam Nguyen**, Pedro HJ Nardelli, Evangelos Pournaras, Plinio Pelegrini Morita, Demóstenes Zegarra Rodríguez, Merim Dzaferagic et al. "Corrigendum: Containing Future Epidemics with Trustworthy Federated Systems for Ubiquitous Warning and Response." *Frontiers in Communications and Networks* (2021): 35.

[4] Pandey, Shashi Raj, **Duc Lam Nguyen**, and Petar Popovski. "A Contribution-based Device Selection Scheme in Federated Learning." submitted to *IEEE Communication Letters* for publication (2022).

List of Figures and Tables

List of Figures

List of Tables

Introduction

1 Introduction

1.1 Overview

Recent development in the Information and Communication Technology (ICT) has leveraged the evolution of traditional computer-based systems towards smart infrastructures [1]. During this evolution, the emergence of the Internet of Things (IoT) has played a vital role of connecting together various areas e.g, Artificial Intelligent (AI), Big Data, and communication to achieve technological advancements for multiple benefits. The applications of the IoT are spread across a wide diversity of industrial applications such as smart agriculture, healthcare, and industrial manufacturing [2]. IoT has deeply changed the traditional ways of conducting several human activities through extensively networked automation, system monitoring, and control. However, the widespread application of IoT raised numerous challenges including resource-constrained IoT devices, security, privacy, vulnerabilities, energy sustainability, and heterogeneity of IoT networks [3]. In addition, sharing IoT data in the wireless IoT environment is usually considered unsafe and vulnerable to cyber-threats.

The revolution of Distributed Ledger Technologies (DLTs) has brought new promising solutions to address the aforementioned drawbacks of state-of-the-art IoT technologies. DLT is a database managed by multiple participants across the network [4]. Blockchain is one type of distributed ledger which allows recording, synchronizing, and sharing of formatted transactions in their electronic ledgers instead of keeping data centralized in a conventional database. Besides, there are other types of DLTs such as Directed Acyclic Graph (DAG) [5], and Hashgraph [6]. In the scope of this research, we focus on the Blockchain type which is essentially a distributed ledger spreading over the whole distributed network. The data from clients are formatted in Blockchain-type transactions and grouped in blocks. These blocks have certain storage capacities and, when filled, are closed and linked to the previously filled block, forming a chain of data known as the Blockchain[1]. All new information that follows that freshly added block is compiled into a newly formed block that will then also be added to the chain once filled [7]. Instead of recording the data centralized as a standard database, Blockchain with its distributed consensus mechanism allows the system to achieve a common agreement among involved participants across the network. Outside of its role in financial transactions, DLTs are seen as a key enabler for trusted and reliable distributed monitoring systems. The

[1]The terms DLT and Blockchain will be used interchangeably throughout this article, Blockchains are a type of DLT, where chains of blocks are made up of digital pieces of information called transactions and every node maintains a copy of the ledger

authentication process for DLTs relies on consensus among multiple nodes in the network [8]. In Blockchain-enabled IoT networks [9], transactions can include sensing data, or monitoring control messages, and these are recorded and synchronized in a distributed manner among all the participants of the system. Blockchain enables a *decentralized* validation of the transactions, avoiding the storage and processing bottlenecks of centralized systems [10]. In addition, Blockchain-based transactions are recorded *immutably* in the distributed ledger since every node in the network has a copy of transactions and is synchronized together. Finally, the clients can easily query information of transactions from the distributed ledger, e.g, data source, timestamp [11], etc. Therefore, Blockchain is considered a *transparency* system and can be applied in various domains to guarantee the trust of information for participants. With these natural features, Blockchain is an essential complement to the IoT with the enhancements of trust, security, privacy, and heterogeneity [8].

1.2 Problem Statement

Integrating Blockchain with IoT systems faces several challenges, including scalability, interoperability, resource-constrained IoT devices, data privacy, and so on. Among these aspects, the scalability of storage resources is one of the most important to address, as the size of a typical DLT/Blockchain is usually very large and impossible to implement on resource-limited devices [12], for example, the current size of Bitcoin Blockchain is *324 gigabytes* [13], size of Ethereum Blockchain is *991.56 GB* [14]. It means that there is a need for an enormous amount of data synchronized among nodes.

These issues have been investigated and three categories of solutions for the communication between the Blockchain and IoT nodes have been defined [15]. In type I, the IoT devices act as Blockchain full nodes which can host the whole distributed ledger and also do mining tasks as usual. All the information e.g, blocks, and data are synchronized among Blockchain full nodes. Type I is mostly impossible in real-life implementation because of tiny IoT devices. In type II, IoT devices work as light clients which function as wallets but do not store the entire distributed ledger. These light clients must connect to Blockchain full nodes to broadcast their transactions to the network [16]. Only a part of blocks, e.g, block headers, is exchanged between light clients and full nodes. In type III, the IoT devices transfer data to a proxy which is responsible for formatting transactions, signing, and forwarding transactions to Blockchain full nodes. There is a trade-off while choosing the communication protocol for specific applications depending on the device capacities. These three communication protocols generally cover the communication between IoT devices and Blockchain in a wireless environment. But in massive IoT scenarios, there is an open challenge for an efficient communication architecture to combine Blockchain with IoT. Additionally, most IoT devices are miniature and very limited when it comes to the computing resources necessary for secure capabilities. It is difficult to know who owns or possesses them, if they have been hacked, and if they are acting in undesired ways. This makes IoT devices not very trustworthy.

Besides, the exploitation of IoT data usage also needs to be addressed. IoT data is an important asset in the digital economy and is driving the rise of data markets. Meanwhile, data markets promote data trading efficiently and improve the utilization of data. Conventional trading systems (e.g. Paypal) feature a single point of failure, the lack of trust, transparency, and incentive for data trading, which is preventing the availability of digital information from data providers to customers [17]. On the other hand, Distributed ledger technologies (DLTs) and

Blockchains support immutable and transparent information sharing among involved untrusted parties [7]. In addition, DLTs allow the storage of all transactions into immutable records and every record is distributed across many participants. Thus, security in DLTs comes from the decentralized operation, but also from the use of strong public-key cryptography and cryptographic hashes. The benefits of the integration of DLTs into IoT data trading systems include: i) guarantee of immutability and transparency for environmental sensing data, ii) removal of the need for third parties, iii) development of a transparent system for heterogeneous IoT data trading networks to prevent tampering and injection of fake data from the stakeholders [8].

Finally, the potential and applications of DLTs in wireless IoT environment are very limited due to lack of adoption, and high cost. There is a need for realistic use cases which are rather than track-and-trace applications.

1.3 Research Objectives and Contributions

The goal of this research is to design disruptive innovations in terms of data monitoring, data trading, and use cases for the DLT-based IoT networks.

Objective 1: Building a scalable Blockchain-based framework which can reduce the traffic and execution cost of main chain for massive IoT Networks.

Solutions and results. In this research, we introduce a new lightweight distributed ledger mechanism called *wiBlock*. This scheme aims to solve the scalability problems and execution cost of Blockchain in a massive IoT environment by defining the new concepts of global transactions and local transactions. The *wiBlock* addresses scalability, trust and implementation cost issues of implementing Blockchain in IoT by: i) enabling the use of DLTs for recording IoT data, ii) limiting the number of transactions that must be processed at the main distributed ledger (or called main chain), and iii) eliminating the need for complex computations and supporting sleep-awake mechanisms at the IoT devices, respectively. The key innovation of *wiBlock* is a *witness* system that can process transactions locally and communicate directly with the main ledger. The local transactions can be used internally in Blockchain-based IoT networks and record transactions locally in witness, and global transactions can be used in heterogeneous networks and recorded in the main chain. Therefore, the *witness* system reduces the number of transactions that need to be processed by the main chain. The results show that our proposed scheme improves the scalability of integrated blockchain and IoT monitoring systems by processing a fraction of the transactions, inversely proportional to the number of witnesses, locally. Hence, reducing the number of global transactions processed in the main chain. The detail of this research is presented in **paper A** [9].

In addition, we aim to investigate the feasible of integration of Blockchain in large-scale monitoring networks. In this research, we choose Narrowband Internet of Things (NB-IoT) [18] as connectivity method over LoRaWAN [19], and Sigfox [20]. Specifically, we first present a Blockchain-powered IoT framework for environmental monitoring systems that address the problem of trust and privacy. Second, we evaluate the proposed framework via extensive experiments, in which the NB-IoT monitoring system and a suitable DLT platform are integrated. Third, realizing the lack of studies on communication aspects of current Blockchain-enabled IoT systems, we analyze and evaluate the interaction between Blockchain and the NB-IoT monitoring systems in terms of overall throughput, E2E latency, and communication overhead via

two case studies. Regarding previous studies [21], to our best knowledge, these studies mainly focus on specific applications of Blockchain-enabled IoT, and how to integrate Blockchain with IoT. In our studies, communication aspects between Blockchain nodes and IoT devices are investigated. We studied the flexible of NB-IoT uplink and downlink traffic is suitable solution to integrate with Blockchain synchronization protocols. We successfully built a proof-of-concept for the integration of Hyperledger Fabric with NB-IoT to monitor the CO_2 level in the air. The detail of the system design and proof are shown in **paper B** [8].

Objective 2: Building an IoT data marketplace for trading in wireless IoT environment which focuses on the communication efficiency among stakeholders and provides data privacy and data valuation capabilities.

Solutions and results. Based on the research in **objective 1**, we realized that the IoT data collected from sensors could be recorded to the distributed ledger for analyzing and accounting purposes. The clients can query the recorded data in the distributed ledger based on defined Application Programming interfaces (APIs) and transaction ID. However, the IoT data is valuable and can be exchanged between different stakeholders. In this research, we designed an IoT data marketplace based on the autonomous execution of the smart contract on top of DLT infrastructure which allows customers or IoT devices can buy or sell the IoT data via Smart Contracts. Specifically, we first present a solution for a systematic DLT-based IoT data trading toward a city-level network using NB-IoT connectivity. Then, we introduced the concepts of three trading protocols based on the interaction between buyers, sellers, and smart contracts. These protocols are *General Trading (GT)* Protocol , *Buying on Demand (BoD)* protocol, and *Selling on Demand (SOD)* protocol. Each trading protocol can be implemented and used depending on the specific scenarios. For example, GT could be used as the standard trading scheme on data marketplace, and trading platform, while BoD and SoD are implemented based on the demands from either buyers or sellers. Via this research, we provide a benchmark for IoT data trading protocols based on smart contracts, which not only analyzes the communication aspects among participants, but also the insightful terminology of an IoT data marketplace. The detail of the design and results are presented in the **paper C** [22].

However, a significant problem with most the data marketplaces is data privacy issue [23]. Trading IoT data over a wireless IoT network means that the data is exchanged between involved buyers and sellers. Therefore, the IoT data is vulnerable for attacks and hijacks. To address this issue, we leverage Federated Learning into data marketplace. The emergence of Federated Learning (FL), which acts as a special machine learning technique for privacy-preserving, offers to contextualize data in IoT networks. With FL, instead of sharing IoT data over wireless networks, the local trained ML is exchanged among participants. In specific, we introduce a DLT-based model trading system which enables a secured and trusted marketplace to collaboratively train ML models as well as guarantees fair incentives for every participant and privacy of data. Based on the quality of the uploaded models, which is quantified by using a distributed Data Shapley Value (DSV), the participants can get the incentive based on the updated models, for example, as tokens or fiats. Note that based on our proposed system, the parties do not need to share their local data, but only provide customized models or query interfaces to the marketplace. Consequently, the proposed system allows multiple participants to jointly train the ML models in the marketplace based on their own training data. Buyers who need to train their ML model will pay the market for the improvement of their model, and sellers who sell their

contribution to train the ML models will get paid by the market via smart contracts. We design a communication architecture for trading ML model over Wireless IoT network, and propose an incentive mechanism for evaluating the valuation of IoT data based on new concept of Distributed Shapley Value. The detail of the design ML marketplace and results are demonstrated in **paper D** [24].

Objective 3: Designing novel use cases and applications based on proposed framework named: i) CO_2 emission trading, and ii) manufacturing sharing which towards to address sustainability issue.

Solutions and results. Motivated by the designed framework, we investigated two realistic use cases namely CO_2 Emission trading and manufacturing sharing in Industrial. In the first use case, we first tackle the challenges of the current European Emission Trading (EU-ETS) system by proposing a distributed emissions allowance trading system called Blockchain-based Emission Trading System (B-ETS). The system creates an account for the emissions generated from each vehicle and allows exchanges among vehicles in a trusted manner based on Blockchain and Smart Contracts. In B-ETS, each vehicle acts as a light client in the global Blockchain network and manages its own Emission Allowance Balance (EAB) which is reset at the beginning of each day. The EAB data is recorded transparently and immutably in the distributed ledger. It should be noted that we use one day as our unit of time without loss of generality. Any other unit (a week, a month) could be used if that seemed more suitable. Then, we introduce an economic incentive-based mechanism that attracts drivers to change their driving behavior in order to reduce emissions. Each vehicle's generated emissions are calculated and the data are recorded immutably in the distributed ledger. If the emission level is higher than the defined threshold, the EAB will be reduced. If the EAB goes to zero, the driver needs to buy credits in the form of EAB from others. The detail of system design and analysis are described in the **paper E** [25].

Second, we designed a proof-of-concept for a shared manufacturing application. Specifically, the framework allows users to rent robots and machines from plant companies via smart contracts and make the payments on the top of trusted DLT infrastructure. The contribution of this research are presented as follow. We first analyze deeply different 5 DLT platforms named Ethereum, IoTA, Hyperledger Fabric, Quorum, and Solana in various aspects e.g, latency, communication and computation overhead, and their capabilities when used in industrial manufacturing environments. Then, we proposed an industrial ystem design for DLT-based IIoT manufacturing systems that can integrate and adapt multiple features and components for facilities sharing services, and evaluation of communication and computation overhead of different DLTs in resource-constrained IoT networks. This benchmark of different DLTs for manufacturing scenarios will help interested parties to understand the trade-offs in DLT-based systems. The proof-of-concept system is demonstrated at Siemens AG. The detail is presented in **paper F**.

2 Background and State-of-the-art

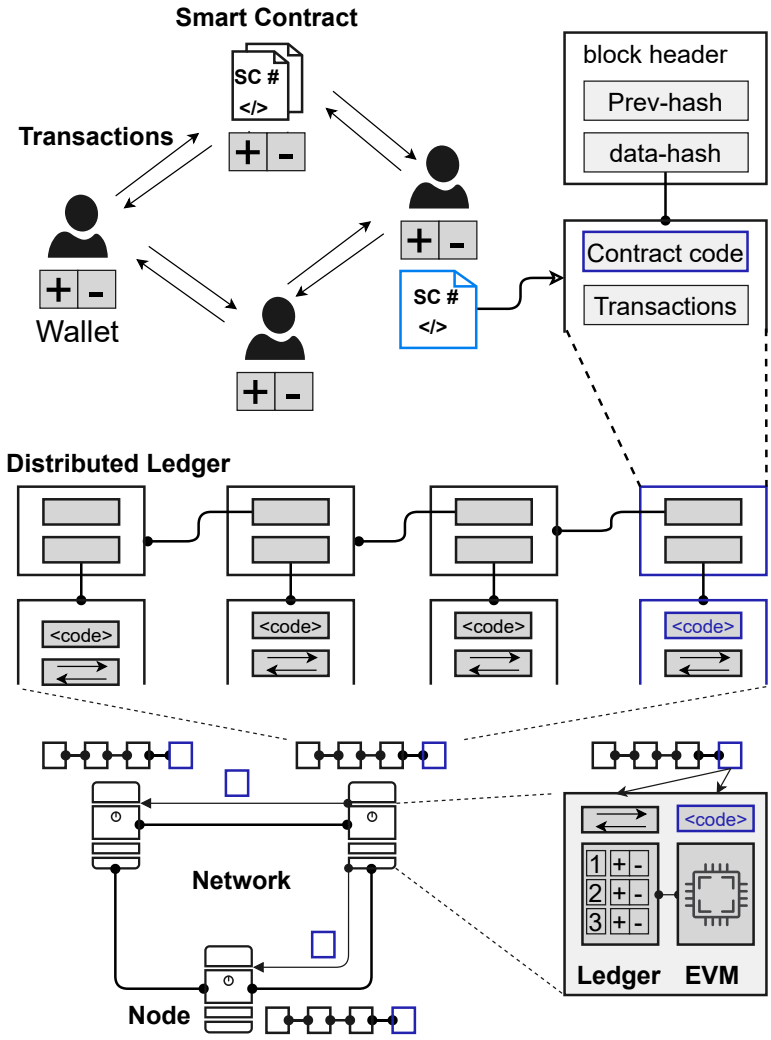


Fig. 1: Blockchain-enabled smart contract architecture.

2.1 Distributed Ledger Technologies over Wireless IoT

In recent years, DLT has been the focus of large research efforts spanning several application domains. Starting with the adoption of Bitcoin and Blockchain, DLT has received a lot of attention in the realm of IoT, as the technology promises to help address some of the IoT heterogeneity, security and scalability challenges [26]. For instance, in IoT deployments, the recorded data are either centralized or spread out across different heterogeneous parties. These data can be both public or private, which makes it difficult to validate their origin and consistency. In addition, querying and performing operations on the data becomes a challenge due to the incompatibility between different Application Programming Interfaces (APIs). For ex-

ample, Non-Governmental Organizations (NGOs), Public and Private sectors, and industrial companies may use different data types and databases, which leads to difficulties when sharing the data [27].

A DLT system offers a tamper-proof ledger that is distributed on a collection of communicating nodes, all sharing the same initial block of information, the genesis block [28]. In order to publish data to the ledger, a node includes data formatted in transactions in a block with a pointer to its previous block, which creates a chain of blocks, the so called Blockchain. A block generated by a node usually needs to solve a mathematical crypto-puzzle and gives the solution as a proof of its workload to get a reward [29]. This process is called mining. The difficulty of the crypto-puzzle is adjusted based on the total computational power or mining power of the network. Each correctly behaving miner needs to adhere to the same protocol for creating and also validating new blocks. After successfully mining a block, a miner broadcasts it for validation. Each transaction recorded in distributed ledger is essentially immutable since each DLT node in the network keeps all the committed transactions in the ledger. In addition, cryptography mechanisms e.g, hash functions, asymmetric encryption algorithms, and digital signature, guarantee the integrity of data blocks in the DLTs. Therefore, the DLTs can ensure non-repudiation of transactions. Moreover, each transaction is historically timestamped and identified, so that it is traceable to every user. The overall system architecture of Blockchain is shown in Fig. 1. With the aforementioned characteristics, the advantages of the integration of DLTs into wireless IoT networks consist of: i) guarantee of immutability and transparency for recorded IoT data; ii) removal of the need for third parties; iii) development of a transparent system for heterogeneous IoT networks to prevent tampering and injection of fake data from the stakeholders. Specifically, in healthcare systems, DLTs can potentially address the challenges of privacy-preserving and security of sensitive healthcare records. For instance, the authors in [30] demonstrated that applying DLTs can protect healthcare records which are stored in distributed cloud servers. Besides, the medical sensors can automatically gather healthcare data and transmit to the ledger via smart contract executions, which supports the instant access and security of patient monitoring [31].

2.2 Smart Contract

The term "Smart Contract" was originally invented to refer to the automation of legal contracts by Nick Szabo [32]. The advent of DLT has recently brought much interest on smart contracts and its applications. Recently, smart contract [33] is referred as a distributed app that lives in the DLTs. This app is, in essence, a programming language class with fields and methods, and they are executed in a transparent manner on all nodes participating in a Blockchain [34]. Smart contracts are the main DLT-powered mechanism that is likely to gain a wide acceptance in IoT, where they can encode transaction logic and policies, which includes the requirements and obligations of parties requesting access, the IoT resource/service provider, as well as data trading over wireless IoT networks [35].

DLTs can be categorized into 2 types namely permissioned and permissionless. The permissionless DLT platforms allow any user to join the network while permissioned DLT platforms allow only permitted users to join [21]. Different DLT platforms provide different support for smart contracts. For instance, some DLTs, e.g, Bitcoin, may only allow users to use a simple scripting language to develop smart contracts with simple logic, while others, e.g, Ethereum and Hyperledger fabric support much more advanced programming languages for writing smart con-

tracts. The codes of Ethereum smart contract are written in a stack-based byte-code language and executed in a virtual machine called Ethereum Virtual Machine (EVM) [33]. Ethereum Smart Contract is currently the most popular platform for developing distributed application running on top of Blockchain.

Smart Contracts have been applied in various IoT areas such as healthcare [30, 31], supply chain [36], smart manufacturing [37], and vehicular networks [38]. For instance, in the smart manufacturing area, the work described in [37] investigates DLT-based security and trust mechanisms and elaborates a particular application of DLTs for quality assurance, which is one of the strategic priorities of smart manufacturing. Data generated in a smart manufacturing process can be leveraged to retrieve material provenance, facilitate equipment management, increase transaction efficiency, and create a flexible pricing mechanism. In this research, we exploit the capabilities of smart contract to enable an distributed shared manufacturing application to allow people to rent facilities and make micro-payments based on completed tasks.

One of the challenges of implementing DLT-based smart contracts in IoT and edge computing is the limited computation and communication capabilities of some of the nodes. In this regard, the authors in [35, 39] worked on the communication aspects of integrating DLTs with IoT systems. The authors studied the trade-off between the wireless communication and the trustworthiness with two wireless technologies, LoRa and NB-IoT. The authors in [12] introduced a system called TinyEVM to generate and execute off-chain smart contracts based on sensor data in low-level devices to perform micro-payments and address device constraints.

2.3 Federated Learning

Implementing intelligent IoT systems with distributed Machine Learning (ML)/Artificial Intelligence (AI) over wireless networks (e.g., NB-IoT) needs to consider the impact of the communication network (latency and reliability under communication overhead and channel dynamics) and on-device constraints (access to data, energy, memory, compute, and privacy, etc.). Obtaining high-quality trained models without sharing raw data is of utmost importance, and redounds to the trustworthiness of the system. In this view, Federated Learning (FL) has received a groundswell interest in both academia and industry, whose underlying principle is to train a ML model by exchanging model parameters (e.g., Neural Network (NN) weights and/or gradients) among edge devices under the orchestration of a federation server and without revealing raw data [40]. Therein, devices periodically upload their model parameters after their local training to a parameter server, which in return does model averaging and broadcasting the resultant global model to all devices.

FL has been proposed by Google for its predictive keyboards [41] and later on adopted in different use cases in the areas of intelligent transportation, healthcare and industrial automation, and many others [42, 43]. While FL is designed for training over homogeneous agents with a common objective, recent studies have extended the focus towards personalization (i.e., multi-task learning) [44], training over dynamic topologies [45] and robustness guarantees [46, 47]. In terms of improving data privacy against malicious attackers, various privacy-preserving methods including injecting fine-tuned noise into model parameters via a differential privacy mechanism [48–51] and mixing model parameters over the air via analog transmissions [52, 53] have been recently investigated.

3 Dissertation Organization

The rest of this thesis is organized as follows:

Paper A details our witness-based solution for scalable Blockchain-based IoT networks. We introduce a lightweight distributed ledger scheme to integrate Proof-of-Work blockchain into IoT. In our scheme, we classify transactions into two types: 1) global transactions, which must be processed by global blockchain nodes and 2) local transactions, which can be processed locally by entities called *witnesses*. Performance evaluation demonstrates that our proposed scheme improves the scalability of integrated blockchain and IoT monitoring systems by processing a fraction of the transactions, inversely proportional to the number of witnesses, locally. Hence, reducing the number of global transactions.

Paper B presents we present a blockchain-powered IoT framework for environmental monitoring systems that addresses the problem of trust and privacy. Second, we evaluate the proposed framework via extensive experiments, in which the NB-IoT monitoring system and a suitable DLT platform are integrated. Third, realizing the lack of studies on communication aspects of current blockchain-enabled IoT systems, we analyze and evaluate the interaction between blockchain and the NB-IoT monitoring systems in terms of overall throughput, E2E latency, and communication overhead via two case studies. Regarding previous studies [21], to the best of our knowledge, these studies mainly focus on specific applications of blockchain-enabled IoT and how to integrate blockchain with IoT. In our studies, communication aspects between blockchain nodes and IoT devices are investigated.

Paper C describes a solution for a systematic DLT-based IoT data smart trading toward city-level networks using NB-IoT connectivity. Next, we propose three IoT data trading protocols namely GT, BoD, and SoD. The cost model of each trading protocol is derived and analyzed along with NB-IoT connectivity. Both resources consumed by executing DLT/smart contracts and NB-IoT devices are investigated. Finally, the analysis and the associated experimental results provide a benchmark for data trading protocols in wide-area IoT networks.

Paper D shows a new ecosystem of ML model trading over a trusted Blockchain-based network is proposed. The buyer can acquire the model of interest from the ML market, and interested sellers spend local computations on their data to enhance that model's quality. In doing so, the proportional relation between the local data and the quality of trained models is considered, and the valuations of seller's data in training the models are estimated through the distributed Data Shapley Value (DSV). At the same time, the trustworthiness of the entire trading process is provided by the distributed Ledger Technology (DLT). Extensive experimental evaluation of the proposed approach shows a competitive run-time performance, with a 15% drop in the cost of execution, and fairness in terms of incentives for the participants.

Paper E summarizes provides a state-of-the-art overview of these technologies and illustrates their functionality and performance, with special attention to the tradeoff among resources, latency, privacy and energy consumption. Finally, the paper provides a vision for integrating these enabling technologies in energy-efficient iIoTe and a roadmap to address the open research challenges

Paper F propose a new distributed Blockchain-based emissions allowance trading system called B-ETS. This system enables transparent and trustworthy data exchange as well as trading of allowances among vehicles, relying on vehicle-to-vehicle communication. In addition, we introduce an economic incentive-based mechanism that appeals to individual drivers and leads them to modify their driving behavior in order to reduce emissions. The efficiency of

the proposed system is studied through extensive simulations, showing how increased vehicle connectivity can lead to a reduction of the emissions generated from those vehicles. We demonstrate that our method can be used for full life-cycle monitoring and fuel economy reporting. This leads us to conjecture that the proposed system could lead to important behavioral changes among the drivers

Paper G presents potential DLT technologies for an efficient and intelligent integration of DLT-based solutions in manufacturing environments. We propose a general framework to adapt DLT in manufacturing, then we introduce the use case of *shared manufacturing*, which we utilize to study the communication and computation efficiency of selected DLTs in resource-constrained wireless IoT networks.

4 Discussion and Future Work

This section discusses the main contributions of this dissertation and introduces future research directions that are worth investigating and can leverage the frameworks proposed in this research.

4.1 Discussion

Although the convergence of DLTs and wireless IoT has the potential to revolutionize several industrial sectors, there are many challenges to be addressed before the potential of DLTs in this context can be fully unleashed. During this research, we investigate the main challenges for the integration of DLT and IoT technologies, and propose potential solutions that cover the design of a scalable DLT-based IoT system to reduce the execution cost of the distributed ledger and the design of data marketplaces for trading IoT data as well as ML models.

Engineering Design of DLT-based Wireless IoT

In order to design efficient DLT-based IoT systems, in the scope of this research, we discuss three important factors: i) scalability, ii) choosing an appropriate DLT platform, and iii) communication architecture.

Scalability. The scalability problem of current DLTs limits the wide usage in large-scale networks. The scalability issue addressed in **paper A** can be analyzed in the aspects of throughput, latency, and communication. First, Bitcoin [7], and Ethereum [33] is considered as the most popular payment application based on Blockchain, however, the throughput of Bitcoin is restricted to approximately 7 transaction per second (tps), and 25 tps. Meanwhile, in an IoT network, there could be an enormous number of devices that generate millions of requests per day. These DLTs can not handle the high number of requests from IoT clients. The throughput of DLT platforms is dependent on block interval time and the number of transactions in each block. There are currently various solutions to address the scalability issue of DLTs such as increasing block size, reducing transaction size, sharding [54], and off-chain transactions [55]. Compared to these solutions, *wiBlock* solved the problem by clarifying transaction types namely global transactions and local transactions. The advantages of *wiBlock* are that i) it can be deployed and integrated with any DLT-based systems to reduce the number of transactions exponentially processed in the main chain, and ii) *wiBlock* also addresses the heterogeneity issue of the IoT.

However, the downside is that i) *wiBlock* is designed based on PoW Blockchains, which consume energy for mining tasks, ii) the clarification of DLT transactions is still simple with global and local transactions, and the number of transactions processed at global transaction is increased with the number of witnesses. The chosen number of witness nodes is not taken into account in various aspects. We observe that there is no one-fit-all solution to address the scalability issue so that *wiBlock* can be integrated with other solutions e.g, off-chain transactions and sharding to enhance the performance.

Which DLT should be chosen for your application? Depending on the type of IoT application, we can choose an appropriate DLT platform to fit with the target system. This step can help to reduce the difficulty in the practical implementation of the systems and enhance the performance of the overall systems. In **paper B** and **paper F**, we discussed the pros and cons of different DLTs, e.g, Bitcoin, Ethereum, Solana [56], IOTA [57], Hyperledger Fabric [58], and Quorum [59]. PoW Blockchains could not be used for IoT applications because of their low speed, and consumption of more energy than other types of consensus algorithms [60]. In **paper B**, the results show a significant contribution in building a DLT-based large-scale NB-IoT. One of the findings is that the flexibility in the uplink and downlink of NB-IoT could be a key factor to implement DLTs in LPWAN networks. In the scope of this research, we chose Hyperledger Fabric to build a proof-of-concept solution, and it is a private DLT designed for the purposes of monitoring and accounting emission data.

Communication Architecture. Communication architecture is one of the important factors that needs to be taken into account. In DLT-based IoT networks, lightweight designs [61] [62] for Blockchain clients are used widely in various applications e.g smart homes, and industrial manufacturing, to implement Blockchain in current standard IoT systems. Since IoT devices have different installed operating systems and configurations, it is difficult to establish a single architecture that can be universally applied. IoT devices with limited memory and computational resources pose an even greater concern as they often lack the resources to host a communication protocol. In this thesis, we have introduced various communication architectures for different applications, e.g, data trading **paper C**, ML trading in **paper D**, emission trading in **paper E**, and manufacturing sharing in **paper F**. Each communication architecture has different requirements on device capacities, latency, and throughput depending on the specific application.

Blockchain is not only about storing data, it is about trusted sharing.

Blockchain is usually known as a distributed database where data can be recorded transparently and immutably. Blockchain guarantees the trust of the data origin and source of the data [63]. In a standard IoT system, it is hard to allow resource-constrained devices to make payments due to lack of security and payment channels for tiny amounts. In **paper E**, we have built an IoT data market with a benchmark based on Blockchain. The proposed marketplace has the advantages of: i) providing a benchmark with 3 different IoT data trading protocols for different scenarios and ii) allowing IoT devices to trade data together via smart contracts running on top of DLT infrastructure. This work opens an interesting possibility of trading IoT data over wireless IoT networks.

However, the IoT data marketplace has a significant challenge with the valuation of IoT data. During the trade one can only guarantee data provenance, but there is no estimate of the the value or quality of the IoT data. We have leverage Federated Learning to deep dive

into the data and use local weight updates to exchange information, while respecting privacy constraints. Our proposed scheme in **paper F** introduced with the distributed Shapley Value concept can address the aforementioned issue. Therefore, we argue that Blockchain can be used for trading and sharing data over wireless IoT networks and is suitable to apply to many applications, such as CO_2 emission trading.

The Issue of Sustainability

There is an argument that Blockchain is a source of emission because of its mining tasks. However, from our point of view, we exploit Blockchain as a trusted infrastructure for monitoring CO_2 emission level of a specific part of a large-scale area covered by NB-IoT, presented in **paper B**, and for trading CO_2 emission allowance among vehicles in **paper E**. Besides the power-hungry blockchains, such as PoW Bitcoin, and Ethereum [64], there are hundreds of DLT platforms [65] currently on the market with various consensus strategies that do not impose mining tasks. Depending on the specific application, an appropriate DLT platform can be chosen for deployment. For demonstration, we have introduced a use case of industrial manufacturing sharing where robots and machines on the field can be rented from customers via predefined smart contracts. The results regarding communication and computation overhead are our analysis of the performance of different DLT platforms. Furthermore, we have also investigated the problem of CO_2 generated from the process. This research could be a benchmark for choosing and implementing a DLT in industrial IoT applications.

4.2 Future Work

As discussed in the section above, our proposals for integration of DLTs in wireless IoT for monitoring, accounting, and trading have advantages and disadvantages.

First, the *wiBlock* system has the potential to improve with the detailed analysis of a number of witnesses in the system and the type of DLT transactions. DLT transactions can be classified in terms of specific functions, type of data, etc. In addition, integrating *wiBlock* with other schemes, e.g, off-chain transactions, could significantly reduce the number of transactions processed by the main chain. The off-chain transactions are processed offline among involved participants, and the nodes just upload the final report to the main chain.

Second, the IoT data marketplace based on DLTs with Federated Learning brings a promising solution for trading IoT data. We plan to investigate the communication efficiency of the data market over multiple wireless interfaces, e.g, LoraWAN, NB-IoT, Sigfox, Wifi, and Zigbee. Each communication interface has different requirements and protocols for communication. This has a direct impact on the Blockchain synchronization protocols, block time, and overall system. The distributed Shapley Value proposed in the **paper D** raises the issue of the training time that is required to carry out valuation of the data. In a big data market, the time required for training and for synchronize blocks in the DLT network is an important contributor to the overall latency.

Finally, the new applications of Blockchain and smart contracts may give rise to new communication challenges for the Blockchain-based IoT networks and these challenges need to be addressed under the constraints put forward by the requirement for sustainable operation.

References

[1] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[2] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[3] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of iot: Applications, challenges, and opportunities with china perspective," *IEEE Internet of Things journal*, vol. 1, no. 4, pp. 349–359, 2014.

[4] M. Rauchs, A. Glidden, B. Gordon, G. C. Pieters, M. Recanatini, F. Rostand, K. Vagneur, and B. Z. Zhang, "Distributed ledger technology systems: A conceptual framework," *Available at SSRN 3230013*, 2018.

[5] F. M. Benčić and I. P. Žarko, "Distributed ledger technology: Blockchain compared to directed acyclic graph," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1569–1570.

[6] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance," *Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep*, vol. 34, 2016.

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[8] L. D. Nguyen, A. E. Kalor, I. Leyva-Mayorga, and P. Popovski, "Trusted wireless monitoring based on distributed ledgers over nb-iot connectivity," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 77–83, 2020.

[9] D.-L. Nguyen, I. Leyva-Mayorga, and P. Popovski, "Witness-based approach for scaling distributed ledgers to massive iot scenarios," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. IEEE, 2020, pp. 1–6.

[10] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT professional*, vol. 19, no. 4, pp. 68–72, 2017.

[11] Y. Zhang, C. Xu, N. Cheng, H. Li, H. Yang, and X. Shen, "Chronos+: An accurate blockchain-based time-stamping scheme for cloud storage," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 216–229, 2019.

[12] C. Profentzas, M. Almgren, and O. Landsiedel, "Tinyevm: Off-chain smart contracts on low-power iot devices," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, pp. 507–518.

[13] "Bitcoin blockchain size 2009-2022 | statista," https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/, (Accessed on 05/08/2022).

[14] "Ethereum chain full sync data size," https://ycharts.com/indicators/ethereum_chain_full_sync_data_size, (Accessed on 05/08/2022).

[15] P. Danzi, A. E. Kalor, C. Stefanovic, and P. Popovski, "Analysis of the communication traffic for blockchain synchronization of iot devices," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.

[16] A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor, "Blockchain and scalability," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2018, pp. 122–128.

[17] R. Hossain, D. Sarker, S. S. Meem, K. Shahrina, and M. Al-Amin, "Analysis of centralized payment eco-system: A systematic review on e-payments," 2020.

[18] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "Nb-iot system for m2m communication," in *2016 IEEE wireless communications and networking conference*. IEEE, 2016, pp. 1–5.

[19] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[20] A. Lavric, A. I. Petrariu, and V. Popa, "Sigfox communication protocol: The new era of iot?" in *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*. IEEE, 2019, pp. 1–4.

[21] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[22] L. D. Nguyen, I. Leyva-Mayorga, A. N. Lewis, and P. Popovski, "Modeling and analysis of data trading on blockchain-based market in iot networks," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6487–6497, 2021.

[23] C. Hine, "Privacy in the marketplace," *The Information Society*, vol. 14, no. 4, pp. 253–262, 1998.

[24] L. D. Nguyen, S. R. Pandey, S. Beatriz, A. Broering, and P. Popovski, "A marketplace for trading ai models based on blockchain and incentives for iot data," *arXiv preprint arXiv:2112.02870*, 2021.

[25] D. L. Nguyen, A. Lewis, I. Leyva-Mayorga, A. Regan, and P. Popovski, "B-ets: A trusted blockchain-based emissions trading system for vehicle-to-vehicle networks," in *7th International Conference on Vehicle Technology and Intelligent Transport Systems*. SCITEPRESS Digital Library, 2021, pp. 171–179.

[26] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the internet of things," *Ieee Access*, vol. 6, pp. 32 979–33 001, 2018.

[27] L. D. Nguyen, A. E. Kalor, I. Leyva-Mayorga, and P. Popovski, "Trusted wireless monitoring based on distributed ledgers over nb-iot connectivity," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 77–83, 2020.

[28] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2008.

[29] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

[30] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, 2018.

[31] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, no. 7, pp. 1–7, 2018.

[32] N. Szabo, "Formalizing and securing relationships on public networks," *First monday*, 1997.

[33] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[34] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.

[35] L. D. Nguyen, I. Leyva-Mayorga, A. N. Lewis, and P. Popovski, "Modeling and analysis of data trading on blockchain-based market in iot networks," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[36] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, "Blockchain technology and its relationships to sustainable supply chain management," *International Journal of Production Research*, vol. 57, no. 7, pp. 2117–2135, 2019.

[37] Y. Zhang, X. Xu, A. Liu, Q. Lu, L. Xu, and F. Tao, "Blockchain-based trust mechanism for iot-based smart manufacturing system," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1386–1394, 2019.

[38] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.

[39] P. Danzi, A. E. Kalor, R. B. Sorensen, A. K. Hagelskjær, L. D. Nguyen, C. Stefanovic, and P. Popovski, "Communication aspects of the integration of wireless iot devices with distributed ledger technology," *IEEE Network*, vol. 34, no. 1, pp. 47–53, 2020.

[40] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," in *Proc. of NIPS Wksp. PMPML*, Barcelona, Spain, Dec. 2016. [Online]. Available: https://arxiv.org/abs/1610.05492

[41] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," *arXiv preprint arXiv:1812.02903*, 2018.

[42] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2019.

[43] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[44] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. of NIPS*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Long Beach, USA: Curran Associates, Inc., Dec. 2017, pp. 4424–4434. [Online]. Available: http://papers.nips.cc/paper/7029-federated-multi-task-learning.pdf

[45] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," *arXiv preprint arXiv:1901.11173*, 2019.

[46] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," *ArXiv preprint*, vol. abs/1806.00582, Jun. 2018.

[47] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, 2019.

[48] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[49] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K. Y. Lam, "Local differential privacy based federated learning for internet of things," 2020.

[50] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020.

[51] M. Yang, L. Lyu, J. Zhao, T. Zhu, and K.-Y. Lam, "Local differential privacy and its applications: A comprehensive survey," 2020.

[52] Y. Koda, K. Yamamoto, T. Nishio, and M. Morikura, "Differentially private aircomp federated learning with power adaptation harnessing receiver noise," 2020.

[53] A. Elgabli, J. Park, C. B. Issaid, and M. Bennis, "Harnessing wireless channels for scalable and privacy-preserving federated learning," *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5194–5208, 2021.

[54] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 931–948.

[55] L. Gudgeon, P. Moreno-Sanchez, S. Roos, P. McCorry, and A. Gervais, "Sok: Off the chain transactions." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 360, 2019.

[56] X. Li, X. Wang, T. Kong, J. Zheng, and M. Luo, "From bitcoin to solana–innovating blockchain towards enterprise applications," in *International Conference on Blockchain*. Springer, 2021, pp. 74–100.

[57] G. Bu, Ö. Gürcan, and M. Potop-Butucaru, "G-iota: Fair and confidence aware tangle," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 644–649.

[58] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[59] M. Mazzoni, A. Corradi, and V. Di Nicola, "Performance evaluation of permissioned blockchains for financial applications: The consensys quorum case study," *Blockchain: Research and applications*, vol. 3, no. 1, p. 100026, 2022.

[60] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[61] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Lsb: A lightweight scalable blockchain for iot security and anonymity," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 180–197, 2019.

[62] W. Zhang, J. Yu, Q. He, N. Zhang, and N. Guan, "Tick: Tiny client for blockchains," *IEEE Internet of Things Journal*, 2020.

[63] X. Liang, J. Zhao, S. Shetty, and D. Li, "Towards data assurance and resilience in iot using blockchain," in *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 2017, pp. 261–266.

[64] S. Jiang, Y. Li, Q. Lu, Y. Hong, D. Guan, Y. Xiong, and S. Wang, "Policy assessments for the carbon emission flows and sustainability of bitcoin blockchain operation in china," *Nature communications*, vol. 12, no. 1, pp. 1–10, 2021.

[65] "Number of crypto coins 2013-2022 | statista," https://www.statista.com/statistics/863917/number-crypto-coins-tokens/, (Accessed on 05/13/2022).

Paper A

Witness-based Approach for Scaling Distributed Ledgers to Massive IoT Scenarios

Authors:
Duc-Lam Nguyen, Israel Leyva-Mayorga, and Petar Popovski

The paper has been published in the
IEEE 6th World Forum on Internet of Things (WF-IoT), pp. 1-6. IEEE, 2020.
⋆ **Best Student Paper Award** ⋆

The layout has been revised.

Abstract

Distributed Ledger Technologies (DLTs) are playing a major role in building security and trust in Internet of Things (IoT) systems. However, IoT deployments with a large number of devices, such as in environment monitoring applications, generate and send massive amounts of data. This would generate vast number of transactions that must be processed within the distributed ledger. In this work, we first demonstrate that the Proof of Work (PoW) blockchain fails to scale in a sizable IoT connectivity infrastructure. To solve this problem, we present a lightweight distributed ledger scheme to integrate PoW blockchain into IoT. In our scheme, we classify transactions into two types: 1) global transactions, which must be processed by global blockchain nodes and 2) local transactions, which can be processed locally by entities called witnesses. Performance evaluation demonstrates that our proposed scheme improves the scalability of integrated blockchain and IoT monitoring systems by processing a fraction of the transactions, inversely proportional to the number of witnesses, locally. Hence, reducing the number of global transactions.

1 Introduction

Distributed Ledger Technologies (DLTs) provide high levels of security, accountability, tractability, and privacy to the transmitted data [1]. This is achieved by enabling key functionalities, such as transparency, distributed operation, and immutability [2]. The benefits of DLTs are particularly appealing for Internet of Things (IoT) applications, where large amounts of data are generated and the devices can only implement weak security mechanisms [3].

The trust provided by DLTs is greatly valuable in IoT monitoring applications with a large number of devices. As an example, consider an urban IoT application that monitors the air quality and gas emissions. The data generated by this application is critical, so it must be protected, tractable, immutable, and transparent. Nevertheless, in a traditional monitoring system, the inter-organization sharing the data may be untrusted, complex, unreliable, and non-transparent. Besides, the current IoT-based monitoring systems are centralized, which leads to a single point of failure, where data can be lost or modified [4].

The problems described above may be solved by integrating Blockchain into IoT applications. However, Blockchain architectures were not designed to handle a large number of transactions, which would be generated by naively integrating Blockchain into IoT. Specifically, IoT deployments usually present a star topology, in which the devices communicate directly to the base stations (BS), which then redirects the gathered data to the destination [5] (e.g., from Narrowband IoT (NB-IoT) or LoRa deployments to a cloud server), as shown in the left part of Fig. A.1. In the most Blockchain and IoT integration, this same architecture would be used, and the BS would be in charge of communicating with the Blockchain [6]. Thus, every packet generated by the IoT devices would represent a transaction, which can easily overload the Blockchain.

Three main challenges must be overcome to achieve an efficient integration of Blockchain into IoT. First, DLTs use diverse resource-intensive gating functions, for example, *Proof-of-Work (PoW)* and *Proof-of-Stake (PoS)*, while IoT devices are resource-constrained. As a consequence, the processing time of these functions in IoT devices would be restrictive. Second, the widely-used Blockchain arrangement cannot handle the massive transactions generated by

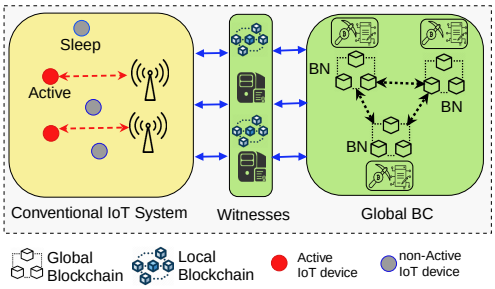


Fig. A.1: Overview of our Blockchain-enabled IoT system *wiBlock*. The IoT nodes generate and send the transactions to the base stations, which in turn send them to the witnesses. These decide which transactions must be sent to the GB and process the rest.

IoT devices. For example, Bitcoin network produces 1 MB blocks, roughly once every 10 minutes, with an average size of transaction around 500 bytes, which give 7 transactions per second (tps). In comparison, Visa system performs 2000 tps on average, and an average daily peak of 4000 tps, with a maximum capacity of 56000 tps. Third, the power saving mechanisms of the IoT devices can cause problems during knowledge dissemination and synchronization. For instance, an update may be severely delayed or even fail to arrive if a device is in sleep mode.

In this paper, we present a witness-based Blockchain system called *wiBlock*, especially designed to integrate Blockchain into resource-constrained IoT applications. It is aimed to solve three of the main problems of traditional IoT monitoring systems, namely trust, scalability, and cost. This is achieved by: 1) enabling the use of DLTs to store IoT data, 2) limiting the number of transactions that must be processed at the Global Blockchain (GB), and 3) eliminating the need for complex computations and supporting sleep-awake mechanisms at the IoT devices, respectively.

The architecture of *wiBlock* is illustrated in Fig. A.1, where the IoT devices interact exclusively with the *witness* system, which then may process the transactions locally or communicate directly with the GB. The transactions that must be processed by the GB are called *global transactions*, whereas the transactions that can be verified locally at the *witness system* are called *local transactions*. In order to see the need for this differentiation, consider a pollution monitoring system, in which a number of sensors in a given local area are associated to the same witness. Then a local transaction can be used to send local sensing data from a device associated with the same witness. For instance, the alarm sensor periodically requests gas sensor which collects the concentration of pollutants e.g., SO_2, CO_2, NO to detect the abnormal condition in air. In order to see the need for a global transaction, note that sensors may wish to store their sensing data to external storage system e.g, IPFS [7] or control a thermostat sensor, which is located in a different area and associated with a different witness to adapt temperature. In this case there is a need to communicate via different heterogeneous networks and record the transaction results to the GB via global transactions. Thus, the witness system reduces the number of transactions that need to be processed by the *GB* and the latency of transaction verification. Furthermore, *wiBlock* allows each IoT device to communicate with several witnesses. This avoids having a single point of failure (i.e., bridge) between the IoT device and the GB, which in turn greatly increases the reliability of the IoT application. For example, Blockchain witness models have been found to be beneficial for Cloud Service Level Agreement [8].

The contributions of this work are as follows:

1. We investigate the possibilities of naively integrating Blockchain directly into resource-constrained IoT systems. We identify some of the major problems that arise in this setup, which illustrate that Blockchain technology is not directly applicable to massive IoT.

2. We propose a new IoT-friendly distributed ledger system named *wiBlock*. It aims to solve the scalability issues of Blockchain in massive IoT environment by defining two types of transactions: global and local.

3. We thoroughly compare the performance *wiBlock* with that of a naive Blockchain and IoT integrated architecture. Our results show that our proposed system enhances the scalability of the GB network.

The remainder of this paper is organized as follows. In Section 2, we present the system model, followed by the design of our novel *wiBlock* system in Section 3. We present the analysis and performance evaluation of *wiBlock* in Section 4 and Section 5, respectively. Finally, we conclude the paper in Section 6.

2 System model

We consider an IoT application with k devices. These are deployed uniformly at random in a squared area of interest $A \in \mathbb{R}^2$. The IoT devices generate transactions with the data collected from the environment according to a Poisson process with rate λ.

In the most simple Blockchain and IoT integrated architecture, the transactions are sent to the BS, which then redirects them to the GB. In *wiBlock*, the transactions are sent to the *witness system* instead. This is a set of v witnesses, which have the capacity to verify transactions locally and to communicate with the GB. The time needed for a witness to perform these operations determine its capacity and depend on numerous factors. However, it is out of the scope of this paper to derive their precise values. Transactions are grouped into blocks of size b. Therefore, a new block is created when b new transactions are received at a server.

Witnesses may be either physical or logical entities, hence, their organization is flexible. For simplicity, throughout this paper we assume one witness is deployed at each BS and use these terms interchangeably. The BSs are distributed randomly within A. We denote the set of IoT devices and witnesses as $\mathcal{D} = \{1, 2, \ldots, k\}$ and $\mathcal{W} = \{1, 2, \ldots, v\}$, respectively.

The IoT devices and witnesses communicate through wireless links under a standard path loss model and large-scale (slow) fading. Thus, a transaction is transmitted successfully from IoT device i to a witness $w \in \mathcal{W}$ with probability $p_s(i, w)$. The IoT device i selects the witness w according to a predefined strategy. If the transmission fails, i attempts the transmission to a different witness. This process is repeated until the transaction is confirmed or until a given number of attempts is reached without success.

We consider a simple shadowing propagation model for the communication between IoT devices and witnesses where, for a given transmission power P_t and carrier frequency f, the received power at a distance d is

$$P_r(d) = 10 \log_{10} \left(\frac{P_t G_t G_r \, c^2}{(4\pi f)^2 d^\beta} \right) + N(0, \sigma_{\text{dB}}) \text{ dB} \tag{A.1}$$

where G_t and G_r are the transmitter and receiver antenna gains, respectively, $c = 3 \cdot 10^8$ m/s is the speed of light, $N(0, \sigma_{\text{dB}})$ is a zero-mean Gaussian random variable (RV) with standard deviation σ_{dB} dB, and β is the path loss exponent.

From there, the outage probability at a given distance and receiver sensitivity γ is

$$p_{\text{out}}(d) = 1 - Q\left(\frac{1}{\sigma_{\text{dB}}} 10 \log_{10}\left(\frac{\gamma (4\pi f)^2 d^{\beta}}{P_t G_t G_r c^2}\right)\right) \tag{A.2}$$

and $p_s(i, w) = 1 - p_{\text{out}}(d(i, w))$. Throughout this paper, we assume that the wireless resources are sufficient to support the communication between the IoT devices and the witness system and do not go into the details of the access protocols. Therefore, collisions caused by simultaneous transmissions from the IoT devices to a witness w can be avoided or resolved if the links toward w are not in outage. Finally, no errors occur in the communication between the witness system and the GB.

3 *WiBlock* Design

This section presents the detailed description of *wiBlock* architectural elements and operation.

3.1 Witness-based Blockchain System

As illustrated in Fig. A.1, the witness-based Blockchain System consists of three main components: the GB, the witness system, and the physical IoT devices. The first action performed by the IoT devices after deployment is authentication. For this, each device $i \in \mathcal{D}$ performs a key exchange procedure with a witness $w \in \mathcal{W}$ to gain the necessary permissions and build secure channels to perform transactions. After authentication, the tuple (i, w) is added by w to the shared registry of the witness system $\mathcal{R}$. For this, w shares the authentication information of i (i.e., credentials) with the rest of the witnesses. By keeping a shared registry, device i can communicate with any witness, even though is registered with w. After authentication, IoT devices collect the data and sign it by using a `SecretKey` $s_{\text{key}}(i)$ that is unique for each i as *`Sign`(data, $s_{key}(i)$, timestamp τ)*. Next, the transaction is created and transmitted to a *witness* w. Note that this latter witness may be different to the one which i is registered with. Local transactions, denoted as L_l, are exchanged exclusively between w and all the IoT devices registered with it $\{i \in \mathcal{D} : (i, w) \in \mathcal{R}\}$, for which the managers implement a consensus procedure, as shown in Fig. A.2. On the other hand, Global transactions, denoted as L_G, must be sent from a *witness* w to the GB when $(i, w) \notin \mathcal{R}$. These two types of transactions are further described in the following.

Local Transactions

These transactions are transmitted from IoT device i to the witness w, whose key management component confirms that $(i, w) \in \mathcal{R}$. Then, this same component checks whether the `PublicKey` $p_{\text{key}}(i)$ of i has been associated with any block in the *local ledger*. If $p_{\text{key}}(i)$ has not been associated with any block, the *witness* w generates a new block for the given i. Then, w arranges the transactions in order, updates the *local ledger*, and a notification feedback message is transmitted to the devices.

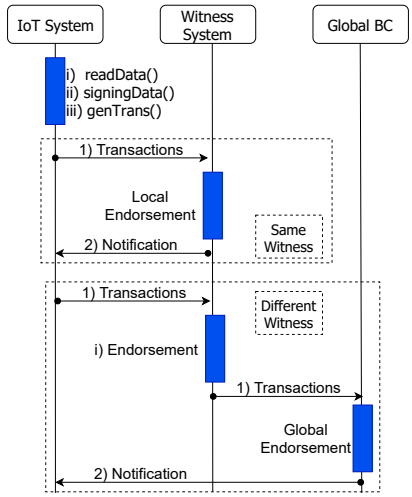


Fig. A.2: Transaction flow in *wiBlock*, from generation to confirmation.

Global Transactions

These transactions are transmitted from IoT device i to the witness w, whose key management component confirms that $(i, w) \notin \mathcal{R}$. Then, this same component will clarify which *witness* i is registered with. If $\exists w' \in \mathcal{W}$ s.t. $(i, w') \in \mathcal{R}$, the transaction is forwarded to the GB. In case the GB has a block associated with given device i, the transaction will be validated based on the corresponding signature $\mathtt{Sign}$*(data,* $s_{key}(i)$*, timestamp* τ*)* and, if the signature is valid, the transaction is appended to the block and transmitted back to the *witness* w'. Note that this type of transactions will be frequently generated when the IoT devices are mobile. For example, cargo, supply chain, and car subsystem monitoring.

3.2 Witness Selection

Numerous witness selection strategies can be implemented at the IoT devices and each one may offer different benefits. However, the focus of the present work is to evaluate the benefits of the witness-based architecture, rather than to identify an optimal witness selection strategy. Therefore, we consider the following a heuristic witness selection strategies and evaluate the performance of the witness system. As illustrated in Fig. A.3, IoT devices select one of the v available witnesses with probability $1/v$ and transmit the transaction. Then, if the link between IoT device i and witness w is not in outage, the transaction is confirmed. Otherwise, i selects a new witness uniformly at random from $\mathcal{W} \setminus w$ and transmits the transaction. This process is repeated until the transaction is confirmed or until a given number of attempts $l \leq v$ is reached without success. This is the simplest strategy and assumes the IoT devices have no information about the state of the wireless channel toward each witness separately.

4 Analysis

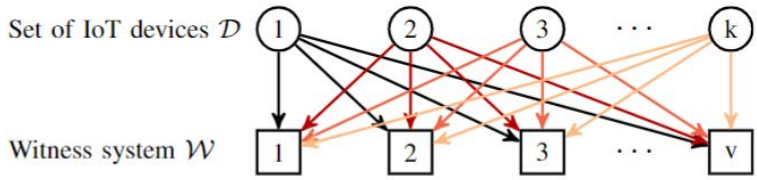


Fig. A.3: In *wiBlock*, each IoT device has a list of eligible witnesses. Transactions generated by the IoT devices are sent to a witness in this list, according to the witness selection strategy.

4.1 Queuing model of the witness system

We consider a queuing model for witness-based Blockchain network as described in Fig. A.4. The witnesses and Blockchain are modelled as queuing nodes to capture the number of transactions that must be i) processed locally by witnesses and ii) forwarded to the GB to be processed. We assume that transactions are generated by the IoT devices following a Poisson process. Hence, we denote $\lambda(i)$ as the transaction generation rate at IoT device i.

Let $p(i,w)$ be the probability that i chooses witness w and $p_s(i,w)$ be the probability that the link between i and w is not in outage. Building on this, the average transaction arrival rate at the witness w is

$$\lambda_w = \sum_{i=1}^{k} p(i,w) p_s(i,w) \lambda(i). \tag{A.3}$$

Hence, the transaction arrival rate of different witnesses depends on the density and location of the deployed IoT devices and witnesses, but also on the witness selection criteria.

The probability $p(i,w)$ depends on the witness selection strategy. For the strategy 1, random selection, let $A(w,u)$ be the matrix of permutations of u elements taken from $\{1 - p_s(i,w')\}_{w' \in \mathcal{W} \setminus w}$ with ${}_{(v-1)}P_u$ rows and u columns. The element in row x and column $y \leq u$ of $A(w,u)$ is denoted $a_{xy}(w,u)$. From there, we can calculate $p(i,w)$ as:

$$p(i,w) = \frac{1}{v} + \frac{1}{v!} \sum_{u=1}^{l-1} (v-u-1)! \sum_{x=1}^{{}_{(v-1)}P_u} \prod_{y=1}^{u} a_{xy}(w,u) \tag{A.4}$$

As mentioned above, generated transactions are either *global* L_G or *local* L_l. We define p as the probability that a transaction sent to a witness is *Global*. Hence, $1-p$ is the probability that a transaction is *local*. Please observe that the value of p only depends on the number of witnesses v and is $p := \Pr[(i,w) \notin \mathcal{R}] = (v-1)/v$.

The transaction processing time is assumed to follow an exponential distribution with service rates μ_1 and μ_2 for *global* and *local* transactions, respectively, and transactions are served according to a first-come first-served (FCFS) policy. Building on this, we model the operation of each witness as an M/H2/1 queue, which means that transactions arrive at the witness w at a rate λ_w and the service time is represented by a two-phase hyper-exponential distribution. With probability p, the first transaction in the queue receives service at rate μ_1, while with probability $1-p$, it receives service rate at rate μ_2. That is, the type of transaction is defined at the beginning of service.

The state of each witness is represented by a pair (m,n), in which m is the total number of transactions in the witness and $n \in \{1,2\}$ is the current service phase, which depends on

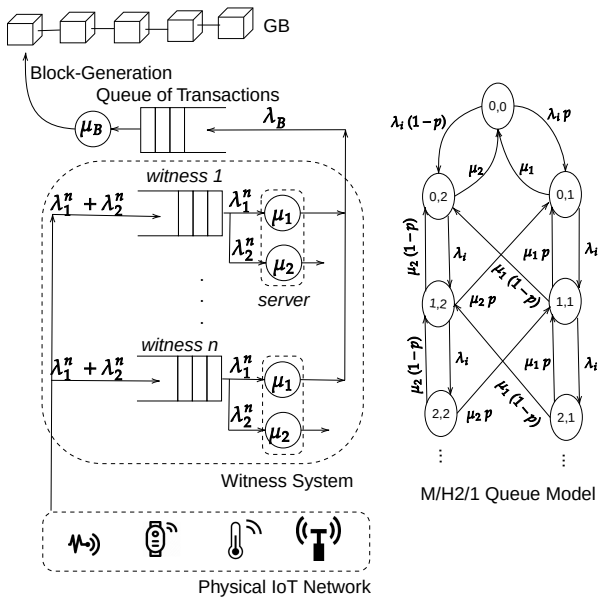


Fig. A.4: Witness-based Blockchain queuing model described in Section 4.

the type of transaction being served. The stationary distribution of this queue in the witness w can be obtained by Neuts' Matrix Geometric Method [9]. We denote the stationary probability vector as:

$$\tau^{(w)} = \left[\tau_0^{(w)}, \tau_1^{(w)}, \tau_2^{(w)}, \ldots, \tau_k^{(w)}, \ldots\right], \tag{A.5}$$

where $\tau_m^{(w)}$ is the steady-state probability of m transactions in the witness w. Alternatively, the mean service rate is

$$\mu = \left(\frac{p}{\mu_1} + \frac{1-p}{\mu_2}\right)^{-1}, \tag{A.6}$$

and the offered load to w is $\rho_w = \lambda_w/\mu$. From there, the we calculate the variance of the service time

$$\sigma_w^2 = 2\left(\frac{p}{\mu_1^2} + \frac{1-p}{\mu_2^2}\right) - \frac{1}{\mu^2} \tag{A.7}$$

and the coefficient of variation $C_w^2 = \mu^2\sigma_w^2$. Then, the average number of transactions in the queue of w is

$$L(w) = \sum_{m=0}^{\infty} m\tau_m^{(j)} = \rho_w + \left(\frac{1+C_w^2}{2}\right)\frac{\rho_w^2}{1-\rho_w}. \tag{A.8}$$

Then, the number of *local transactions* and *Global transactions* handled by w are, respectively,

$$L_g(w) = pL(w) \tag{A.9}$$

and

$$L_l(w) = (1-p)L(w) = L(w) - L_g(w). \tag{A.10}$$

Table A.1: Parameter settings for the performance evaluation.

Parameter	Symbol	Value
Area of deployment	A	100×100 m^2
Number of IoT devices	k	500
Number of witnesses	v	$\{2, 3, ..., 10\}$
Carrier frequency	f	914 MHz
Transmission power	P_t	0.28183815 W
Antenna gains	G_t, G_r	1
Receiver sensitivity	γ	$3.652 \cdot 10^{-10}$ W
Standard deviation of shadow fading	σ_{dB}	6 dB
Path loss exponent	β	3
Block size	b	1000 transactions

4.2 Global Blockchain (GB) System

We model the GB as a modified $M/G^B/1$ queue as in [10]. Let L_g and T_g be the RVs that define the number of transactions in the Blockchain queue and the confirmation time. We are interested in finding their mean values. For this, we define b to be the maximum number of transactions in a block (i.e., the maximum block size). Hence, transactions are grouped into blocks and a new block is created when there are b transactions in the Blockchain server.

Given that p is the probability that a transaction sent to a witness is processed at the GB, the transaction arrival rate at the GB from the v witnesses in the IoT deployment is

$$\lambda_B = \sum_{w=1}^{v} \lambda_w p. \tag{A.11}$$

We denote U as of the block generation time (i.e., the time it takes to generate a block) at the GB. Then, we define U to be the continuous RV of the processing (i.e., service) time of a block at the GB. Hence, the system is stable and a limiting probability exists if and only if $\lambda_B E[U] < b$. The cumulative distribution function (CDF) and the probability density function (pdf) of U are denoted $G(x)$ and $g(x)$, respectively. We use these to calculate the hazard rate of U as

$$\theta(x) = \frac{g(x)}{1 - G(x)}. \tag{A.12}$$

Next, we define $L_g^s(t)$ as the number of transactions in server at time t, $L_g^q(t)$ as the number of transaction in the queue at time t, and $X(t)$ as the elapsed service time of the current transaction at t. From [11], we define

$$P_{m,n}(x,t)dx = \Pr\left[L_g^s(t) = m, L_g^q(t) = n,\ x < X(t) \leq x + dx\right] \tag{A.13}$$

to be the joint probability that, at time $t \geq 0$, there are $m \in \{0, 1, 2, ..., b\}$ and $n \in \{0, 1, 2, ..., x\}$ transactions in server and queue, respectively, and the elapsed service time lies between x

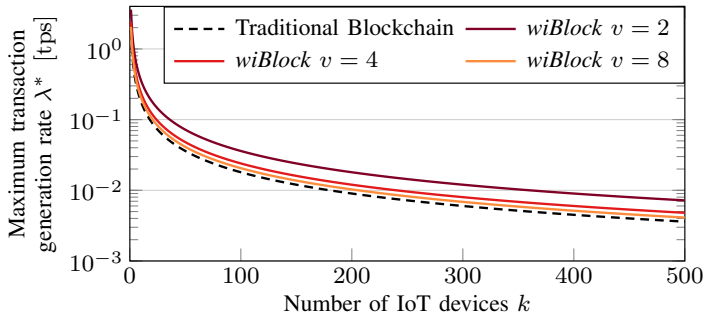


Fig. A.5: Maximum transaction generation rate per IoT device λ^* for traditional Blockchain IoT and *wiBlock* with random witness selection.

and $x + dx$. Next, we denote $P_{m,n}(x) = \lim_{t\to\infty} P_{m,n}(x,t)$ and consider the two following cases. In the first one we have $\frac{d}{dx}P_{m,n}(x) = -\left[\lambda_B + \theta(x)\right] P_{m,n}(x) + \lambda_B P_{m,n-1}(x)$, for $0 \leq m \leq b$ and $n \geq 1$, which shows that the number of transactions in the server and the queue does not change during a small interval. In the second one we have $\frac{d}{dx}P_{m,0}(x) = -\left[\lambda_B + \theta(x)\right] P_{m,0}(x)$, for $0 \leq m \leq b$, which occurs when a transaction arrives at the system with 0 transactions in the queue. For the purposes of our study, it is sufficient to calculate the mean confirmation time as

$$E[T_g] = \Bigg[\lambda_B^2 E[U^2] - b(b-1) - 2\left(b - \lambda_B E[U^2]\right) + \sum_{n=0}^{b-1} \alpha_n \Big(\lambda_B E[U^2](b-n) + 2bE[U](b-n) + E[U]\left(b^2 - b - n^2 + n\right)\Big)\Bigg] \frac{1}{2\lambda_B(b - \lambda_B E[U])}, \quad \text{(A.14)}$$

where $\alpha_n = \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\theta(x)dx$. The interested reader is referred to [10] for the fully detailed Blockchain queuing model.

5 Performance Evaluation

In this section, we use the queuing models described in Section 4 to evaluate the performance of *wiBlock* in terms of scalability. For this, we obtain the maximum transaction generation rate, along with the mean confirmation time and ledger size for both, the local and global Blockchain. We use the performance of a naive Blockchain and IoT integrated architecture, where the IoT devices communicate directly to the GB, as a benchmark. The mean results regarding the connectivity of the IoT devices with the witness system are obtained by a large number of Monte Carlo simulations and then used as an input to the queuing models.

In our analysis, each device generates transactions at a rate $\lambda(i) = \lambda$ for all $i \in \mathcal{D}$. The block generation time U is exponentially distributed with parameter $\mu_B = 1.8 \cdot 10^{-3}$ blocks per second. So we have $g(x) = \mu_B \exp(-\mu_B x)$, $E[U] = 1/\mu_B$, and $E\left[U^2\right] = 1/\mu_B^2$. Furthermore, we define the default block size to be $b = 1000$ transactions. The rest of relevant parameters are listed in Table A.1; these values are used unless otherwise stated.

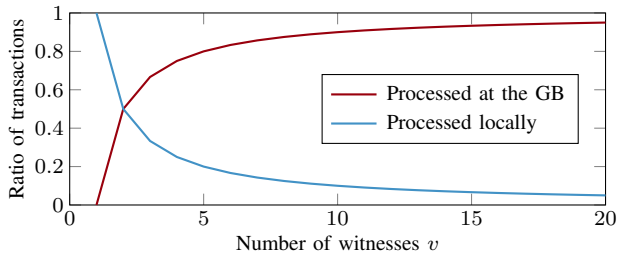


Fig. A.6: Ratio of transactions processed at the GB and at the witness system as a function of the number of witnesses v.

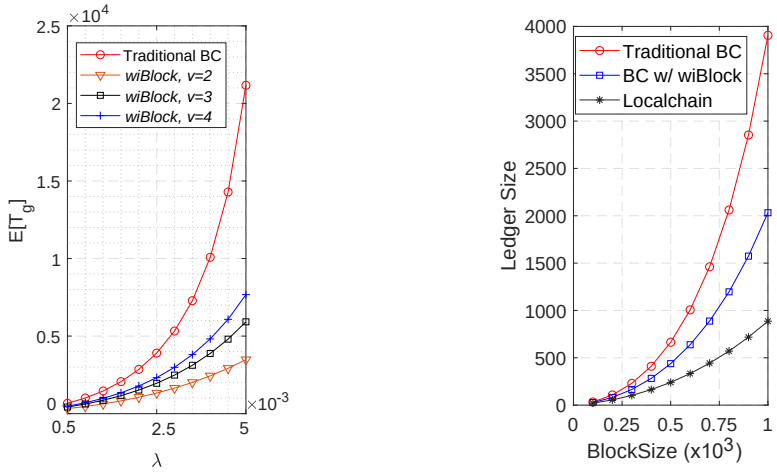


Fig. A.7: (a) Mean transaction confirmation time $E[T_g]$ as a function of the transaction generation rate at the IoT devices λ and (b) ledger size at the GB and at each witness for $v = 2$ as a function of the block size b for traditional Blockchain IoT and *wiBlock*.

For the selected parameter settings, the GB system is stable when $\lambda_B^* = b/E[U] = 1.8$. Building on this, from (A.11) we have that $\lambda < b/(kE[U]) = 1.8/k$ must hold for the GB to be stable in a traditional Blockchain architecture with k identical IoT devices. Conversely, for *wiBlock* with random witness selection, we have that only a fraction $p = (v-1)/v$ of the transactions must be sent to the GB. Hence, assuming no wireless channel errors occur and all the generated transactions are sent to a witness (i.e., $\sum_{w=1}^{v} \lambda_w = k\lambda$), the maximum load per IoT device that *wiBlock* can handle is

$$\lambda < \frac{bv}{kE[U](v-1)} = \frac{1.8v}{k(v-1)} = \lambda^*(v), \tag{A.15}$$

as shown in Fig. A.5 for $v = \{2, 4, 8\}$.

Hence, from the GB perspective, *wiBlock* allows to deploy $1/p = v/(v-1)$ times more IoT devices than the naively integrated approach, as illustrated by Fig. A.6. Note that the greatest gains in the scalability are obtained when v is small, however, other factors such as the area

coverage and processing capacity of the witness system must be taken onto account to select adequate values of v.

Next, we evaluate the mean transaction confirmation time at the GB $E[T_g]$. Note that, in case a single witness is deployed in the system, all the transactions generated by the IoT devices will be considered as local transactions and processed locally. This can overload the witness, depending on its capabilities. In particular, the witness is stable if and only if the load offered to the witness is $\lambda_1 < \mu_2$. Furthermore, deploying a single witness does not provide the necessary wireless coverage. That is, the more witnesses are deployed, the higher the probability of being able to communicate to, at least, one of them. Hence, we consider the cases where at least two witnesses are deployed, as shown in Fig. A.7a for $v = \{2, 3, 4\}$.

As Fig. A.7a shows, the witness system reduces the number of transactions sent to the GB and, as a consequence, greatly reduces the transaction confirmation time. Besides, the ledger size is considerably reduced, depending on the number of witnesses. This can be seen in Fig. A.7b for $v = 2$, where the ledger size of the GB is half of that with the traditional Blockchain and IoT integration, and the local ledger size at each witness is $1/v^2 = 1/4$ of it.

6 Conclusion

In this paper, we presented and evaluated the performance of a novel witness-based Blockchain system for IoT applications. As a starting point, we described the benefits of integrating Blockchain into IoT and the main challenges that must be overcome to achieve this integration. Building on these, we designed *wiBlock*, an IoT-friendly distributed system that incorporates a witness system to address scalability issues of Blockchain. The scalability gains provided by *wiBlock* are achieved by processing some of the transactions generated by the IoT devices locally, at the witness system. Our results show that the witness system greatly reduces the number of transactions transmitted to the Blockchain network and the transaction confirmation time. Future work includes the design of witness selection algorithms and implement *wiBlock* in real testbed to further exploit the benefits provided by the witness system.

7 Acknowledgment

This work has been in part supported by the European Research Council (Horizon 2020 ERC Consolidator Grant Nr.648382 WILLOW).

References

[1] P. Danzi, A. E. Kalor, R. B. Sorensen, A. K. Hagelskjær, L. D. Nguyen, C. Stefanovic, and P. Popovski, "Communication aspects of the integration of wireless iot devices with distributed ledger technology," *IEEE Network*, vol. 34, no. 1, pp. 47–53, 2020.

[2] M. Swan, *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc, 2015.

[3] A. Dorri, S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE International Conference on*

Pervasive Computing and Communications Workshops (PerCom Workshops), Mar. 2017, pp. 618–623.

[4] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based IoT: Challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, Jan. 2017.

[5] T. Maksymyuk, S. Dumych, M. Brych, D. Satria, and M. Jo, "An IoT based monitoring framework for software defined 5G mobile networks," in *Proc. of the 11th International Conference on Ubiquitous Information Management and Communication*, 2017, pp. 105:1–105:4.

[6] P. Danzi, A. Kalør, Č. Stefanović, and P. Popovski, "Delay and communication tradeoffs for blockchain systems with lightweight IoT clients," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2354–2365, Apr. 2019.

[7] M. S. Ali, K. Dolui, and F. Antonelli, "Iot data privacy via blockchains and ipfs," in *Proceedings of the Seventh International Conference on the Internet of Things*. ACM, 2017, p. 14.

[8] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat, and Z. Zhao, "A blockchain based witness model for trustworthy cloud service level agreement enforcement," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2019, pp. 1567–1575.

[9] W. Stewart, *Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling*. Princeton University Press, 2009.

[10] Y. Kawase and S. Kasahar, "Transaction-confirmation time for bitcoin: A queueing analytical approach to blockchain mechanism," in *Proc. International Conference on Queueing Theory and Network Applications*, 2017, pp. 75–88.

[11] M. Chaudhry and J. Templeton, "The queuing system M/G^B/l and its ramifications," *European Journal of Operational Research*, vol. 6, no. 1, pp. 56–60, 1981.

Paper B

Trusted Wireless Monitoring based on Distributed Ledgers over NB-IoT Connectivity

Authors:
Duc-Lam Nguyen, Anders E Kalor, Israel Leyva-Mayorga,
and Petar Popovski

The paper has been published in the
IEEE Communications Magazine 58, no. 6 (2020): 77-83.

The layout has been revised.

Abstract

The data collected from Internet of Things (IoT) devices on various emissions or pollution, can have a significant economic value for the stakeholders. This makes it prone to abuse or tampering and brings forward the need to integrate IoT with a Distributed Ledger Technology (DLT) to collect, store, and protect the IoT data. However, DLT brings an additional overhead to the frugal IoT connectivity and symmetrizes the IoT traffic, thus changing the usual assumption that IoT is uplink-oriented. We have implemented a platform that integrates DLTs with a monitoring system based on narrowband IoT (NB-IoT). We evaluate the performance and discuss the tradeoffs in two use cases: data authorization and real-time monitoring.

1 Introduction

An important element in the process of combating climate change and protecting public health is the reliable and trustworthy measurement of various emissions and air pollutants. Prime examples include CO_2 and NO_x, for which monitoring systems based on Internet of Things (IoT) technology have been reported in [1]. The emission information is critical and can have a significant economic value, such that the stakeholders have incentives to manipulate the data. The way this information from IoT-based monitoring systems is stored and collected raises concerns about data integrity, trust, security, transparency, and public availability. For instance, in IoT deployments, the measured data are either centralized or spread out across different heterogeneous parties. These data can be both public or private, which makes it difficult to validate their origin and consistency. Besides, querying and performing operations on the data becomes a challenge due to the incompatibility between different application programming interfaces (APIs). For instance, Non-Governmental Organizations (NGOs), Public and Private sectors, and industrial companies may use different data types and databases, which leads to difficulties when sharing the data.

Data authorization represents another critical component in many monitoring applications, in which the validity of the received information is critical. To this end, IoT monitoring systems often rely on an intermediary entity to validate the device signatures, e.g., a certificate authority (CA) server, which suffers from the issue of a single point of failure. As a result, the data from authenticated devices are vulnerable to tampering using, for example, man-in-the-middle attacks against the CA server.

Distributed ledger technologies (DLTs) are positioned as a key enabler for trusted and reliable distributed monitoring systems, since these support the immutable and transparent information sharing among involved untrusted parties [2]. In DLTs, the authentication process relies on consensus among multiple nodes in the network. While the terms DLT and *Blockchain* will be used interchangeably throughout this paper, Blockchains are a type of DLT, where chains of blocks are made up of digital pieces of information called transactions and every node maintains a copy of the ledger. Therefore, in a Blockchain-enabled IoT network, transactions contain, for example, environmental sensing data, or monitoring control messages, and these are recorded and synchronized in a distributed manner in all the participants of the system. These participants are called miners or peers and, in some specific DLTs, users are charged a transaction fee to perform (crypto) transactions. In addition, DLTs allow the storage of all transaction into immutable records and every record distributed across many participants. Thus, security in DLTs

comes from the distributed characteristic, but also the use of strong public-key cryptography and strong cryptographic hashes.

The benefits of the integration of DLTs into IoT monitoring systems include: i) guarantee of immutability and transparency for environmental sensing data; ii) removal of the need for third parties; iii) development of a transparent system for heterogeneous IoT monitoring networks to prevent tampering and injection of fake data from the stakeholders.

In this article, we describe and analyze the tradeoffs of the integration of DLTs into the narrowband Internet of Things (NB-IoT), which currently is the leading cellular IoT technology [3]. NB-IoT is one of the most efficient low-power wide-area network (LPWAN) technologies in terms of coverage, battery life-time, and support for massive machine-to-machine communications (i.e., scalability) [4]. A feature that makes NB-IoT more suitable than other technologies to support DLT traffic is its high downlink and uplink capacity when compared to other LPWANs such as LoRaWAN or Sigfox [5]. For instance, the suitability of LoRaWAN to support DLT traffic is mainly limited by its modest data rates and its 1 percent duty cycling (i.e., nodes must be idle 99 percent of the time) [2]. Conversely, NB-IoT has been designed with adaptable data rates and high flexibility, bringing significant advantages to sensing and monitoring networks. For instance, NB-IoT can be configured to use a wide range of sub-carrier spacing settings, which allows the protocol to be tailored for the specific deployment scenario and data rates can be increased 12 times by allocating multiple sub-carriers to the devices. NB-IoT provides extended coverage low-power devices with battery life-time up to 15 years [3]. Besides, NB-IoT is optimized for regular and small data transmissions, so it is well suited for monitoring devices acting as air quality, gas, and water meters [6].

We aim for a full integration where the NB-IoT devices generate transactions and receive the corresponding confirmations, but do not act as Blockchain nodes. Such integration is analogous to the P2 protocol described by Danzi *et. al* [7], which provides end-to-end (E2E) security and trust without increasing the storage and computation load of IoT devices. On the downside, such integration raises the following questions: i) how does Blockchain consensus and synchronization affect the NB-IoT connectivity in terms of uplink and downlink traffic and end-to-end (E2E) latency? and ii) which trade-offs arise from integrating DLTs into NB-IoT monitoring systems? For instance, the traffic patterns generated by DLTs are different to traditional IoT traffic, where the ratio of uplink (UL) to downlink (DL) data is oftentimes small. That is, most of the data is usually transmitted from the NB-IoT devices to the network to be stored and processed. Instead, the need to maintain a ledger for all the participants increases the amount of data transmitted from the base station (i.e., DL).

To answer these above questions, we first describe in detail the essential elements of an integrated Blockchain and NB-IoT system that addresses the problem of trust and privacy (Section 2.1). Then, we discuss and analyze the relevant characteristics of popular DLTs platforms such as Bitcoin, Ethereum, Hyperledger Fabric, and IoTA, and select the most promising to integrate into IoT environment monitoring systems. Additionally, we provide an overview of the operation of NB-IoT. Finally, we investigate the suitability of NB-IoT to connect the physical monitoring system with the Blockchain in two specific use cases, namely data authorization and real-time (i.e., timely) monitoring of gas emissions (Section 3). In particular, we analyze and evaluate the effect of Blockchain in NB-IoT monitoring systems in terms of traffic balance (DL to UL), communication overhead, and E2E latency, measured as the transaction confirmation time.

Our results, obtained from extensive experiments, show that the mining and consensus

mechanisms allow Blockchain nodes to reach a secure and tamper-resistant consensus in collected sensing data. On the downside, we observed an increase in the amount of DL data and E2E latency, in the order of a few seconds, when compared to traditional NB-IoT packet transmissions.

Despite these minor drawbacks, we consider that Blockchain and NB-IoT can have a symbiotic relationship to provide data integrity, trust, security, transparency, and public availability for a wide range of monitoring systems with a minimal impact on energy-efficiency.

In particular, the contributions of this work are threefold. First, we present a Blockchain-powered IoT framework for environmental monitoring systems that addresses the problem of trust and privacy. Second, we evaluate the proposed framework via extensive experiments, in which the NB-IoT monitoring system and a suitable DLT platform are integrated. Third, realizing the lack of studies on communication aspects of current Blockchain-enabled IoT systems, we analyze and evaluate the interaction between Blockchain and the NB-IoT monitoring systems in terms of overall throughput, E2E latency, and communication overhead via two case studies. Regarding previous studies [8], to our best knowledge, these studies mainly focus on specific-applications of Blockchain-enabled IoT, and how to integrate Blockchain with IoT. In our studies, communication aspects between Blockchain nodes and IoT devices are investigated.

2 Blockchain-powered IoT monitoring systems

In this section, we describe the essential architectural elements for integration of Blockchain into NB-IoT monitoring systems, evaluate the numerous DLT alternatives, and give a brief overview of the operation of NB-IoT.

2.1 Essential architectural elements

The overall integrated system consists of 4 key components: DLT network, physical sensors, edge network, and external resources, as illustrated in Fig. B.1. These components are described in following.

DLT Network: This component includes all modules to build various features of Blockchain technologies such as consensus, smart contract, data authorization, identity management, and peer-to-peer (P2P) communication. These components must ensure that every change to the ledger is reflected in all copies in seconds or minutes and provide mechanisms for the secure storage of the data generated by IoT devices and parameter configurations. There are numerous DLTs with different characteristics that may be beneficial for different target applications. The DLT nodes can be located everywhere and connected with NB-IoT base stations via the Internet.

Physical Sensors: The set of resource-limited devices which have the responsibility of collecting environmental data such as temperature, humidity, gas emissions and air quality levels. The collected data are transmitted to edge nodes or base stations, which can be static, such as access points, gateways, or mobile terminals, such as drones and mobile devices.

Edge Network: Even though DLT-based solutions offer significant countermeasures to secure data from tampering and support the distributed nature of the IoT, the massive amount of generated data from sensors and the high energy consumption required to verify transactions make these procedures unsuitable to execute directly on resource-limited IoT devices. Instead, edge servers with high computation resources can be used to handle real-time applications and

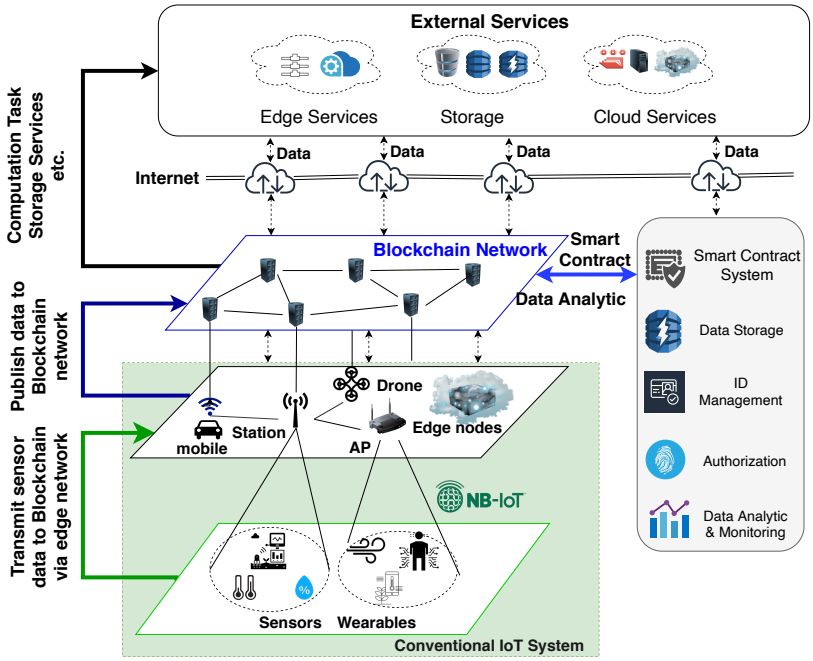


Fig. B.1: General DLT-enabled NB-IoT pollution monitoring architecture

to further increase the degree of privacy (e.g., through cloud computing) [9]. The edge network is a potential entity to cooperate with the Blockchain network in computationally heavy tasks and return the estimation results (e.g., from solving proof-of-work (PoW) puzzles, hashing or algorithm encryption) to the Blockchain network for verification.

External Services: IoT physical devices are resource-constrained with limited storage space and low computation capacity. Hence, external infrastructure may be incorporated to provide external services such as storage and computing. For example, the Interplanetary File System (IPFS) is a distributed file storage system that can store data generated from IoT networks and return a hash to the ledger based on the content of the data. Since the ledger cannot handle and store the massive amount of environmental data collected by the sensors, the services provided by the IPFS are a vital component.

2.2 Suitability of different DLTs for IoT monitoring

Although a large number of Blockchain DLTs are available, the most prominent platforms include Bitcoin, Ethereum, IOTA, and Hyperledger Fabric. In the following, we compare these DLTs in five different aspects: scalability, latency, throughput, security, and the level of smart contract functionalities.

Scalability, latency, and throughput are deeply related and of vital importance for IoT applications. For instance, the large number of sensors in smart cities may generate millions of transactions per day. This requires high efficiency of the consensus mechanism, including the way in which transactions are processed by the peers, known as *endorsing peers* in Hyperledger

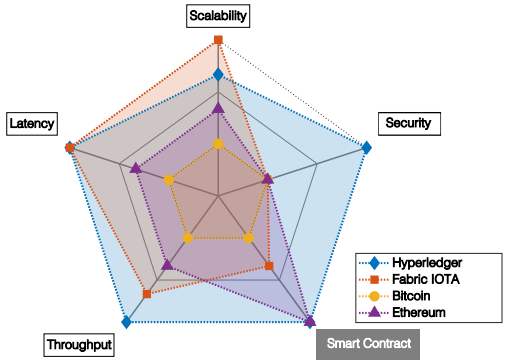


Fig. B.2: Performance of four different DLTs in five essential aspects for IoT monitoring systems dealing with sensitive information.

Fabric and full nodes (peers) in Bitcoin and Ethereum. Regarding latency, the transaction confirmation time must be sufficiently short to avoid queueing in the Blockchain and to ensure consistency in the ledgers. Bitcoin and Ethereum confirmation times per transaction are around 10 minutes and 25 seconds, respectively. These latencies might not be suitable for real-time IoT monitoring, while the confirmation time of Fabric and IOTA is much lower [10]. Note that the transaction confirmation time is only part of end-to-end latency, as it does not account for the communication latency at the radio access network.

The charge of fees to process the transactions, commonly known as gas is yet another factor to take into account to select the appropriate DLT. These may greatly increase the operational costs of the network, which negatively impacts the throughput of the DLT. On the one hand, transaction fees pose a problem in massive IoT scenarios if the generation of a large number of transactions is essential. On the other hand, these fees may contribute to minimize the amount of redundant transactions generated by the sensors, which in turn offloads the Blockchain. Among the considered DLTs, Ethereum requires fee and gas for each transaction whereas Hyperledger Fabric and IOTA provide free solutions to exchange transactions.

It is clear that IoT applications will involve many stakeholders with different roles, functionalities, and information with access rules, identities and security factors. An important factor to provide security is the support for permissioned and permissionless (i.e., hybrid) solutions to validate participating nodes. Both Ethereum and Hyperledger Fabric support public and private solutions, while Bitcoin and IOTA only provide public ones. Although IoT networks, such as smart cities, may have a large number of stakeholders willing to contribute to the security of a permissionless Blockchain network, permissioned networks could also be beneficial. For example, in smart homes where the homeowner wants to validate the transactions via home miners or validators [11]. Regarding security, public networks may be more secure than private ones if these are able to provide transparency and distributed storage. For instance, in a permissionless Blockchain, the data is encrypted and stored in all the devices, which makes it definitely transparent. Besides, the more users a permisionless Blockchain has, the more secure it is. However, permissionless Blockchains are not ideal for enterprise use, where companies deal with highly sensitive data and cannot allow anyone join their network. A permissioned Blockchain can be altered by its owners, making it more vulnerable to hacking [12]. In addition, permissioned Blockchains provide very low or no fee for validation and a faster consensus process.

Finally, smart contracts act as autonomous entities on the ledger that deterministically execute logic expressed as functions of the data that are written on the ledger. Therefore, smart contracts can be established to have automatic reactions from the DLT network to specific events. For example, in case of carbon emissions, smart contracts can be used for real-time policy enforcement upon changes in the emission patterns. The smart contract feature currently is supported by Ethereum and Hyperledger Fabric (Chaincode). An IOTA smart contract type called Quobic is still in progress. Besides, only Hyperledger Fabric supports data confidentially via in-band encryption and guarantees the privacy of data by creating private channels. Hyperledger Fabric provides a solution with various features such as identity management, transaction integrity and authorization with a trusted CA. These features are vital in a trusted IoT system. The comparison of the DLTs mentioned above in these areas illustrated in Fig. B.2, where each aspect has been given an abstract score based on the previous discussion. Note that the smart contract aspect is a functionality rather than a strict performance indicator and can only be scored qualitatively. This makes it different to the rest of the aspects reflected in Fig. B.2, hence, it is shown in a gray background.

Based on these scores, we decided to implement Hyperledger Fabric as DLT platform for our experiments on IoT monitoring.

2.3 DLT traffic over NB-IoT

In the following, we provide a brief description on the operation of NB-IoT devices, hereafter referred to as user equipments (UEs), in monitoring applications. NB-IoT UEs have only two modes of operation, namely radio-resource control (RRC) idle and RRC connected. In the former, the UEs can only receive the system information from the BS and, only in the latter, data can be transmitted. UEs are in idle mode before initial access to the network, but may also enter this mode during power saving or after an explicit disconnection request. To transition from idle to connected mode, the UEs (clients) must first acquire the basic system information and synchronization as illustrated in the upper part of Fig. B.3. For this, the UE receives the master information block (MIB-NB) and the system information blocks 1 (SIB1-NB) and 2 (SIB2-NB). These are transmitted periodically through the downlink shared channel (DL-SCH) and carry the basic cell configuration, timing, and access parameters [13]. In addition, SIB1-NB carries the scheduling information for the rest of the SIBs.

After the system information has been acquired, the UEs must perform the RA procedure to transition to RRC connected mode [14]. The RA procedure is a four-message handshake, initiated by the UEs by transmitting a single-tone frequency-hopping pattern, called preamble, through the NB Physical Random Access Channel (NPRACH). In most cases, the RA procedure is contention-based, hence, the preamble is chosen randomly from a predefined pool of up to 48 orthogonal sub-carrier frequencies.

After completing the RA procedure, and if the control-plane (CP) cellular IoT (CIoT) is used, UEs may piggyback short UL data packets along with the RRC Connection Setup Complete message. Otherwise, the non-access stratum (NAS) setup must be completed before eNB allocates resources for uplink transmission through the NB Physical UL Shared Channel (NPUSCH) and data can be transmitted. The resource unit (RU) is the basic unit for resource allocation in the NPUSCH and comprises a set of sub-frames in the time domain and sub-carriers in the frequency domain. The downlink (DL) data is transmitted through the NB physical DL shared channel (PDSCH).

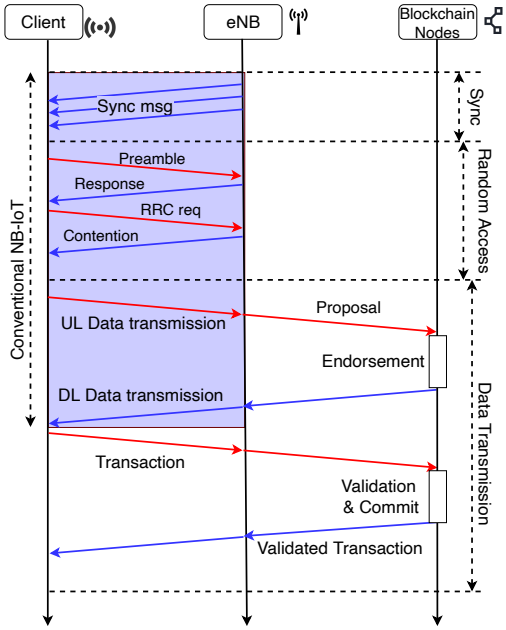


Fig. B.3: A sequence of message exchanges between DLT with UEs and eNB.

In a traditional NB-IoT monitoring system, the UL data generated by the UEs is transmitted though the NPUSCH and routed towards a data center or cloud server to be stored and processed. At this point, the monitoring system has no control on the collected data, so modification, corruption, and losses may occur. Conversely, in our Blockchain-enabled NB-IoT setup, the uplink data generated by the UEs is transmitted to a randomly chosen group of endorsing peers of Hyperledger Fabric as transaction proposals. Then, each of the peers signs the transaction using Elliptic Curve Digital Signature Algorithm (ECDSA) and adds the signature before returning the signed message back to the UEs.

The peers that provide an endorsement of the proposed ledger send an update to the application, but do not immediately apply the proposed update to their copy of the ledger. Instead, a response is sent back to the UEs to confirm that the transaction proposal is correct, has not been previously submitted to ledger, and has a valid signature. Therefore, the security increases with the number of endorsing peers. In addition, smart contracts can be executed to update or query the ledger. A simple example of a smart contract in air pollution monitoring systems would be to set the system to calculate average values of the collected data and to generate an alarm message whenever these exceed a predefined threshold.

Then, the UEs broadcast the confirmed transaction proposals along with the response (confirmation) to the ordering service. The received transactions are ordered chronologically to create blocks. These blocks of transactions are delivered to all the peers for validation. The peers append the block to the ledger, and the valid transactions are committed to the current state database. Finally, a confirmation message is emitted and transmitted back to the UEs to notify that the submitted transaction has been immutably published to the ledger. In our previous work [2], we have shown that two-way wireless communication is required to enable high

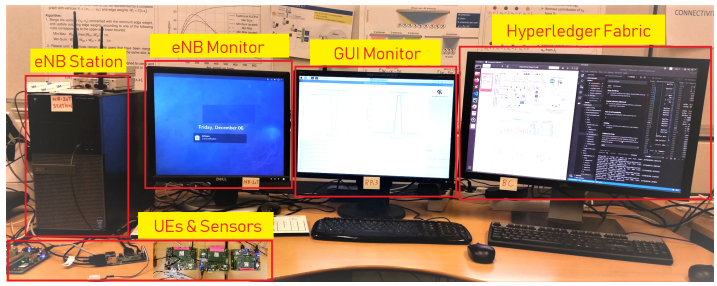


Fig. B.4: Blockchain-enabled NB-IoT implementation and setup.

decentralization. In communication aspects, we studied and analyzed DLT traffic in both uplink and downlink over IoT networks. The confirmation serves as proof that a transaction executed and recorded in distributed ledger. This confirmation helps to check for errors or strange occurrences, for example, in retail industry, the confirmation support to detect the number of items re-bought, etc. Then, it is fed into the overall retail system, inventory system, and more are updated. Furthermore, if the sensors do not receive confirmation from the distributed ledger, the next action depends on specific applications and configuration.

3 Case Studies

In this section, we evaluate the performance of an integrated Blockchain and IoT monitoring system with Hyperledger Fabric and NB-IoT under two use cases. The first one focuses on the data authentication aspect provided by Blockchain. We then extend the use case to include smart contracts to processes the sensor measurements. Our experimental setup is based on Hyperledger Fabric v1.4, NB-IoT development kits Sara EVK N211, and one NB-IoT Amarisoft eNB station, and is illustrated in Fig. B.4.

3.1 Use case 1: Data Authorization

The focus of this use case is to evaluate the communication overhead of data authentication in our setup. Our setup includes a single UE with a single sensor that follows the procedure described in Section 2.3; illustrated in Fig. B.3. The metric we use to evaluate the communication efficiency is the average UL to DL data traffic ratio, where only data packets are considered.

The size of the payload in the transmitted packets plays a vital role in the performance of DLT-based NB-IoT systems. Therefore, we varied the UL payload size from 50 B to 200 B and set the UE to generate a total of 1000 transactions (i.e., UL data packets). The DL payload size is set to 31 B, so the UL payload size is at least 1.61 times the DL payload size, and the block size is configured to 30 transactions per block. We ran our experiments with different number of endorsing peers in the DLT network E to observe the communication overhead of an increase in security, which, naturally, increases with E.

Our results are presented in Fig. B.5, where it can be seen that, naturally, the average UL to DL traffic ratio increases with the UL payload size. However, this increase is more rapid with traditional NB-IoT than with Blockchain and the average UL to DL traffic ratio decreases as E increases. For instance, with an UL payload size of 50 B, and $E = 2$, the DL traffic is

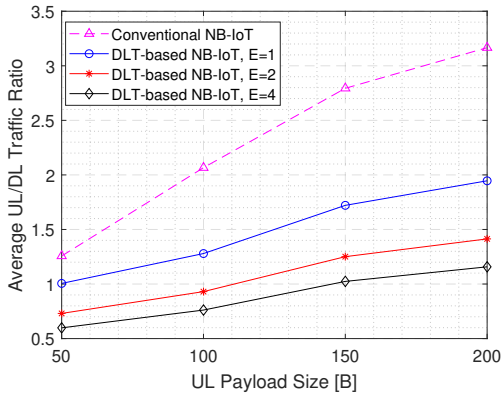


Fig. B.5: Average ratio of UL and DL traffic per transaction with different payload sizes and numbers of endorsing peers E.

almost twice as high as the UL traffic. The reason for this is that, certainly, the average traffic increases with the number of endorsing peers E, but the increase in DL traffic is much greater than the increase in UL traffic. Hence, these results highlight the fact that the use of DLTs heavily increases the traffic load in the DL channels of IoT networks, namely, in the PDSCH of NB-IoT.

3.2 Use case 2: Real-time Monitoring of Air Pollution

We now study the specific use case of a real-time CO_2 emission monitoring system that includes smart contracts in the DLT. The environment data is collected by $S8$ Miniature 10000 ppm CO_2 sensors in the UEs; a simple smart contract is defined to compute the average CO_2 level and trigger updates to the ledgers when these levels are abnormally high. Note that high indoor CO_2 levels are greatly correlated to human metabolic activity and can cause headaches or make the population to function at lower activity levels. Nevertheless, the CO_2 emissions generated by working equipment (e.g., computers, machines, etc.), also have an impact on total amount of CO_2 emissions. Our experiments on CO_2 and NOx data were conducted using the same methods to collect and process the data. Hence, the type of sensor does not affect the generated traffic and system process.

The focus in this use case is to evaluate the E2E latency, defined as the time elapsed from the generation of a transaction at the IoT device until its verification. This includes the latency at the NB-IoT radio link and at the DLT, which comprises the execution time of the smart contract and transaction verification. Therefore, the E2E latency of smart contract execution depends on the numerous parameters such as block size and transaction generation rate. Among these, we evaluate the impact of the block size on E2E latency. Our results indicate that integrating DLTs into NB-IoT monitoring applications symmetrizes the data traffic by slightly increasing the amount of data transmitted in the downlink. However, by adequately choosing the DLT and its parameters, the impact of DLT traffic on the battery lifetime of NB-IoT nodes may be relatively low when compared to that of the traffic pattern of the monitoring application and of the implemented power saving techniques. Hence, our results can be combined with detailed energy consumption models that include the different possible states and power saving

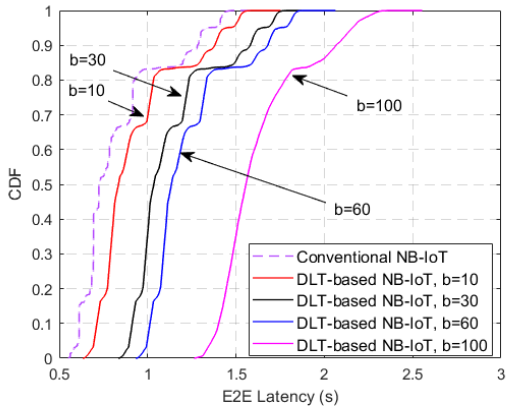


Fig. B.6: E2E latency of our Blockchain-enabled NB-IoT monitoring system.

techniques of NB-IoT nodes (e.g., [15]) to estimate their battery lifetime.

In these experiments, we configured two UEs, with one CO_2 sensor each, to gather data and upload to the ledger once every 10 s. Our results are shown in Fig. B.6, where various block sizes b, from 10 to 100 transactions per block, were considered; the E2E latency for traditional NB-IoT packets is included as a reference.

As expected, Fig. B.6 shows that an increase in block size comes with a slight increase in E2E latency. However, this increase is minor, even when compared to conventional NB-IoT packets, and may be suitable for most IoT monitoring applications. Specifically, the E2E latency is doubled from an average 0.832 s for conventional NB-IoT to 1.63 s for DLT-based NB-IoT with block size $b = 100$ transactions per block. Naturally, smaller block sizes lead to a smaller E2E latency, which is comparable to that of conventional NB-IoT, especially for $b = 10$. The reason is that the block creation time in Hyperledger Fabric increases with the increase of the block size. On the other hand, it is advisable to use small block sizes in monitoring applications where even conventional NB-IoT is close to the upper limit of the acceptable E2E latency of the system.

4 Conclusion

Monitoring of emissions based with IoT devices requires sets high demands for data reliability and trustworthiness. A promising approach in that direction is integration of a Blockchain into the IoT system. We have implemented Blockchain into an environmental monitoring system based on NB-IoT, analyzed the tradeoffs and evaluated the performance. Our results show that the integration of Blockchain increases the load in the downlink (DL) channel of NB-IoT, unlike the plain variant of NB-IoT that does not use Blockchain. Furthermore, both the level of security and the DL traffic load increase with the number of endorsing peers in Hyperledger. Besides, the E2E latency of the monitoring system increases slightly with the block size. This behavior was expected, but our results show that the increase in E2E latency is small even when compared to conventional NB-IoT. Therefore, integrated Blockchain and NB-IoT monitoring systems provide valuable benefits to a wide range of environmental applications and, in par-

ticular, to those that deal with sensitive information, such as carbon emissions or air pollution monitoring.

5 Acknowledgment

This work has been in part supported by the European Research Council (Horizon 2020 ERC Consolidator Grant Nr. 648382 WILLOW).

References

[1] Abawajy, J. H., & Hassan, M. M. (2017). Federated internet of things and cloud computing pervasive patient health monitoring system. *IEEE Communications Magazine*, 55(1), 48-53.

[2] Danzi, Pietro, et al. "Communication aspects of the integration of wireless iot devices with distributed ledger technology." *IEEE Network* 34.1 (2020): 47-53.

[3] Wang, Y-P. Eric, et al. "A primer on 3GPP narrowband Internet of Things." *IEEE Communications Magazine* 55.3 (2017): 117-123.

[4] Azari, Amin, et al. "On the Latency-Energy Performance of NB-IoT Systems in Providing Wide-Area IoT Connectivity." *IEEE Transactions on Green Communications and Networking* (2019).

[5] Mekki, Kais, et al. "A comparative study of LPWAN technologies for large-scale IoT deployment." *ICT express* 5.1 (2019): 1-7.

[6] Feltrin, Luca, et al. "Narrowband IoT: A survey on downlink and uplink perspectives." *IEEE Wireless Communications* 26.1 (2019): 78-86.

[7] Danzi, Pietro, et al. "Analysis of the communication traffic for blockchain synchronization of IoT devices." 2018 *IEEE International Conference on Communications (ICC)*. IEEE, 2018.

[8] Dai, Hong-Ning, Zibin Zheng, and Yan Zhang. "Blockchain for internet of things: A survey." *IEEE Internet of Things Journal* 6.5 (2019): 8076-8094.

[9] Xiong, Zehui, et al. "When mobile blockchain meets edge computing." *IEEE Communications Magazine* 56.8 (2018): 33-39.

[10] Xiao, Yang, et al. "A survey of distributed consensus protocols for blockchain networks." *IEEE Communications Surveys & Tutorials* (2020).

[11] Lin, Huichen, and Neil W. Bergmann. "IoT privacy and security challenges for smart home environments." *Information* 7.3 (2016): 44.

[12] Wüst, Karl, and Arthur Gervais. "Do you need a blockchain?." 2018 *Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018.

[13] 3gpp., “Radio resource control (rrc); protocol specification.”TS 36.331V15.3.0., Sept. 2018.

[14] 3GPP., “Medium access control (mac) protocol specification.”TS 36.321V15.2.0., July 2018.

[15] Lauridsen, Mads, et al. "An empirical NB-IoT power consumption model for battery lifetime estimation." 2018 *IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 2018.

Paper C

Modeling and Analysis of Data Trading on Blockchain-based Market in IoT Networks

Authors:
Duc-Lam Nguyen, Israel Leyva-Mayorga, Amari N. Lewis,
and Petar Popovski

The paper has been published in the
IEEE Internet of Things Journal 8, no. 8 (2021): 6487-6497.

The layout has been revised.

Abstract

Mobile devices with embedded sensors for data collection and environmental sensing create a basis for a cost-effective approach for data trading. For example, these data can be related to pollution and gas emissions, which can be used to check the compliance with national and international regulations. The current approach for IoT data trading relies on a centralized third-party entity to negotiate between data consumers and data providers, which is inefficient and insecure on a large scale. In comparison, a decentralized approach based on distributed ledger technologies (DLT) enables data trading while ensuring trust, security, and privacy. However, due to the lack of understanding of the communication efficiency between sellers and buyers, there is still a significant gap in benchmarking the data trading protocols in IoT environments. Motivated by this knowledge gap, we introduce a model for DLT-based IoT data trading over the Narrowband Internet of Things (NB-IoT) system, intended to support massive environmental sensing. We characterize the communication efficiency of three basic DLT-based IoT data trading protocols via NB- IoT connectivity in terms of latency and energy consumption. The model and analyses of these protocols provide a benchmark for IoT data trading applications.

1 Introduction

In 2025, the volume of sensing data generated by personal IoT devices is expected to reach 79.4 ZB globally [1]. Many attempts have been made to improve and adapt business workflows to exploit the availability of IoT data [2, 3]; among these, IoT data trading is the most popular approach. Various services for trading of IoT data are emerging, connecting various devices and distributed IoT data sources, thereby facilitating data providers to exchange their data [4].

Interesting use cases for data trading include public transport systems, for example, the bus network in Aalborg, Denmark. In these systems, the density of personal travel card swipes at specific bus stations could be useful information, not only to the administration of transport systems, but also to the local taxi companies. The taxi companies benefit from the data of anomalous passenger traffic patterns for the purposes of improving ride-sharing and private services [5]. Also, analyzed traffic data of passengers can be collected via IoT infrastructure and recommendation services to taxi companies can be sold. Besides, drivers can exchange information about the traffic status of a particular street with others to avoid traffic jams or to exchange green house gas emission information with manufacturers. Hence, IoT data can be considered as a tradable digital asset.

Traditional trading systems (e.g. Paypal) feature a single point of failure, the lack of trust, transparency, and incentive for data trading, which is preventing the availability of digital information from data providers to customers. On the other hand, Distributed ledger technologies (DLTs) and Blockchains[1] support immutable and transparent information sharing among involved untrusted parties [6]. Outside of its role in financial transactions, DLTs are seen as a key enabler for trusted and reliable distributed monitoring systems. The authentication process for DLTs relies on consensus among multiple nodes in the network [7]. In Blockchain-enabled

[1]The terms DLT and *Blockchain* will be used interchangeably throughout this paper, Blockchains are a type of DLT, where chains of blocks are made up of digital pieces of information called transactions and every node maintains a copy of the ledger

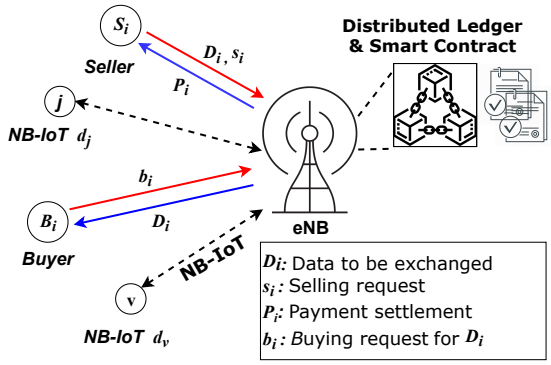


Fig. C.1: General system model of DLT-enabled IoT Data Trading via NB-IoT connectivity where seller $\mathcal{S}_i$ and buyer $\mathcal{B}_i$ make a deal on the data D_i.

IoT networks [8], transactions can include sensing data, or monitoring control messages, and these are recorded and synchronized in a distributed manner in all the participants of the system. These participants are called miners or peers and, in some specific DLTs, users are charged a transaction fee to deploy and execute transactions.

In addition, DLTs allow the storage of all transactions into immutable records and every record is distributed across many participants. Thus, security in DLTs comes from the decentralized operation, but also from the use of strong public-key cryptography and cryptographic hashes. The benefits of the integration of DLTs into IoT data trading systems include: i) guarantee of immutability and transparency for environmental sensing data, ii) removal of the need for third parties, iii) development of a transparent system for heterogeneous IoT data trading networks to prevent tampering and injection of fake data from the stakeholders [9].

With the spread of ubiquitous marketplaces, it became relevant to explore the use of IoT data trading in marketplace environments. For instance, in [10], Gupta et al. introduced the architecture for a dynamic decentralized marketplace for trading IoT data. The approach involves a 3-tier design: 1) provider, 2) broker and 3) consumer. The use of DLTs in their work is primarily to manage the terms of agreement between involved parties. Additionally, a reputation system is used in the design to penalize the participants and reduce their rating. Bajoudah et al. present a marketplace model and architecture for the trading of IoT streaming data in [11]. Within their work, periodic checkpoints during data exchange are introduced to limit fraudulent activity on either side. In [12], Missier et al. propose another marketplace, where streams of IoT data are the main assets traded utilizing Oracles for the off-chain queries. Xiong et al. [13] present a trading mode based on smart contracts. It incorporates machine learning to guarantee fairness of data exchange and utilizes arbitration institution to deal with the dispute over the data availability in the data trading. However, the arbitration institution in the trading mode is a trusted entity of trading parties. Dai et al. [14] introduced a secure data trading ecosystem based on Blockchain by combining the Intel Software Guard Extensions (SGX). The proposed ecosystem securely processes the data, but, the data source and analysis results highly depend on a trusted SGX-based execution environment. In [15], the authors proposed a decentralized Blockchain-based platform for data storage and trading in a wireless powered IoT crowd-sensing system. The data from RF-energy beacons are transmitted to the ledger for decentralized services, which supports the analytical condition for valuable results about the

equilibrium strategies in the distributed systems.

The related work indicates a knowledge gap in terms of: 1) a benchmark for IoT data trading, and 2) analysis of the cost of IoT data trading in terms of communication, specifically in city-level networks. The efficiency of a Blockchain-based data trading protocol is a major concern for data traders. Future markets will be highly dynamic and low latency trading is critical to maximize the efficiency of the marketplace. However, currently there is a lack of a general framework that provides a guideline for the use of trading protocols based on a set of neutral and commonly accepted rules. A proper benchmark helps the interested parties to understand the tradeoffs in Blockchain-based systems and the associated performance indicators.

In this paper, first, we design a DLT-based trading system for exchanging IoT data. We have chosen the NB-IoT standard [16] as the underlying connectivity solution, as it is seen by the mobile operators as a major candidate to dominate wide-range connectivity for future smart cities. Unlike many other IoT technologies, NB-IoT is able to offer symmetric uplink/downlink throughput, which is an essential feature from the viewpoint of a DLT [7, 17]. The proposed trading system includes the following IoT data trading protocols; *General Trading (GT)*, *Buying on Demand (BoD)*, and *Selling on Demand (SoD)*. Here, we use the term "on demand" from the perspective of the smart contracts that implement the transactions between buyers and sellers. Each trading protocol is customized for different scenarios. *GT* could be considered as the usual trading protocol in the data marketplace, while the *BoD* and *SoD* are protocols used to support particular demands from either sellers or buyers.

The analysis and simulation results show that the *GT* protocol has outstanding performance in terms of latency and energy consumption; however, it requires mechanisms to guarantee the continuous availability of data. On the contrary, the *BoD* protocol can be implemented in Vehicle-to-Infrastructure (V2I) networks, where vehicles can trade their emission information with manufactures. Finally, the *SoD* protocol is particularly useful when customers are interested in collecting specific data, which, however, may not be immediately available on the market. This protocol can also be deployed in Vehicle-to-Vehicle (V2V) networks where the drivers want to buy traffic jam information of a specific street from other vehicles on the road. Clearly, *SoD* protocols, on their own, would face situations in which the data is no longer available for customers after the initial advertising phase. In practice, the three trading protocols present interesting synergies and can be implemented together in a single system, which will select the best one based on the actual situation.

The contributions of this paper can be stated as follows. First, we present a solution for a systematic DLT-based IoT data smart trading towards city-level networks using NB-IoT connectivity. Next, we propose three IoT data trading protocols namely *General Trading (GT)*, *Buying on Demand (BoD)*, and *Selling on Demand (SoD)*. The cost model of each trading protocol is derived and analyzed along with NB-IoT connectivity. Both resources consumed by executing DLT/smart contracts and NB-IoT devices are investigated. Finally, the analysis and the associated experimental results provide a benchmark for data trading protocols in wide-area IoT networks.

The remainder of this paper is organized as follows. In the next section, we outline the general architecture of DLT-based trading system and introduce three IoT data trading protocols. Then, we present the system model, including the physical deployment of the devices. In Section III and IV, we model and analyze the performance of Blockchain-enabled IoT network in terms of latency and energy. Then, we evaluate and prove the derived model and design in Section V. Finally, we conclude the paper in Section VI.

Table C.1: Nomenclature

Parameters	Descriptions	Values
General		
N	Total number of NB-IoT devices	10000
M	Number of DLT miners	≤ 20
$\mathcal{S}$	Set of Data Providers (Sellers)	$\lvert\mathcal{S}\rvert \leq 10^4$
$\mathcal{B}$	Set of Data Consumers (Buyers)	$\lvert\mathcal{B}\rvert \leq 10^4$
$\mathcal{D}$	Data to buy or sell	–
$\mathcal{T} = \{\mathcal{T}_i\}$	Set of trades	–
λ^u	Uplink request arrival rate	Eq.3
λ^d	Downlink request arrival rate	Eq.3
G_t, G_r	Transmitter and receiver antenna gains	1
β	Path loss exponential	$\{2.4, 2.7, 3.0, 3.3\}$
γ	Receiver sensitivity	$3.6 * 10^{-10}$
λ_c	Computing speed of a miner	0.3
λ_0	Scaling factor	0.05
P_c	Power of miner	6
τ	The unit length	10 ms
d	The average time interval between two NPDCCH	$[0.05 : 0.2]$
$\mathcal{R}^u, \mathcal{R}^d$	Uplink and Downlink transmission rate	-
$E_{\mathcal{T}_i}$	Energy required to complete a trade $\mathcal{T}_i$	Eq. 7
E^u	Uplink energy consumption	Eq. 7
E^d	Downlink energy consumption	Eq. 7
E_{DLT}	Blockchain energy consumption	Eq. 6
E_{sync}	Energy required for synchronization	-
E^u_{rr}	Uplink: Energy for resource reservation	Eq.11
E^u_{rr}	Uplink: Resource reservation energy	Eq.11
E^d_{sync}	Downlink: Energy for synchronization	0.33
E^d_{rr}	Downlink: Resource reservation energy	Eq.11
E^d_{rx}	Downlink: receive energy	Eq.17
P_l, P_I	Listening Power	0.1 W
P_I	Idle Power	0.2 W
P_t	Transmission Power	0.2 W
P_c	Power consumption in electronic circuits	0.01 W
L_W	Average computation latency of a miner	Eq.19
L_{DLT}	Total DLT latency	Eq.
L_{tM}	DLT average transmission latency	Eq.18
L_{sync}	Synchronization Latency	0.33s
$N_{r_{max}}$	Maximum number of attempts	10
P_{rr}	Probability of resource reservation	Eq.8

2 DLT-enabled IoT Data Trading Architecture and Protocols

This section presents the general system model of DLT-based IoT data trading as well as the data trading system with the three protocols tailored to different scenarios. Table C.1 summarizes the used notation.

2.1 DLT-enabled Data Trading via NB-IoT

The general architecture of DLT-based IoT data trading includes three main components: data providers (sellers $\mathcal{S}$), data consumers (buyers $\mathcal{B}$) and a distributed ledger, shown in Fig. C.1. Each seller or buyer can own one or more devices in the network. Here we assume that buyers and sellers act as digital wallets in a distributed network. During a trade denoted by $\mathcal{T}_i$, the seller $\mathcal{S}_i \in \mathcal{S}$ and buyer $\mathcal{B}_i \in \mathcal{B}$ communicate using the wide-area NB-IoT links. The trading procedure occurs to complete a deal between $\mathcal{S}_i$ and $\mathcal{B}_i$, exchanging data $\mathcal{D}_i \in \mathcal{D}$ and payment $\mathcal{P}_i$. First, $\mathcal{B}_i$ completes the payment $\mathcal{P}_i$ to $\mathcal{S}_i$ in reference to the requested data, $\mathcal{D}_i$, and $\mathcal{S}_i$ delivers $\mathcal{D}_i$ to $\mathcal{B}_i$ immediately. The general procedure from Fig.C.1 can be described as follows:

Buyer $\mathcal{B}_i$

Subscribes to the IoT data in distributed ledger generated and published by $\mathcal{S}_i$, and $\mathcal{B}_i$ makes a data request, b_i regarding its preferred data, $\mathcal{D}_i$. The b_i will be transmitted to $\mathcal{S}_i$ and recorded in the ledger via transaction $T_{i,add}$ for negotiation based on factors such as amount of data, quality of data, price, discount, etc. After choosing $\mathcal{D}_i$ from the list, $\mathcal{B}_i$ generates a transaction $T_{i,commit}$ which executes payment from B_i's wallet. Once $\mathcal{B}_i$ receives the $\mathcal{D}_i$ via $T_{i,settle}$, it will generate a confirmation back to ledger.

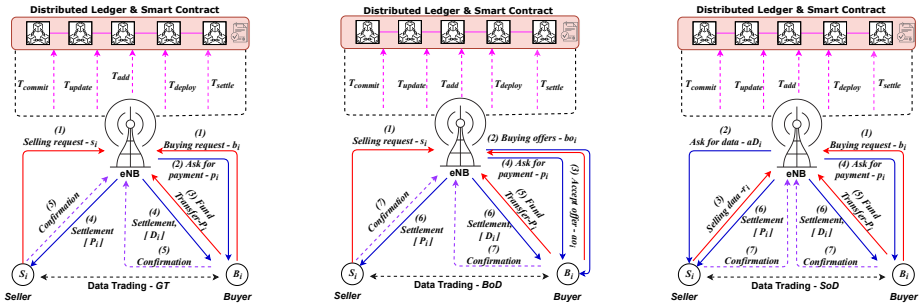
Seller $\mathcal{S}_i$

Has two main roles; to collect data from the environment (e.g., environmental sensing data, geographical data or data from surveillance systems) and to act as a hub gathering data from neighboring devices to sell on the market. $\mathcal{S}_i$ aims to earn the payment $\mathcal{P}_i$ from $\mathcal{B}_i$ by delivering $\mathcal{D}_i$ to $\mathcal{B}_i$. After publishing a hashed version of its data and prices to the market via $T_{i,add}$, $\mathcal{S}_i$ waits for buying requests. Based on the predefined rules in the smart contract system, upon receiving a request from $\mathcal{B}_i$ and the appearance of $T_{i,commit}$, generated by $\mathcal{B}_i$, the seller $\mathcal{S}_i$ can receive the payment $\mathcal{P}_i$. Finally, it confirms to the ledger that the trade $\mathcal{T}_i$ is complete.

Distributed Ledger

The DLT manages a distributed ledger to record all data trading history which is grouped into blocks and linked together chronologically. The deployed smart contracts autonomously control the order and automate payments from parties without the need of human interaction. The smart contracts guarantee trust, transparency and speed of exchanging information. These can be deployed based on the negotiation between data providers and customers via $T_{i,deploy}$. Any change in smart contracts (e.g. change of price, amount of data, or discount) can be performed via $T_{i,update}$.

In order to minimize the cost of storage, the sensing data could be hashed and recorded at more powerful DLT nodes, and only the hash of data is recorded to ledger. Then, a message



(a) *General Trading (GT)* **(b)** *Buying on Demand (BoD)* **(c)** *Selling on Demand (SoD)*

Fig. C.2: Three IoT data trading protocols: (a) *General Trading protocol (GT), (b) Buying on Demand (BoD)*, and (c) *Selling on Demand (SoD).*

is sent back to confirm that the data has been added to the ledger. After both $\mathcal{S}_i$ and $\mathcal{B}_i$ are satisfied with the terms of the contract, $T_{i,settle}$ is executed to get the payment $\mathcal{P}_i$ from $\mathcal{B}_i$ to transfer to $\mathcal{S}_i$'s wallet, while the data D_i is transmitted to the storage address of $\mathcal{B}_i$. We assume that the data services, (e.g., data storage, trading and task dispatching) are implemented on top of a permissionless Blockchain. The sensing data are formatted into normal transactions of fixed size. To enhance efficiency, only the digest of each transaction is stored on the chain, and the content of the transactions are stored by each consensus node off-chain or at the IPFS (InterPlanetary File System) storage.

2.2 IoT Data Trading Protocols

General Trading (GT)

The *GT* protocol procedure is shown in Fig. C.2a. In a trade $\mathcal{T}_i$, the buyer $\mathcal{B}_i$ sends a buying request r_i to market via transaction $T_{i,add}$ to express its need in specific data $\mathcal{D}_i$. After collecting sensing data, the data producer and seller $\mathcal{S}_i$, begins publishing its data information D_i, to the market. The smart contracts receive the requests from both customers $\mathcal{B}$ and producers $\mathcal{S}$ and then map the buying requests and selling requests to satisfy both parties based on their expected data and price. The buyer $\mathcal{B}_i$ commits to the request with a fund transfer via $T_{i,commit}$. After the smart contract receives the payment from $\mathcal{B}_i$, it executes $T_{i,settle}$ to transfer requested $\mathcal{D}_i$ to $\mathcal{B}_i$ and $\mathcal{P}_i$ to $\mathcal{S}_i$. Finally, both $\mathcal{B}_i$ and $\mathcal{S}_i$ confirm to the ledger that they have received $\mathcal{P}_i$ and $\mathcal{D}_i$, respectively.

A marketplace exchange of streaming IoT data, with a massive amount of data, requests, and a large number of parties, is an appropriate use case for the *GT* protocol. The environmental sensing data such as accurate real-time measurement data for control and alarm systems are exchanged between interested customers. More specifically, this protocol is used for the aforementioned use case due to its wide range of data continuously being pushed to the market. The open advertisement style of the *GT* protocol is appealing to potential buyers, encouraging the safe buying and selling of IoT data in a decentralized IoT data marketplace.

Buying on Demand (BoD)

BoD protocol describes a process where the producer $\mathcal{S}_i$ publishes data $\mathcal{D}_i$ to the market for selling via s_i request. The smart contract will broadcast information of received data from buying offer bo_i to other parties. For example, in a buying offer, $\mathcal{B}_i$ would ask whether others are interested in buying $\mathcal{D}_i$. If $\mathcal{B}_i$ is interested in D_i, it will accept the offer by generating $T_{i,add}$, and commit by $T_{i,commit}$ when the payment requests from smart contract is received. Then, the deal is settled as *GT* protocol via $T_{i,settle}$. The process of *BoD* protocol is described in Fig. C.2b.

Vehicle-to-Industry (V2I) emission trading, with a frequent exchange of data between vehicles and the vehicle industry, is an appropriate use case for the *BoD* protocol. In this scenario, the vehicles on the network act as the sellers of their emissions data, e.g., CO_2, NOx, while manufacturers (vehicle industry), **GoV** e.g air quality management department, and data analytic organizations act as the buyers for maintaining accurate, secure tamper-proof vehicular emissions data. In V2I, the data being exchanged are used for the purpose of creating a trusted life-cycle emission or fuel economy monitoring.

Selling on Demand (SoD)

The *SoD* protocol is described in Fig. C.2c. In this case, the smart contract receives the buying requests b_i from a customer $\mathcal{B}_i$, but there is no available appropriate data on the ledger to satisfy the requirements from $\mathcal{B}_i$. Hence, the smart contract sends an *ask-for-data* request aD_i to producers to ask whether they can provide the required data D_i. The providers $\mathcal{S}_i$ after a while can gather data from the environment or from other sources then answer to the market by s_i including D_i information as well as price P_i. Then, the smart contract asks $\mathcal{B}_i$ for fund transfer with an amount of P_i. The $\mathcal{B}_i$ make payment via $T_{i,commit}$. Then $T_{i,settle}$ are executed to complete the deal between $\mathcal{B}_i$ and $\mathcal{S}_i$. Finally, the confirmations are sent to the ledger from both parties.

This *SoD* protocol is beneficial, for example, when a party needs a type of data that is not available on the market and there is the need to trade in real time. In the scope of this study, we assume that, when a provider receives *ask-for-data* from a smart contract, it can provide the required data to the market. In real-life scenarios, some of the requests from customers cannot be satisfied immediately, so these requests are queued in the systems until the data is available. A Vehicle-to-Vehicle (V2V) use case is appropriate for this protocol where vehicles can purchase traffic information for a specific street which drivers expect to use in the near future. The vehicles that have the requested information can be traded with the buyers on the road. Finally, similar to V2I, V2V involves the continuous wireless exchange of IoT data collected from vehicle sensors. The V2V use case contributes in generating a life-cycle emissions or fuel economy monitoring system amongst vehicles. This form of communication helps to manage the safety of the road, as well as increase vehicle awareness.

2.3 Communication System Model

We consider an NB-IoT cell with eNB located in its center, including N devices uniformly distributed within the area. A data provider or consumer can consist of a single or multiple NB-IoT devices. For simplicity, we assume that each buyer or seller owns a single NB-IoT device to exchange assets and all devices belong to the normal coverage class. The DLT nodes

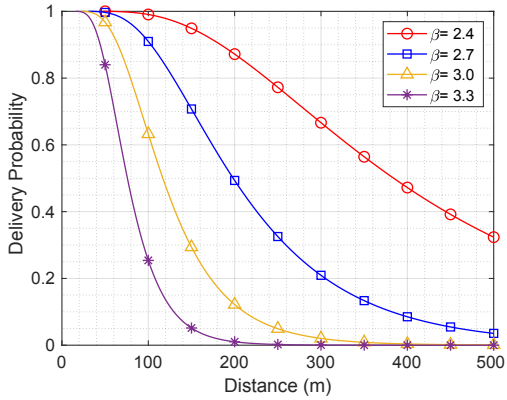


Fig. C.3: Delivery probability versus distance for a standard deviation σ_{dB} = 6 dBs.

are NB-IoT devices that have more computational power than seller/buyer nodes. In our model, involved sellers and buyers use NB-IoT as wireless network interfaces. In reality, the involved parties can use various wireless interfaces or networks for trading purposes but, our general model and analysis can be applied in these cases because of its modular and versatile design satisfies a broad range of interfaces and networks.

Our propagation model takes into account shadowing, but not small-scale fading; which is a sufficient first approximation as detailed physical layer modeling is not the focus of the work. Hence, for a given transmission power P_t and carrier frequency f, the received power at a distance d between the base station BS and sensor i is:

$$P_r(d) = 10\log_{10}\left[\frac{P_t G_t G_r\, c^2}{(4\pi f)^2 d^\beta}\right] + N(0, \sigma_{\text{dB}})\ \text{dB} \tag{C.1}$$

where G_t and G_r are the transmitter and receiver antenna gains, respectively, $c = 3 \cdot 10^8$ m/s is the speed of light, $N(0, \sigma_{\text{dB}})$ is a zero-mean Gaussian RV with standard deviation σ_{dB} dB, and β is the path loss exponent. From there, the outage probability at a given distance and receiver sensitivity $\gamma = 3.65 \cdot 10^{-10}$ W is:

$$p_{out} = 1 - Q\left[\frac{1}{\sigma_{\text{dB}}} 10\log_{10}\left(\frac{\gamma(4\pi f)^2 d^\beta}{P_t G_t G_r c^2}\right)\right] \tag{C.2}$$

Fig. C.3 demonstrates the delivery probability $p_d = 1 - p_{out}$ at varying distances, for four different β path loss exponent values, a standard deviation of $\sigma_{dB} = 6$ dBs. In this work, we choose $\beta = 2.7$ for urban area. We are aware that the model lacks a mobility aspect, however for this initial work, we have decided to use a simple model as previously described.

The arrival rate of uplink including selling and buying requests, respectively, to the system are: $\lambda_s = |\mathcal{S}|Tp_s p_d$ and $\lambda_b = |\mathcal{B}|Tp_b p_d$ in which T is number of communication sessions that an IoT device performs daily; p_s and p_b are probability a device request a selling service and a buying service, respectively. When an NB-IoT sensor device attempts to join the network, it first listens for the cell information, e.g, NPSS and NSSS messages to synchronize with the eNB. NB-IoT UEs have only two modes of operation, namely radio-resource control (RRC) idle and RRC connected [18]. In the former, the UEs can only receive the system information from the

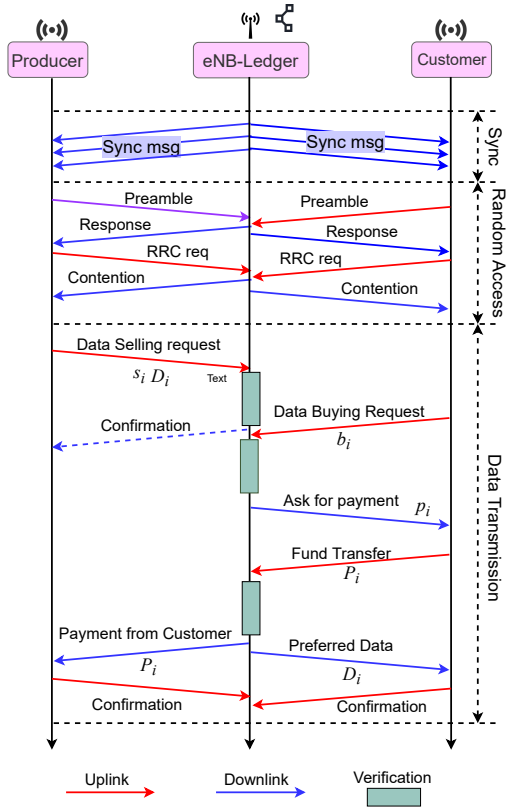


Fig. C.4: Communication diagram of the *GT* protocol.

BS and, only in the latter, data can be transmitted. UEs are in idle mode before initial access to the network, but may also enter this mode during power saving or after an explicit disconnection request. To transition from idle to connected mode, the UEs (clients) must first acquire the basic system information and synchronization as illustrated in the upper part of Fig. C.4. For this, the UE receives the master information block (MIB-NB) and the system information blocks 1 (SIB1-NB) and 2 (SIB2-NB). These are transmitted periodically through the downlink shared channel (DL-SCH) and carry the basic cell configuration, timing, and access parameters [19]. In addition, SIB1-NB carries the scheduling information for the rest of the SIBs.

After the system information has been acquired, the UEs must perform the RA procedure to transition to RRC connected mode. The RA procedure is a four-message handshake, initiated by the UEs by transmitting a single-tone frequency-hopping pattern, called preamble, through the NB Physical Random Access Channel (NPRACH). In most cases, the RA procedure is contention-based, hence, the preamble is chosen randomly from a predefined pool of up to 48 orthogonal sub-carrier frequencies. Consequently, the main reasons for an access failure are the lack of power in the transmission and simultaneous transmissions of the same preamble, which leads to collisions.

After completing the RA procedure, and if the control-plane (CP) cellular IoT (CIoT) is used, UEs may piggyback short UL data packets along with the RRC Connection Setup Com-

plete message. Otherwise, the non-access stratum (NAS) setup must be completed before eNB allocates resources for uplink transmission through the NB Physical UL Shared Channel (NPUSCH) and data can be transmitted. The resource unit (RU) is the basic unit for resource allocation in the NPUSCH and comprises a set of sub-frames in the time domain and sub-carriers in the frequency domain. The downlink (DL) data is transmitted through the NB physical DL shared channel (PDSCH). Data is exchanged based on the three defined trading protocols, *GT, BoD* and *SoD*. Fig. C.4 shows the physical operations of *GT* protocol as an example. *BoD* and *SoD* could be considered as extensions of the *GT* protocol, those protocols are especially beneficial when the data is not available in the market.

2.4 Performance metrics

Latency and the time required to complete a trade is one of the most important concerns of involved users. Latency directly influences the amount of time it takes for a trader to interact with the data market, the timely reception of relevant market information and the ability to act upon its receipt. The spread of the automatized data trading amplifies the impact of latency in terms of its competitive advantage. On top of this, IoT environments should be characterized with high energy efficiency. All these factors have motivated this investigation on the total E2E latency and energy consumption to complete a trade $\mathcal{T}_i$.

Latency

The latency to complete a trade $\mathcal{T}_i$ between seller and buyer are formulated as:

$$L_{\mathcal{T}_i} = L_{UD} + L_{DLT}, \tag{C.3}$$

where L_{UD} is the transmission latency between $\mathcal{S}_i$ and $\mathcal{B}_i$ which act as light nodes and full DLT nodes; While, L_{DLT} represents the DLT mining and synchronization latency. In detail, $L_{UD} = L^u + L^d$, where L^u, L^d are NB-IoT uplink and downlink latency, respectively; $L_{DLT} = L_v + L_{DLTsync}$, where L_v is block verification time at DLT nodes, and $L_{DLTsync}$ is synchronization time between DLT nodes via NB-IoT connectivity.

Total energy consumption

Similarly, the energy consumption model of a trade includes the energy consumption for uplink E^u, downlink E^d transmission between NB-IoT sensors with DLT full nodes, among DLT full nodes, and the energy consumed in verification process known as mining in DLT nodes E_{DLT}.

$$E_{\mathcal{T}_i} = E_{UD} + E_{DLT} \tag{C.4}$$

where E_{UD} and E_{DLT} are energy consumed by communication between sellers/buyers and DLT nodes and the energy performed among full DLT nodes, respectively. The transmission power and latency depend significantly on the physical deployment, such that we analyze both analyze the resource consumed in physical communication and the application layer. In next parts, we formulate the latency and energy consumption of each process.

3 Transmission Latency and Energy Consumption Models

As described in the previous section, the total E2E latency includes two parts, the latency of transmissions of uplink and downlink between buyers/sellers and DLT nodes, where latency occurs in the DLT verification process. For the first part, we define an adapted queuing model for DLT-based NB-IoT, based on the queuing model of the NB-IoT access network [20], the uplink and downlink radio resources are modeled as two servers which visit and serve their traffic queues in both directions.

End-to-End latency

The E2E latency of NB-IoT uplink and downlink can be formulated as:

$$L_{UeD} = L^u + L^d = L^u_{sync} + L^u_{rr} + L^u_{tx} + L^d_{sync} + L^d_{rr} + L^d_{rx} \tag{C.5}$$

where L^u_{sync}, L^u_{rr}, L^u_{tx}, L^d_{synch},L^d_{rr}, L^d_{rx} are energy consumption of synchronization, resource reservation, and data transmission of uplink and downlink, respectively. L^u_{sync} has been defined in [19] with the values of $0.33s$. L_{rr} is given as:

$$L_{rr} = \sum_{l=1}^{N_{r_{max}}} (1 - P_{rr})^{l-1} P_{rr} l (L_{ra} + L_{rar}) \tag{C.6}$$

in which $N_{r_{max}}$ is the maximum number of attempts, P_{rr} is the probability of successful resource reservation in an attempt, $L_{ra} = 0.5t + \tau$, is the expected latency in sending an RA control message, τ is the unit length and equal to the NPRACH period for the coverage class 1 which is varied from 40 ms to 2.56 s [19], and $L_{rar} = 0.5d + 0.5\mathcal{Q}fu + u$, is the expected latency in receiving the RAR message, where $\mathcal{Q}$ are requests waiting to be served.

In the following, we provide a simple technique based on *drift approximation* [21] to calculate P_{rr} recursively. Therefore, we treat the mean of the random variables involved in the process as constants. Besides, we assume that sufficient resources are available in the PDCCH so that failures only occur due to collisions in the PRACH or to link outages.

Let $\lambda^a = \lambda^u + \lambda^d$ be the arrival rate of access requests per PRACH period and $\lambda^a(l)$ be the mean number of devices participating in the contention with their l-th attempt. Note that in a steady state $\lambda^a(l)$ remains constant for all PRACH periods. Next, let $\lambda^a_{tot} = \sum_{l=1}^{N_{r_{max}}} \lambda^a(l)$ and that the collision probability in the PRACH can be calculated using the drift approximation for a given value of λ^a_{tot} and for a given number of available preambles K as:

$$P_{\text{collision}}(\lambda^a_{tot}) = 1 - \left(1 - \frac{1}{K}\right)^{\lambda^a_{tot}-1} \approx 1 - e^{-\frac{\lambda^a_{tot}}{K}}. \tag{C.7}$$

From there, we approximate the probability of resource reservation as a function of λ^a_{tot} as $P_{rr}(\lambda^a_{tot}) \approx p_d\, e^{-\frac{\lambda^a_{tot}}{K}}$. This allows us to define λ^a_{tot} as:

$$\lambda^a_{tot} = \lambda^a + (1 - P_{rr}(\lambda^a_{tot})) \sum_{l=2}^{N_{r_{max}}} \lambda^a(l) \tag{C.8}$$

since $\lambda^a(l) = (1 - P_{rr}(\lambda^a_{tot}))\,\lambda^a(l-1)$ for $l \geq 2$ and $\lambda^a(1) = \lambda^a$. Finally, from the initial conditions $\lambda^a(l) = 0$ for $l \geq 2$, the values of $\lambda^a(l)$ and λ^a_{tot} can be calculated recursively by: 1) applying (C.8); 2) calculating $P_{rr}(\lambda^a_{tot})$ for the new value of λ^a_{tot}; and 3) updating the values of $\lambda^a(l)$. This process is repeated until the values of the variables converge to a constant value. The final value of $P_{rr}(\lambda^a_{tot})$ is simply denoted as P_{rr} and used throughout the rest of the paper.

Assuming that the transmission time for the uplink transactions follows a general distribution with the first two moments l_1, l_2, then first two moments of the distribution of the packet transmission time are $s_1 = (f_1 l_1) / (\mathcal{R}w)$, and $s_2 = (f_1 l_2) / (\mathcal{R}^2 w^2)$. Applying the results from [22], considering L_{tx} as a function of scheduling of NPUSCH, we have:

$$L_{tx} = \frac{f\lambda^u s_1 s_2}{2s_1(1 - fGs_1)} + \frac{f\lambda^u s_1^2}{2(1 - f\lambda^u s_1)} + \frac{l_1}{\mathcal{R}^u w} \tag{C.9}$$

where $\mathcal{R}^u$ is the average uplink transmission rate, $\lambda^u = \lambda_s + \lambda_b$, and $f(\lambda_s + \lambda_b)s_1$ is the mean batch-size. The latency of data reception is defined as:

$$L_{rx} = \frac{0.5Fh_1 t^{-1}}{h_1(1 - Fht^{-1})} + \frac{Fh_1}{1 - Fht^{-1}} + \frac{m_2}{\mathcal{R}^d y} \tag{C.10}$$

in which, $h_1 = fm_1(\mathcal{R}^d y)^{-1}$, $h_2 = fh_2^2 m_2((\mathcal{R}^d)^2 y^2)^{-1}$ are two moments of distribution of the packet transmission time, assuming that Assuming that packet length follows a general distribution with moments m_1, m_2, $F = f\lambda^d t$, $\mathcal{R}^d$ is downlink data transmission rate.

Energy consumption

The energy consumption of the protocol 1 are formulated as below:

$$E_{UD} = E^u + E^d = E^u_{sync} + E^u_{rr} + E^u_{tx} + E^u_s + E^d_{sync} + E^d_{rr} + E^d_{rx} + E^d_s \tag{C.11}$$

In which E^u_{sync},E^u_{rr}, E^u_{rr}, E^d_{sync},E^d_{rr}, E^d_{rx} are energy consumption of synchronization, resource reservation, and data transmission of uplink and downlink, respectively. We have:

$$E_{sync} = P_l \cdot L_{sync} \tag{C.12}$$

$$E_{rar} = P_l \cdot L_{rar} \tag{C.13}$$

$$E_{rr} = \sum_{l=1}^{N_{max}} (1 - P_{rr})^{l-1} \cdot P_{rr} \cdot (E_{ra} + E_{rar}) \tag{C.14}$$

$$E_{ra} = (L_{ra} - \tau) \cdot P_I + \tau \cdot (P_c + P_e P_t) \tag{C.15}$$

$$E_{tx} = (L_{tx} - \frac{l_a}{\mathcal{R}^u w}) \cdot P_I + (P_c + P_e P_t)\frac{l_a}{\mathcal{R}^u w} \tag{C.16}$$

$$E_{rx} = (L_{rx} - \frac{m_1}{\mathcal{R}^d y}) \cdot P_I + P_l \frac{m_1}{\mathcal{R}^d y} \tag{C.17}$$

in which, P_e, P_I, P_c, P_l, and P_t are the power amplifier efficiency, idle power consumption, circuit power consumption of transmission, listening power consumption, and transmit power consumption, respectively.

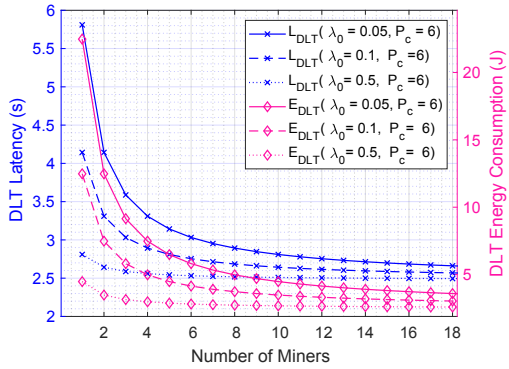


Fig. C.5: DLT performance in latency and energy consumption

4 Resource consumption model of DLT verification process

4.1 System Model

Consider a DLT network that includes M miners. These miners start their Proof-of-work (PoW) computation at the same time and keep executing the PoW process until one of the miners completes the computational task by finding the desired hash value [6]. When a miner executes the computational task for the POW of current block, the time period required to complete this PoW can be formulated as an exponential random variable W whose distribution is $f_W(w) = \lambda_c e^{-\lambda_c w}$, in which $\lambda_c = \lambda_0 P_c$ represents the computing speed of a miner, P_c is power consumption for computation of a miner, and λ_0 is a constant scaling factor. Once a miner completes its PoW, it will broadcast messages to other miners, so that other miners can stop their PoW and synchronize the new block.

$$L_{tM} = L_{newB} + L_{getB} + L_{transB} \tag{C.18}$$

In (18), L_{newB}, L_{getB}, and L_{transB}, are latencies of sending hash of new mined block, requesting new block from neighboring nodes, and new block transmission, respectively. L_{newB} and L_{transB} are computed using uplink transmission, while L_{getB} is computed based on downlink transmission as described in previous section.

For the PoW computation, a miner $i*$, first finds out the desired PoW hash value, $i* = \min_{i \in M} w_i$. The fastest PoW computation among miners is W_{i*}, the complementary cumulative probability distribution of W_{i*} could be computed as $Pr(W_{i*} > x) = Pr(\min_{i \in M}(W_i) > x) = \prod_{i=1}^{H} Pr(W_i > x) = (1 - Pr(W < x))^M$. Hence, the average computational latency of miner $i*$ is described as:

$$L_{W_{i*}} = \int_0^\infty (1 - Pr(W \leq x))^M x = \int_0^\infty e^{-\lambda_c M x} x = \frac{1}{\lambda_c M} \tag{C.19}$$

The total latency required from DLT verification process is $L_{DLT} = L_{tm} + L_{W_{i*}}$. The average energy consumption of DLT to finish a single PoW round is:

$$E_{DLT} = P_c L_{W_{i*}} + P_t L_{tm} \tag{C.20}$$

Table C.2: Comparison in Smart Contract Execution Cost

Protocols	Operations	Ether$\cdot 10^{-4}$	Gas Cost	Approx. USD
GT	Deploy SC	≈1.2	1132443	0.2862
BoD	Deploy SC	≈1.3	1268369	0.2879
SoD	Deploy SC	≈1.4	1582349	0.3783

* 1 Ether = 10^9 Gwei; 1 USD = 4,182,471.9949 Gwei

The performance of DLT system is shown in Fig. C.5. The figure demonstrates that the energy consumed and latency by DLT nodes are reduced with the number of miners. Contrarily, as the number of miners increase, this leads to a higher probability that miners verify transactions, and the mining speeds increase as well.

4.2 Analysis of data trading protocols

In this section, the E2E latency and energy consumption of three protocols are formulated and compared approximately. The resource consumed by each data trading protocol is separated into two parts, namely, 1) the connectivity between $\mathcal{S}_i$ and $\mathcal{B}_i$ acting as light nodes in DLT network with full nodes, and 2) the communication among DLT full nodes.

The E2E latency of trade $\mathcal{T}_i$ using *GT* protocol including the transmission latency between $\mathcal{B}_i$, $\mathcal{S}_i$ and DLT verification nodes is described as below:

$$L_{\mathcal{T}_i}^{P1} = L_{UD}^{P1} + L_{DLT}^{P1} = L^{u,P1} + L^{d,P1} + L_{DLT}^{P1} \tag{C.21}$$

$$E_{\mathcal{T}_i}^{P1} = E_{UD}^{P1} + E_{DLT}^{P1} = E^{u,P1} + E^{d,P1} + E_{DLT}^{P1} \tag{C.22}$$

Assuming that $L_{sync}^{u,P1} = L_{sync}^{d,P1} = 0.33$ s, $L_{rr}^{u,P1}$ and $L_{rr}^{d,P1}$ are computed as (7), $L_{tx}^{u,P1}$ and $L_{rx}^{d,P1}$ are calculated based on (8) and (9) with the defined packet length of uplink and downlink.

Then, the battery lifetime of an NB-IoT device can be computed as below:

$$BTL = E_0 \left[Tp^u(E^u) + Tp^d E^d \right]^{-1}, \tag{C.23}$$

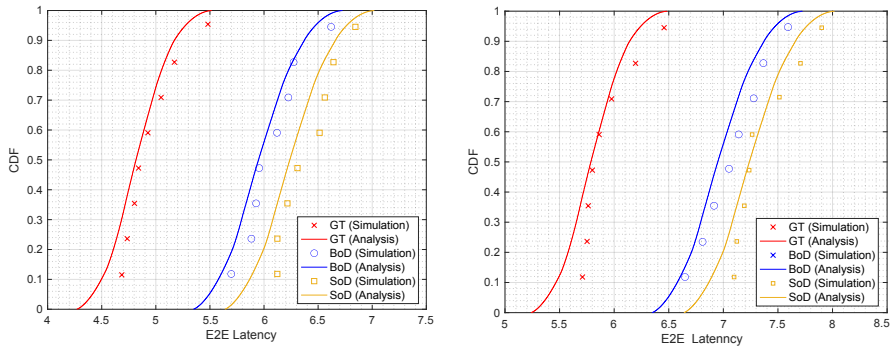
where E_0 is the energy storage on the device battery. Similarly, the performance of *BoD* protocol and *SoD* protocol can be formulated as *GT* protocol.

5 Performance Evaluation

In this section, we will introduce the settings in terms of simulations and experiments. Then, we analyze the performance of proposed trading protocols in terms of latency and battery lifetime.

5.1 Experiment Settings

In this section, we evaluate the derived data trading model, compare and analyze the designed trading protocols. In order to evaluate the derived model and compare the three proposed protocols, we setup a network with $N = 10000$ NB-IoT devices, where devices randomly play roles



(a) M = 20

(b) M = 5.

Fig. C.6: Impact of number of DLT miners to latency of trading strategies.

as sellers or buyers. We validate the results via Monte Carlo Simulations, where we run 1000 realizations for each trading protocol and experiment. The buyer nodes and seller nodes generate requests following a Poisson distribution process with rates of λ_b and λ_s, respectively. The number of buying and selling $\mathcal{T}_i$ requests per day varied from 1 to 20, $\mathcal{T}_i = [1, 20]$. Additionally, the number of buyer and seller nodes varied and remained less than N. The transmission power in the experiments are denoted as $P_t = 0.2$W, $E_0 = 1000$. The number of DLT miners are up to 20 miners at maximum, $M = [1, 20]$.

5.2 Cost of Smart Contracts

The proof of concept for the three proposed trading protocols are deployed in Ganache[2] Ethereum network to evaluate the complexity and the cost of execution of different trading strategies. The smart contracts are implemented and deployed using Remix IDE[3]. In the Ethereum platform, any operation or transaction execution that changes the Blockchain or its state requires that the involved parties pay a fee called *gas*. The gas terminology in Ethereum charges the execution of every operation to guarantee that smart contracts running in Ethereum Virtual Machine (EVM) [23] will be eventually terminated. These costs are calculated by using the amount of gas executed and the unit of gas price. The gas required during any activity reflects the computational complexity or size of the smart contracts, while the gas prices are determined by the Ethereum miners in the network. Each operation or execution on the EVM charges a certain amount of gas and not all transactions are created cost equally. In this work, we used Gwei[4] to evaluate the cost of different operations in the trading process.

Table C.2 shows the cost of the three protocol deployment and transaction costs to complete a deal between a seller and a buyer. We observe that the approximate cost in USD for *GT* is the cheapest in comparison to *BoD* and *SoD* protocols. The cost of smart contract execution is generally expensive, therefore, it is preferred to use the *GT* protocol. In an environment with a massive number of involved parties and transactions (e.g, marketplace), the transactions are

[2]https://www.trufflesuite.com/ganache
[3]https://remix.ethereum.org/
[4]https://www.cryps.info/

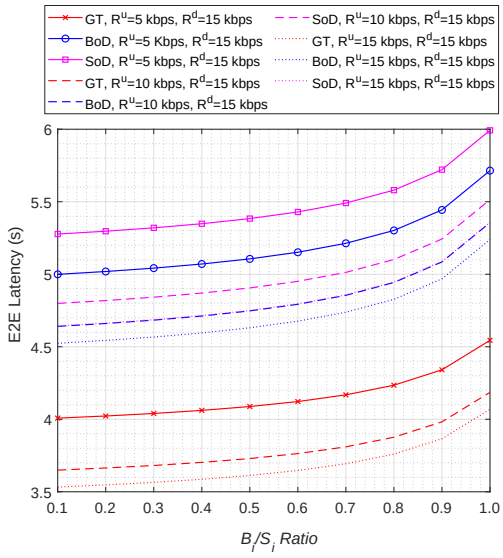


Fig. C.7: Impact of B_i/S_i ratio

executed autonomously to reduce costs using the available resources. While, *BoD* and *SoD* are preferred when the users have requests with specific resources.

5.3 Latency to complete a deal

Impact of number of Miners

Fig. C.6 shows the latency of three trading protocols. Both the analysis and the simulation results show that the *SoD* protocol has higher latency to complete a deal between S_i and B_i because of extra steps. Note that the comparison is evaluated approximately because the latencies depend on various factors such as the number of DLT miners, the length of blocks, and level of difficulty. The verification latency of DLT miners is measured based on the Ganache Ethereum network. In *GT* protocol, the smart contracts map selling requests r_i with available D_i stored in the ledger and make a deal between S_i and B_i immediately, so that it guarantees efficient trading in the market. The average latency to complete a deal of *GT* protocol is around 4.5 seconds including latency of NB-IoT and DLT procedures. The *BoD* and *SoD* latencies are higher because of extra procedures necessary to gather required information between customers and producers. We observe that *GT* could be used in terms of applications which require low latency. The downside of *GT* protocol is that the data requests must always be available to settle the trade, so that it is matched with applications (e.g smart metering) where the type of information is fixed.

Seller and Buyer Ratio

The impact of ratio between the number of sellers and buyers are demonstrated in Fig. C.7. The figure also shows a comparison between trading protocols under varying NB-IoT uplink

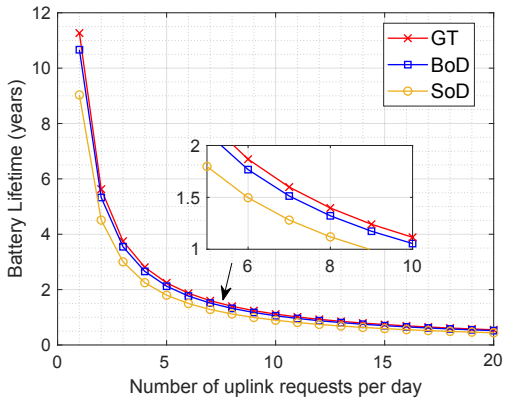


Fig. C.8: NB-IoT Battery lifetime.

transmission rate, $R^u = \{5, 10, 15\}$ Kbps and fixed downlink data rate at $R^d = 15$ Kbps. The results show that i) the increase in the number of buyers requires more delay to complete a trade, and ii) in contracts, increasing data rates help to provide a faster service.

5.4 Battery lifetime of NB-IoT devices

In general, the power consumption of battery lifetime during a reporting period depends on length of data transmitting, bandwidth, MCL, latency, and RF module. Hence, the power consumption of one trading protocol will be higher or lower than the other depending on the values of these parameters. The battery lifetime capabilities of NB-IoT devices among three trading protocols are compared and demonstrated in Fig. C.8. The number of uplink requests are varied from 1 to 20 requests per day. We observe that the number of requests per day significantly impacts to the battery lifetime of NB-IoT devices. In fact, the battery lifetime of around 10 years can be achieved with one report per day, however, for more frequent transmissions (e.g. 8 requests per day) the battery lifetime is reduced to around 1 year. Specifically, the *GT* trading protocol achieves over 11 years for 1 report per day, while *BoD* and *SoD* achieve around 10 years and 9 years, respectively. The fact is that applications such as smart metering, smart parking using NB-IoT connectivity do not require frequent updates from sensors. In terms of increasing number of requests daily up to 5, the battery lifetime is reduced significantly to around 2 years. Because for each buying or selling request, the NB-IoT devices start running protocol with multiple operation until the trade is settled.

6 Conclusion

In this paper, we proposed the first benchmarking framework for evaluating data trading protocols. The framework includes a model and analysis of systematic DLT-based IoT data smart trading protocols in massive NB-IoT deployments. We have proposed and analyzed three IoT data trading protocols named *General Trading*, *Buying on Demand*, and *Selling on Demand*. Considered collectively, these protocols cover a wide range of interesting scenarios, such as

carbon emission trading or monitoring of vehicle emissions. We have conducted a comprehensive analysis of these protocols in terms of communication and evaluated end-to-end latency, battery lifetime, and resource consumption. In terms of performance, each protocol is tailored to a different scenario. We conclude that the *GT* protocol should be used as primary protocol in a data marketplace where massive amounts of data are available. Additionally, the *BoD* and *SoD* protocols can be interchangeably used when there are particular demands from either buyers or sellers.

To the best of our knowledge, this is the first work of its kind, providing a general benchmark framework for data trading protocols in IoT environments. In the next iteration of this work, we will first consider more elaborate utility models for the parties involved in trading. Second, we will evaluate the performance of trading schemes in diverse network interfaces and real-life networks.

7 Acknowledgment

This work was supported in part by the European Research Council (Horizon 2020 ERC Consolidator) under Grant 648382 WILLOW; in part by the European Union's Horizon 2020 Program under Grant 957218 IntellIoT; in part by the Independent Research Fund Denmark (DFF) under Grant 8022-00284B (SEMIOTIC) and Grant 9165-00001B (GROW); and in part by the National Science Foundation Graduate Research Fellowship under Grant DGE-1839285.

References

[1] I. Report, "The growth in connected iot devices is expected to generate 79.4zb of data in 2025, according to a new idc forecast," [Online]https://www.idc.com, 2019, (Accessed on 12/04/2020).

[2] Z. Huang, X. Su, Y. Zhang, C. Shi, H. Zhang, and L. Xie, "A decentralized solution for iot data trusted exchange based-on blockchain," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 1180–1184.

[3] C. Perera, "Sensing as a service (s2aas): Buying and selling iot data," *arXiv preprint arXiv:1702.02380*, 2017.

[4] W. Mao, Z. Zheng, and F. Wu, "Pricing for revenue maximization in iot data markets: An information design perspective," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1837–1845.

[5] S. Jernigan, D. Kiron, and S. Ransbotham, "Data sharing and analytics are driving success with iot," *MIT Sloan Management Review*, vol. 58, no. 1, 2016.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2008.

[7] P. Danzi, A. E. Kalor, R. B. Sorensen, A. K. Hagelskjær, L. D. Nguyen, C. Stefanovic, and P. Popovski, "Communication aspects of the integration of wireless iot devices with distributed ledger technology," *IEEE Network*, vol. 34, no. 1, pp. 47–53, 2020.

[8] D. L. Nguyen, I. Leyva-Mayorga, and P. Popovski, "Witness-based approach for scaling distributed ledgers to massive iot scenarios," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–6.

[9] L. D. Nguyen, A. E. Kalor, I. Leyva-Mayorga, and P. Popovski, "Trusted wireless monitoring based on distributed ledgers over nb-iot connectivity," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 77–83, 2020.

[10] P. Gupta, S. Kanhere, and R. Jurdak, "A decentralized iot data marketplace," *arXiv preprint arXiv:1906.01799*, 2019.

[11] S. Bajoudah, C. Dong, and P. Missier, "Toward a decentralized, trust-less marketplace for brokered iot data trading using blockchain," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 339–346.

[12] P. Missier, S. Bajoudah, A. Capossele, A. Gaglione, and M. Nati, "Mind my value: a decentralized infrastructure for fair and trusted iot data trading," in *Proceedings of the Seventh International Conference on the Internet of Things*, 2017, pp. 1–8.

[13] W. Xiong and L. Xiong, "Smart contract based data trading mode using blockchain and machine learning," *IEEE Access*, vol. 7, pp. 102 331–102 344, 2019.

[14] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, "Sdte: A secure blockchain-based data trading ecosystem," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 725–737, 2019.

[15] S. Feng, W. Wang, D. Niyato, D. I. Kim, and P. Wang, "Competitive data trading in wireless-powered internet of things (iot) crowdsensing systems with blockchain," in *2018 IEEE International Conference on Communication Systems (ICCS)*. IEEE, 2018, pp. 289–394.

[16] S. Popli, R. K. Jha, and S. Jain, "A survey on energy efficient narrowband internet of things (nbiot): architecture, application and challenges," *IEEE Access*, vol. 7, pp. 16 739–16 776, 2018.

[17] T. ETSI, "Lte: Evolved universal terrestrial radio access (e-utra), physical layer procedures-corresponding to 3gpp ts36 213," *3GPP TS*, vol. 136, no. 213, p. V10.

[18] 3GPP, "Radio resource control (rrc); protocol specification," no. TS 36. 331 v9. 3.0, 2010.

[19] O. Liberg, M. Sundberg, E. Wang, J. Bergman, and J. Sachs, *Cellular Internet of things: technologies, standards, and performance*. Academic Press, 2017.

[20] A. Azari, Č. Stefanović, P. Popovski, and C. Cavdar, "On the latency-energy performance of nb-iot systems in providing wide-area iot connectivity," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 1, pp. 57–68, 2019.

[21] C.-H. Wei, G. Bianchi, and R.-G. Cheng, "Modeling and analysis of random access channels with bursty arrivals in ofdma wireless networks," *IEEE transactions on wireless communications*, vol. 14, no. 4, pp. 1940–1953, 2014.

[22] H. Akimaru and K. Kawashima, *Teletraffic: theory and applications*. Springer Science & Business Media, 2012.

[23] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

Paper D

A Marketplace for Trading AI Models based on Blockchain and Incentives for IoT Data

Authors:
Duc-Lam Nguyen, Shashi Raj Pandey, Soret Beatriz, Arne Broering,
and Petar Popovski

The paper has been submitted for publication
IEEE Internet of Thing Journal Vol. XX(X), pp. XXX–XXX, 201X.

Abstract

As Machine Learning (ML) models are becoming increasingly complex, one of the central challenges is their deployment at scale, such that companies and organizations can create value through Artificial Intelligence (AI). An emerging paradigm in ML is a federated approach where the learning model is delivered to a group of heterogeneous agents partially, allowing agents to train the model locally with their own data. However, the problem of valuation of models, as well the questions of incentives for collaborative training and trading of data/models, have received a limited treatment in the literature. In this paper, a new ecosystem of ML model trading over a trusted Blockchain-based network is proposed. The buyer can acquire the model of interest from the ML market, and interested sellers spend local computations on their data to enhance that model's quality. In order to factor in the individual contribution of the local data to the training of the model, we introduce the modified distributed Data Shapley Value (DSV), namely Approximate Federated Shapley Value *(`AFS`). At the same time, the trustworthiness of the entire trading process is provided by the Distributed Ledger Technology (DLT). Extensive experimental evaluation of the proposed approach shows a competitive run-time performance, with a 15% drop in the cost of execution, and fairness in terms of incentives for the participants.*

1 Introduction

1.1 Context, motivation and challenges

Personal IoT devices keep generating an enormous amount of sensing data that is expected to reach 79.4 Zettabytes (ZB) globally in 2025 [1]. Several attempts to enhance and adapt business workflows have been made towards exploiting the provision of IoT data [2, 3]. In this regard, training machine learning models and data sharing are two popular uses of IoT data. Furthermore, emerging diverse platforms for accessing and sharing IoT data connects various distributed IoT devices/data sources, thereby facilitating suppliers to exchange their data [4]. For example, in IoT systems for air quality monitoring and emission control, Air Quality Index (AQI) is a quantity defined to estimate the degree of severity for air pollution and CO2 emission levels. AQI quantifies the concentration of various particles in the air, such as PM2.5 or PM5.0, using state-of-the-art sensor devices [5]. There are two most popular measurement methods for AQI: i) sensing-based [6], and ii) vision-based [7]. In the sensing-based method, the IoT sensor devices are delivered around the area interest, e.g., city, urban, to collect the quality of the air and emission levels. These measurements are then forwarded to the central server for further analysis and calculation of AQI. In the vision-based method, the devices with an embedded camera, such as a camera station in the road, or individual mobile phones, can take photos of a specific area and send them to the server. The server then applies advanced image processing techniques on these images to derive the analysis report of air quality. However, both methods have problems due to (i) high energy consumption for collecting data and transmission, (ii) requirement for a large dataset for a high quality AQI estimation, (iii) the server acting as a single point of failure, and (iv) data privacy concerns under General Protection Regulation (GDPR). Several recent works have addressed issues related to (i) using efficient resource management

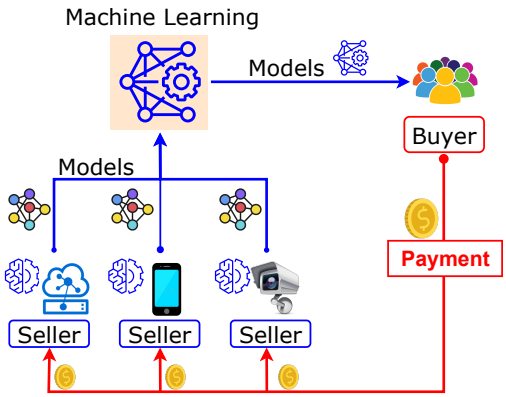


Fig. D.1: A motivation example: IoT devices contribute to train an ML model for the buyer to predict CO2 emission levels and get incentives from their contributions.

techniques and (ii) with dense sensory networks [8]. However, the primary concerns about (iii) and (iv) remain a single point of failure network topology and data privacy protection. They are yet to be addressed in an efficient manner.

In addition, the regulations such as EU's GDPR, California's Consumer Privacy Act (CCPA), and China's Cyber Security Law (CSL) [9] limits the reckless use/collection of personal information and fosters data privacy. Hence, a feasible joint solution to address challenges raised in (iii) and (iv) is imperative for optimized operation of the market for data exchange.

In this regard, more recently, Federated Learning (FL) has been considered a key solution to address the privacy issue in training learning models [10]. FL is a distributed model training paradigm that aims to solve the challenges of data governance and privacy by training algorithms collaboratively rather than transferring the data itself.

For example, in a typical FL setting, at first, as shown in Fig. D.1, the IoT devices collect the pollution and CO2 emission levels and store them in their local database. Consider an interested organization or individual, termed a *buyer*, willing to train an ML model to predict a specific area's emission level. However, they do not possess sufficiently large datasets about sensing information or image data. In such a scenario, they can send their initial model to a model marketplace to find appropriate parties interested in contributing to the model training process. Then, the IoT devices, termed *sellers*, can download the initial model and train it using their local data. After that, the IoT devices can share the updated model weights to the marketplace, where there is an aggregator to aggregate submitted local models to build global models. Based on the aggregated global models, the model buyer can use the global model to predict the emission levels with acceptable precision. In this manner, using the FL approach, we can address the problem of data privacy where data is locally trained without the need to transmit to a central server. However, from a systems perspective, a shared IT environment, such as an aggregator in the marketplace, may become a single point of failure in terms of data integrity, trust, security, and transparency [11]. A conventional data market is often deployed as a centralized service platform that gathers and sells raw or processed data from data owners (e.g., the trained learning models) to the consumers [12] [13]. This leads to two important concerns. *First*, this strategy exposes the platform as a single point of security risk; the malfunctioning

platform servers has serious security concerns including data leakage, inaccurate calculations results, and manipulation of data price. *Second*, collaboration for model training raises questions in terms of how to motivate participants to participate in such an ML training endeavor. The incentive for each IoT client based on their contribution should be fair and transparent. These features are not present in a standard FL setup, as in many applications there is no clear and natural incentive mechanism for involved participants to provide quality information. This calls for a carefully designed mechanisms to reward parties economically and thus incentivize participation [14, 15]. For example, a fixed price per data point could motivate participants to collect massive amounts of low quality or fake data if there is no intermediary process to check quality of training data. Besides, another reason that may disincentives parties from sharing data could stem from privacy and integrity concerns regarding the use of participant's data once it is shared. For instance, the sellers can re-use the data which has already been sold.

The aforementioned challenges can be effectively handled by a Distributed Ledger Technologies (DLTs).[1] DLTs and Blockchains enable untrusted parties to share information in an immutable and transparent manner [16]. Outside of its key role in financial transactions, the applications of DLTs can be seen as a key enabler for trusted and reliable distributed IoT systems, e.g., a distributed IoT data marketplace. For instance, in a Blockchain-enabled IoT data marketplace [17], Blockchain transactions include IoT sensing data, or system control messages, and these are recorded and synchronized in a distributed manner in all the involved participants of the network [18]. Furthermore, DLTs enable the preservation of all transactions in immutable records, with each record being spread across several participants. Thus, the decentralized nature of DLTs ensures security, as does the use of robust public-key encryption and cryptographic hashes. The advantages of incorporating DLTs into trading ML models in IoT systems include: i) ensuring immutability and transparency for historical ML model trading records, ii) eliminating the need for third parties, and iii) developing a transparent system for AI model trading in heterogeneous networks to prevent tampering and injection of fake data from the stakeholders, according to [19, 20]. With the wide spread of ubiquitous marketplaces recently, it became relevant to investigate the use of ML model trading in marketplace environments.

With the aforementioned motivation, we propose a Blockchain-based model trading system which enables a secured and trusted marketplace to collaboratively train ML models as well as guarantees fair incentives for every participants and privacy of data. Based on the quality of the uploaded models, which is quantified by using a distributed Data Shapley Value (DSV), the participants[2] can get the incentive based on the updated models, for example, as tokens or fiats. Note that based on our proposed system, the parties do not need to share their local data, but only provide customized models or query interface to the marketplace. Consequently, the proposed system allows multiple participants to jointly train the ML models on the marketplace based on their own training data. Buyers who need to train their ML model will pay to the market for the improvement of their model, and sellers who sell their contribution to train the ML models will get paid by the market via smart contracts.

The main features of the proposed model trading are:

1. **Trusted and transparent transactions**: The DLT is considered a trusted, tamper-proof, and transparent system in which the participants can check and follow the progress of a

[1] In this work, the terms *Blockchain* and DLT are used interchangeably. Blockchains are a type of DLT, where nodes maintain a copy of the ledger having embedded chains of blocks. These blocks are basically composed of digital pieces of information, particularly defined as *transactions*.

[2] The terms "*participants*","*clients*" and "*agents*" in this work are used to refer to "*IoT devices*".

training task. Based on that, the model is exchanged and traded securely and transparently.

2. **Valuation of Data:** The local models contributed by the trainers (service sellers) are collected and evaluated via Shapley Value (SV) extension to approximately estimate the quality of the models.

3. **Fair Payment:** The participants receive their reward, which is proportional to the usefulness of their data in improving the models. The distributed incentive mechanism for FL based on SV measures participants' contributions in the marketplace.

1.2 Contributions and Paper Organization

In this paper, we develop a marketplace for trading ML model where we leverage the attributes of Blockchain network and unleash a tamper-proof, fair sharing of offered incentives between the participants, particularly, based on the marginal contribution of their data for improving the trained model. In the following, we summarized the major contributions of this work.

- **ML Model Marketplace**: We develop a Blockchain-based model trading system that allows participants to purchase learning models and sell contributions in training them. The system records the trading details in a tamper-proof distributed ledger, similar to [17]. However, differently from [17], where only the system-level parameters concerning computation and communication efficiency for IoT data trading are analyzed, here we design mechanisms to factor in the individual contribution of data and the corresponding incentive design to realize an ML model marketplace.

- **Federated Data Shapley Value (SV):** We propose the use of data SV to estimate the valuation of participants' data for the developed system and evaluate their contribution to the model during local training. We show that the standard SV value is inefficient for distributed ML, and hence, design and deploy an extension of standard SV, namely *Approximate Federated Shapley Value* (`AFS`), for our platform. The method is robust and allows plugging any developed mapping functions related to the device's local data into the proposed distributed Shapley mechanism for value quantification. As a result, one can design a contribution-based, efficient incentive mechanism to stimulate model trading.

- **Fair Incentive Design**: We design a fair incentive mechanism that ensures the amount of tokens gets distributed amongst the participants as per their contribution in improving the model performance. In doing so, we have conducted extensive simulations and experiments, demonstrating that the proposed approach shows a competitive run-time performance, with a 15% drop in the cost of execution and fairness in terms of incentives for the participants.

The rest of the article is organized as follows. Section II, presents the concepts of DLTs, FL, as well as definition of data valuation schemes used in this paper. This is followed by description of the system model of the marketplace for ML model trading and explain in detail how the system works. In Section III, the value of ML models is calculated using the `AFS`. Section IV contains description of the testbed and experimental results. Section V discusses related works and finally, Section VI concludes the paper.

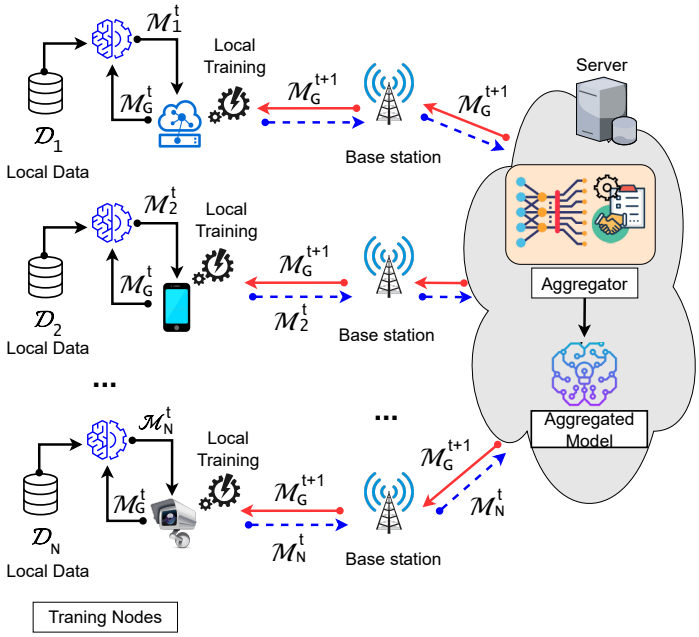


Fig. D.2: Standard FL mechanism. It raises problems on "single point of failure" and transparency of client's contributions.

2 Preliminaries

2.1 Standard FL

FL is a distributed machine learning setting in which numerous entities (clients) cooperate on training a learning model without disclosing their available raw data [10]. Instead, clients distributively perform computations on their data and transfer obtained local learning parameters updates to the server for aggregation process. The aggregated model, i.e., the global model, is broadcast back to the clients for the next round of local computations resulting in the local learning parameters. The interaction between the server and clients to solve the learning problem continues until an acceptable level of model accuracy is achieved [21]. In this manner, FL offers (i) privacy-preserving benefits in the model training approach by not requiring clients to share their local data to the server, and consequently, (ii) lower communication overhead by offering distributed model training paradigm and exchange of model parameters only. Therein, FL enables training ML models at edge networks

Fundamentally, there exists two main actors in the FL system: (i) the data owners, often termed as participants, and (ii) the model owner, which is the FL server. Consider a set of N data owners, defined as $\mathcal{N} = \{1, 2, \ldots, N\}$, where each of them has a private dataset $\mathcal{D}_{i \in \mathcal{N}}$ of size D_i. In Table D.1, we provide the summary of key notations used in this paper. Each data owner i trains a local model $\mathcal{M}_i$ using its dataset $\mathcal{D}_i$ and sends only the obtained local model parameters to the FL server. Then, the FL server aggregates all the collected local models to build a global model, $\mathcal{M}_G = \sum_{i \in \mathcal{N}} \mathcal{M}_i$. This is where, in principle, the FL approach

Table D.1: Summary of key notations.

Notation	Meaning
MI_i	DLT miner i
$\mathcal{N}$	Set of N participants (clients)
$\mathcal{D}_i$	Private local dataset of user $i \in \mathcal{N}$
D_i	Size of local dataset of user i
$\mathcal{M}_i$	Local model of user i
$\mathcal{M}_G$	Global model aggregated at DLTs
$\mathcal{M}_G^0$	Initial global model at DLTs
$L(\cdot)$	Loss function
$\nabla L(\cdot)$	Gradient of the loss function
η	Learning rate
S_i	Model trainer (or Seller) i
B_i	Model owner (or Buyer) i
$\mathcal{P}_d$	Deposit from buyer
$\mathbf{B}$	Training batch size
$\mathcal{A}$	Training algorithm
$U(\cdot)$	Utility function
ϕ_i	Valuation of data contributor i
E	Number of local epochs for model training
T	Number of training interactions
$\mathcal{T}_i$	A trade deal between S_i and B_i
$\widetilde{\mathcal{M}}$	Approximated model

differs from the traditional centralized training where $\mathcal{D} = \cup_{i \in \mathcal{N}} \mathcal{D}_i$ is used to train a model $\mathcal{M}^T$, i.e., data first gets aggregated centrally before the actual model training happens. In Fig. D.2, we show a standard architecture and an overview mechanism of the FL training process. We assume that the data owners are honest, i.e., actual private data will be used for the local training, and correspondingly, the FL server will receive accurate local models from the data owners. Following to which, the workflow of standard FL can be described as below.

First, considering the target application, the server decides the training task and defines the corresponding data requirements. Furthermore, the server also specifies the hyper-parameters of the global model and the training process, e.g., the learning rate η. The server then broadcasts the initialized global model $\mathcal{M}_G^0$ and the learning task to a subset of selected participants. Next, based on the global model $\mathcal{M}_G^t$, where t denotes the current global iteration index, i.e., the communication rounds between the participants and the server, each participant uses its local data to update their model parameters $\mathcal{M}_i^t$. In doing so, during iteration t, the participant $i \in \mathcal{N}$ aims at finding the optimal parameters $\mathcal{M}_i^t$ that minimize the local loss problem $L(\mathcal{M}_i^t)$, defined as the finite-sum of empirical risk functions as [10, 22]:

$$\mathcal{M}_i^t = \arg\min_{\mathcal{M}_i^t} L(\mathcal{M}_i^t). \tag{D.1}$$

Each participant can solve (D.1) using well-known stochastic gradient descent (SGD) algorithm; we formally call this procedure that solves (D.1) as *local iteration*. Note that the FL process

Algorithm 1: Federated Averaging (FedAvg) Algorithm

Input: Local mini-batch size **B**, number of participants per interaction m, number of global interactions T, number of local epochs E, and learning rate η.
Output: Global Model $\mathcal{M}_G$.

Initialize: $\mathcal{M}_G^0$.
for *each interaction* $t = \{0, 1, 2, 3, \ldots, T-1\}$ **do**
  $\mathcal{S}_t \leftarrow$ (random set of m clients);
  for *each participant* $i \in \mathcal{S}_t$ **in parallel do**
    $\mathcal{M}_i^{t+1} \leftarrow$ **LocalTraining**$(i, \mathcal{M}_G^t)$;
  $\mathcal{M}_G^{t+1} = \frac{1}{\sum_{i \in \mathcal{N}} D_i} \sum_{i=1}^{N} D_i \mathcal{M}_i^{t+1}$;
  LocalTraining$(i, \mathcal{M})$: Split local dataset $\mathcal{D}_i$ to mini-batches of size **B** in the set $\mathcal{B}$.
  for *local epoch* $e = \{1, 2, 3, \ldots, E\}$ **do**
    for *each* $b \in \mathcal{B}$ **do**
      $\mathcal{M}_i^t \leftarrow \mathcal{M}_i^t - \eta \nabla L(\mathcal{M}; b)$;
  Return $\mathcal{M}_i^t$ to the server.

can train different ML models that essentially use the SGD method such as Support Vector Machines (SVMs), neural networks, and linear regression. Next, the obtained local model parameters from participants are sent back to the server, where they are aggregated to get the global model parameters $\mathcal{M}_G^{t+1}$. Eventually, the global model is then broadcast back to the data owners for the next round of local iteration, and the iterative process is continued. In doing so, the server minimizes the global loss function $L(\mathcal{M}_G^t)$ as the following approximation in the distributed setting of FL:

$$L\left(\mathcal{M}_G^t\right) = \frac{1}{N} \sum_{i=1}^{N} L\left(\mathcal{M}_i^t\right). \tag{4}$$

However, a single server dependency in the traditional FL framework makes the system vulnerable to threats, such as when the server behaves maliciously. Therefore, integrating FL with DLTs should be a promising approach to address limitations [23].

2.2 Distributed Ledger as a Service for FL

DLT is a peer-to-peer distributed ledger that records transactions in a network in a transparent and immutable manner. Besides, smart contract, which is considered as a key innovation in DLT/Blockchain area, provide programmability contracts to the DLTs, in the sense that the defined agreements in contracts are executed autonomously. With the mentioned nature advantages of DLTs and smart contract, the FL framework running on the top of DLT should be completely distributed and avoid the single point of failure issue.

In the DLT-based FL, we assume each client device is always connected to one of the DLT miners and, if the physical connection with the current DLT miner becomes unavailable, then the device will be automatically associated with another DLT miner. In each miner-device pair, the DLT miner works as the leader of the associated IoT devices, and they are responsible for uploading and downloading data or training models. During the training process, the IoT device

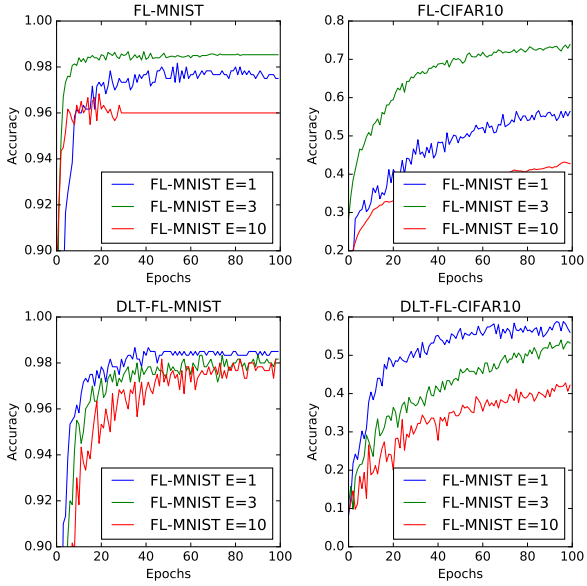


Fig. D.3: The accuracy of Standard FL and DLT-based FL.

downloads the latest global model recorded in the ledger and trains for the updated version of the local model using their private local data. After completing the local training, the device uploads the local model to the paired DLT miner and the global aggregation process starts. In the training time, all involved IoT devices are allow to download the latest information of associated DLT miners to receive the evaluation of the IoT devices and global model updates. Finally, each IoT device publishes its local training model and enters to a new round of local iteration using the newest version of the obtained global model. In this manner, the iterative ML model training process is operated until the global model has achieved a satisfactory accuracy or convergence.

Each miner has its verifier and block to ensure that the real models and the contributions of devices are updated. Each block contains a head and body parts. The blockhead contains a pointer to the next block, and the body part contains a set of validated transaction information. The local models are formed in transaction format and in order to make the solution scalable, the local models are recorded in IFPS storage, such that just a hash version of the models is recorded in the distributed ledger. The basic comparison between standard FL and DLT-based FL is presented in Fig.D.3. The accuracy is similar in both standard FL and DLT-based FL, but the time required for convergence of DLT-based FL is higher than standard FL because of extra verification and consensus in the system.

2.3 Data valuation using Shapley Value

Game theory is an economic tool best-suited to analyze a system where two or more participants get involved in to achieve a desired payoff. The Shapley Value (SV) is a solution concept of fairly distributing the incentive and payoff for the involved parties in coalition [24]. In this

regard, the SV applies mainly in scenarios where the contributions of each involved participant are unequal, but all the participants work in cooperation with each other to achieve the payoff. The SV of user i is defined as the average marginal contribution of i to all possible subsets of $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$ formed by other users as

$$\phi_i(N, U) = \frac{1}{N!} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}} \frac{U(\mathcal{S} \cup \{i\}) - U(\mathcal{S})}{\binom{N-1}{|\mathcal{S}|}}, \tag{D.2}$$

where the function $U(\cdot)$ gives the value for any subset of those users, e.g., let $\mathcal{S}$ be a subset of $\mathcal{N}$, then $U(\mathcal{S})$ gives the value of that subset. This captures the average value of the contributions of user i for subsets of all coalition of users. Intuitively, assume that the user's data is to be collected in a random order, and that every user i receives its marginal contribution for the collected data. If we average these contributions over all the possible orders of N users, we obtain $\phi_i(N, U)$. The importance of the SV is that it is the unique value division scheme that satisfies the following desirable properties described as follows.

- ***Symmetry***: For all $\mathcal{S} \subseteq \mathcal{N} \setminus \{i, j\}$, if user i and j are interchangeable, and $U(\mathcal{S} \cup \{i\}) = U(\mathcal{S} \cup \{j\})$, then, $\phi_i = \phi_j$. Thus, the users i and j contribute the same amount to every coalition of the other agents. Besides, the symmetry axiom states that such agents should receive the same payments.

- ***Dummy User***: User i is considered as dummy user if the amount that i contributes to coalition is exactly the amount that i is able to achieve alone, i.e., $\forall \mathcal{S}, i \notin \mathcal{S}, U(\mathcal{S} \cup \{i\}) - U(\mathcal{S}) = U(\{i\})$. According to the dummy user axiom, dummy users should be compensated exactly for the amount they achieve on their own. Users that make zero marginal contributions to all subsets of the data set, on the other hand, earn no compensation, for example, $\phi_i = 0$ if $U(\mathcal{S} \cup \{i\}) = 0, \forall \mathcal{S} \subseteq \mathcal{N} \setminus \{i\}$.

- ***Additivity***: For any two U_1 and U_2, we have for any user i, $\phi_i(N, U_1 + U_2) = \phi_i(N, U_1) + \phi_i(N, U_2)$, where the game $(N, U_1 + U_2)$ is defined by $(U_1 + U_2)(\mathcal{S}) = U_1(\mathcal{S}) + U_2(\mathcal{S})$ for every coalition $\mathcal{S}$.

Based on these background knowledge, we designed a distributed marketplace for trading ML models based on Blockchain and incentive mechanism in IoT environment.

3 Related Works

In this section, we first present the current works on asset trading based on Blockchain and data valuation.

Blockchain-based asset trading. With the spread of ubiquitous marketplaces, it became relevant to explore the application of IoT data trading in marketplace environments. For instance, the authors in [25] considered a dynamic decentralized marketplace and introduced the architecture for trading IoT data accordingly. The approach involves a 3-tier method is used: 1) data provider, 2) broker and 3) data consumer. The primary purpose of DLTs in their function is to manage the conditions of agreements between the parties involved. In addition, the design has a reputation system that penalizes members and lowers their rating. The authors in [26]

invested the optimization problem of revenue maximization with envy-free guarantee. The authors studied two scenarios including unit demand consumers and single minded consumers, and showed the optimization problem is APX-hard for both scenarios, which can be efficiently addressed by a logarithmic approximation. The authors in [27] took into account the trading of IoT streaming data with the presented marketplace model, where fraudulent activity during data exchange is limited. To do so, the authors introduced periodic checkpoints during data trading. In [28], the authors proposed another marketplace which flows of IoT data are the main digital assets exchanged utilizing Oracles for the off-chain queries.The authors in [29] presented a trading mode based on smart contracts. In particular, the authors employ arbitration that handles disputes during the data trading, particularly, over the data availability, and incorporates AI/ML to ensure fairness during data exchange.

Data Valuation. Evaluating the value of data has been received significant attention from both academia and industrial areas. Several works studied data valuation strategies and their applications. In this regard, the authors in [30] defined the data valuation in several categories, such as: (i) query-based pricing, where prices are attached to user-initiated queries [21], [31], [32], (ii) data attribute-based pricing, where the price model considers data attributes, such as the age of data and its credibility, using the mechanism of public price registries [33], and (iii) auction-based pricing, where the price is dynamically set following auction mechanisms [34, 35]. In [36],multiple approximation strategies for optimizing the computation complex of SV for training data are introduced. Besides, the authors proposed an soltiion to compute exact SC in specific scenario, e.g nearest neighbor classifiers. Besides, the SV also is applied in various ML application, for example, to measure the importance of model features [37, 38]. In specific, the authors addressed the problem when the same data points get the same values, and relationship between data distributions and SV function. In addition, the authors proposed an idea of distributional SV occurs resemblance to the Aumann-SV [39]. In practical manner, the authors in [40] proved that the performance of model training can be improved by removing the data with low SV value. In contrast, the performance will be decreased if we deleting the training data with high SV values.

4 System Design and Analysis

4.1 System Components

The general architecture of DLT-based model trading includes three main components: set of model owners or buyers $\mathcal{B}$, model trainers or sellers $\mathcal{S}$, and a distributed ledger, shown in Fig. D.4. We assume that each seller or buyer owns one device in the network. Within a deal (a trade) by $\mathcal{T}_i$, the seller $S_i \in \mathcal{S}$ and buyer $B_i \in \mathcal{B}$ communicate using wireless links. The ML model trading procedure occurs to complete a trade between S_i and B_i by exchanging model $\mathcal{M}_i$ and payment $\mathcal{P}_i$. First and foremost, B_i completes the deposit $\mathcal{P}_d$ to S_i via smart contracts in reference to the requested training model, $\mathcal{M}_i$. After the sellers complete the requests of the buyers, in terms of accuracy, convergence time, etc, the smart contracts are autonomous executed to pay for the effort of sellers using the amount of deposit $\mathcal{P}_d$ from buyers. Following Fig. D.4, the general procedure of interaction between a single buyer B_i and a single seller S_i can be described as follows:

Model Owner i (as buyer B_i)

B_i could be an individual or organization who needs model training. B_i sends a request b_i including task type, budget, deposit, amount of data, quality of data, price, discount, etc, to the marketplace via smart contracts. b_i will be transmitted to selected S_i and recorded in the ledger via transaction $T_{i,\text{add}}$. After receiving the trained and aggregated models from S_i and marketplace which fulfills requirements regarding to, e.g., accuracy, the B_i generates a transaction $T_{i,\text{commit}}$ which executes payment from B_i's wallet to smart contract, forwards the payment to sellers, and generates a acknowledgment message back to the distributed ledger.

Model Trainer (as seller S_i)

The model trainers play two main roles in the system: i) collects sensing data from the environment (e.g., data from surveillance systems, environmental sensing data, and geographical data), or acts as a data hub gathering data from nearby physical devices; ii) subscribes the model training requests from the buyers, and train the models downloaded from the marketplace with the local data. Seller S_i earns the payment $\mathcal{P}_i$ from B_i after successful delivery of $\mathcal{M}_i$ to B_i. After the trained models achieve a certain accuracy based on the predefined agreements in the smart contract system, upon the appearance of $T_{i,\text{commit}}$ generated by B_i, the seller S_i can receive the payment, e.g., via tokens, $\mathcal{P}_i$, which is in fact the deposited amount $\mathcal{P}_d$ by the buyer B_i, from the marketplace via smart contract. Finally, it confirms to the distributed ledger that the deal $\mathcal{T}_i$ is completed via an acknowledgment message.

Distributed Ledgers

The Blockchain maintains a distributed ledger that stores the history of all traded models in the form of blocks, which are connected in a chronological order. On top of that, the smart contracts are deployed to autonomously control the order and execute payments, e.g,. large payment or micro-payments from involved participants without the need of human intervention. In a distributed manner, the smart contracts ensure transparency, trust and automotive of exchanging data among parties. These features can be deployed based on the negotiation between model owners and customers via $T_{i,\text{deploy}}$. Furthermore, any change in smart contracts, for example, the amount of data or the model price, or updates in the discount offers, can be made via $T_{i,\text{update}}$.

In a trading system, there are an enormous amount of data exchanged among parties. Thus, increasing the number of transactions leads to slower transaction processing time and, consequently, the system's overall speed. This is reasonable as every Blockchain node needs to store and execute a computational task to validate every single transaction. Therefore, to minimize the cost of storage and execution, the trading system should record only the important data, such as payment history, aggregated global models, which could be hashed and recorded at the distributed ledger. Meanwhile, the raw data can be recorded in the distributed off-line storage component. In detail, after both model sellers S_i and model owners B_i have fulfilled requirements defined by smart contracts, the $T_{i,\text{settle}}$ is autonomously executed to query the payment $\mathcal{P}_i$ from B_i. Then, the payment $\mathcal{P}_i$ is transferred to S_i's private wallet, while the aggregated model $\mathcal{M}_i$ is delivered to the storage address of B_i. In the scope of this study, we assume that the data services (e.g., data storage, trading and task dispatching) are implemented on top of a permissionless Ethereum Blockchain [41]. In this work, the control data and ML models are formatted into normal Ethereum transactions. Furthermore, in order to improve efficiency, only

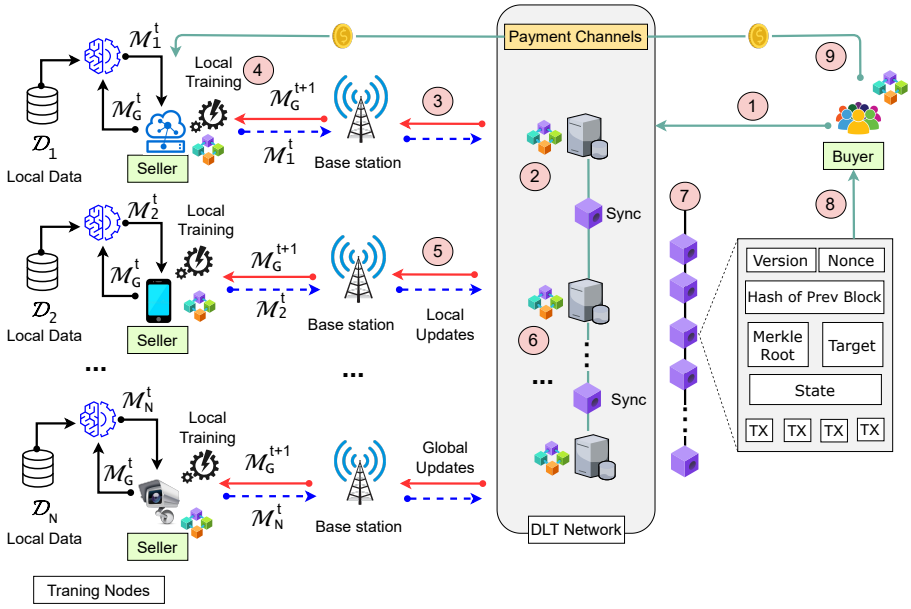


Fig. D.4: DLT-based ML Model Trading Framework Architecture. DLT-based marketplace with autonomous Smart Contract execution provide a trusted, transparent and immutable platform for trading ML models.

the digest of each transaction is recorded in the distribute on-chain ledger, and the raw data is stored off-chain by using IPFS (InterPlanetary File System).

4.2 Communication Workflow

In the DLT-based FL model trading network, we revisit the notation used to denote data owners and define the set of participants as $\mathcal{N} = \{1, 2, 3, \ldots, N\}$. The miner MI_i of DLT network is associated randomly with the IoT device. For simplicity, we consider the case which one miner is assigned to each physical IoT device. Each IoT device has to determine its own learning task-related dataset size and upload it to the ledger system to receive a reward. A distributed SV incorporates the quality of local data[3] to determine the corresponding quality of the local model. We realize that, in some cases, e.g., in healthcare, it would be better to extract features and measuring the quality based on real quality and not quantity, but it is out of scope of this research. In addition, in resource-constraint IoT environments, to reduce latency and optimize energy consumption, each device's local controller performs local optimizations to establish the best scheduling policy for device resources, such as CPU cycles scheduling.

The workflow of the system is described as below.

Step 1 (Model Initialization): The buyer B_i initiates a model $\mathcal{M}_i$ which needs to be trained and publishes to the DLT-based marketplace. The initial model is formed in the DLT transaction format T_{publish}.

[3]The quality of data signifies the size of dataset used in model training, similar to [36]. In this study, we do not consider feature attributes of data to quantify its quality.

Step 2 (Publish initialized model to the ledger): Then, the transaction T_{publish} including initialized model $\mathcal{M}_i$ is verified and recorded to the distributed ledger. At the same time, there might be many available models on the marketplace.

Step 3 (Model Seller download train-required models): The potential seller S_i can see the list of available models on the marketplace and choose to download a copy of one or multiple models of interest to train with its local data of S_i.

Step 4 (Local Model Training): After downloading the models from the distributed ledger, the sellers train the model based on their local data. The device, i.e., the seller S_i, has its own local dataset $\mathcal{D}_i$. Local model training aims to minimize the loss function $f(\mathcal{M}_i, \mathcal{D}_i)$, where $\mathcal{M}_i$ is the local model of device d_i and $\mathcal{D}_i$ is its local dataset.

Step 5 (Local trained model is updated to the ledger): Next, the device S_i is randomly associated with the miner MI_i to which it uploads the trained local model to the distributed ledger via smart contract. The smart contract has functions to record the updated local models from clients via DLT interface, e.g., Web3.

Step 6 (Cross-verification of the local models): After receiving the local model published by IoT device in the format of transactions, the DLT miner MI_i put the local model in newly generated blocks and broadcasts the model to other DLT miners in the network. Next, until other DLT miners receive the broadcasted blocks, including the local models of clients, they will verify the accuracy of local models and put the models to the new generated blocks. During this process, all the aggregated models are broadcasted to the all DLT miners in the system, and DLT miners will compare the consistency and accuracy among aggregated models. To that end, the most one will be chosen as the correct global model. Then, DLTs miners record the correct global model and the contribution of the IoT devices into the distributed ledger via smart contracts features. Otherwise, the rest of global models are considered as faulty updates.

Step 7 (The generation and propagation of blocks): In order to generate a new block in the distributed ledger, DLT miners need to compute a block hash for mining and solve a cryptographic puzzle based on SHA-256, which is a one-way hash function. As defined in popular Proof-of-Work (PoW) Blockchain, e.g., Bitcoin, Ethereum, DLT miners perform a PoW algorithm until it finds a desired nonce value or receives a new generated block from other DLT miners [42]. There is a case, however, that the $MI_i \in MI$ acts as the DLT miner that finds the needed nonce value at the earliest, and its candidate block is generated as a new block and propagated to the other DLT miners in the network. Meanwhile, the chain can be engaged in forked problem in which multiple DLT miners find out a nonce value at the same moment. To address this issue, we use an ACK message that allows DLT miners to transmit only when each DLT miner gets the new block, which determines whether there is a fork on the main chain. Then, the DLT miner MI_0, which creates that newly generated block, will wait for a waiting time defined by the block ACK. Otherwise, if a fork is generated again, the process back to to previous phase to resolve the issue.

Step 8 (Settlement): After the model accuracy achieves a particular value in the smart contract, the smart contract settles the deal between buyer and sellers. The finalized model is updated to the buyer and the incentive is funded to the sellers.

Step 9 (Incentive to Sellers): Based on the contribution of each seller, the smart contract computes their contribution and transfer to sellers appropriate funds. The smart contract provides a mechanism of transparent and immutable recording and accounting contribution logs on the distributed ledger. Based on the contribution history from ledger, the clients can receive the incentives and rewards in tokens via off-chain payment channels.

Algorithm 2: `AFS` Algorithm

Input: Local minibatch size **B**, T is number of global interactions, number of local epochs E, and learning rate η.
Output: $\mathcal{M}^T$, and $\phi_1, \phi_2, \ldots, \phi_n$.
Initialize: $\mathcal{M}^0, \widetilde{\mathcal{M}}^0_{\mathcal{S}}$, where $\mathcal{S} \subseteq \mathcal{N} = \{1, 2, 3 \ldots, n\}$.
for *each round* $t = \{0, 1, 2, \ldots, T\}$ **do**
  Transmit $\mathcal{M}^t$ to $i \in \mathcal{S}$ clients;
  $\mathcal{M}^t_i \leftarrow ModelUpdate(i, \mathcal{M}^t)$;
  $\delta^{t+1}_i \leftarrow \mathcal{M}^t_i - \mathcal{M}^t$, $\forall i \in \mathcal{N}$;
  $\mathcal{M}^{t+1} \leftarrow \mathcal{M}^t + \sum_{i=1}^{n} \frac{D_i}{\sum_{i=1}^{n} D_i} \cdot \delta^{t+1}_i$;
  for *each subset* $\mathcal{S} \subseteq \mathcal{N}$ **do**
    $\widetilde{\mathcal{M}}^{t+1}_{\mathcal{S}} \leftarrow \widetilde{\mathcal{M}}^t_{\mathcal{S}} + \sum_{i \in \mathcal{S}} \frac{D_i}{\sum_{i \in S} D_i} \cdot \delta^{t+1}_i$;
Initialize: $m = 0$.
while Convergence criteria not meet **do**
  $m = m + 1$;
  π^m: random permutation of clients with data samples to collaboratively train $\mathcal{M}^T$;
  $v^m_0 \leftarrow U(\widetilde{\mathcal{M}}^0_{\emptyset})$;
  for $n \in \{1, 2, \ldots, |\mathcal{S}|\}$ **do**
    if $|U(\mathcal{M}_{\mathcal{S}}) - v^m_{n-1}| < PT$ **then**
      $v^m_n = v^m_{n-1}$;
    else
      $\mathcal{S} \leftarrow \{\pi^m[1], \pi^m[2], \ldots, \pi^m[n]\}$; $\mathcal{M}^T_{\mathcal{S}} \leftarrow \sum_{i \in \mathcal{S}} \frac{D_i}{\sum_{i \in \mathcal{S}} D_i} \cdot \widetilde{\mathcal{M}}^T_i$;
      $v^m_n \leftarrow U(\mathcal{M}^T_{\mathcal{S}})$;
    $\phi_{\pi^m[n]} \leftarrow \frac{m-1}{m} \phi_{\pi^{m-1}[n]} + \frac{1}{m}(v^m_n - v^m_{n-1})$;
Return $\mathcal{M}^T$, and $\phi_1, \phi_2, \ldots, \phi_n$.
ModelUpdate$(i, \mathcal{M})$: Split local dataset $\mathcal{D}_i$ to mini-batches of size **B** in the set $\mathcal{B}$.
for *local epoch* $e = \{1, 2, 3, \ldots, E\}$ **do**
  for *each* $b \in \mathcal{B}$ **do**
    $\mathcal{M}^t_i \leftarrow \mathcal{M}^t_i - \eta \nabla L(\mathcal{M}; b)$;
  Return $\mathcal{M}^t_i$ to ledger.

Step 10 (Record receipt to the Ledger): All bills and receipts are recorded immutably in the distributed ledger, which allows participants to check and control their deal. Besides, we also implemented the off-chain storage solution named IPFS to store hashes of data locations on the ledger instead of raw data files. The hashes can be used to query the exact file or models through the DLT systems.

4.3 Distributed Shapley Value (DSV) Calculation

The Standard Federated Shapley Value (SFSV) [43] calculates the SV of data contributors based on (D.2). In our setting, the utility in (D.2) takes into account the value brought by a subset of data contributors $\mathcal{S}$ in improving the performance of the trained global model after trading their local models $\mathcal{M}_{\mathcal{S}}$, defined hereafter as $U(\mathcal{M}_{\mathcal{S}})$. Then, SFSV trains federated models based on the different subsets $\mathcal{S}$ of contributors, and these models are evaluated on the standard test set. However, computing the SV directly according to SFSV is time-consuming because models on all the combinations of data sets need to be trained and evaluated. By default, the SV computes the average contribution of a data source to every possible subset of other data sources. So that, evaluating the SV incurs significant communication and computation cost when the data is decentralized [44]. Consequently, for data SV in the FL environment, the methods in [36, 43] to calculate SV introduce extra training rounds on combinations of datasets from different data providers. Furthermore, the cost for extra rounds for training models could be expensive when the data volume is large. Therefore, there is a need for new strategies to evaluate the data value in FL.

The main idea to that end is to exploit the gradients information during the training process of the global model $\mathcal{M}$ to approximately reconstruct the local models trained with different combinations of the client's datasets. Thereby, our approach (as described in Algorithm 2) eliminates the burden for the local models to be frequently re-trained to evaluate clients' contributions. In fact, the SV does not consider the order of data sources. However, in FL, it is of significant importance to take into account the order of data used for the model training so as to ensure a fair convergence. Furthermore, the updates of model are enforced to diminish over time by using, for example, a decaying learning rate [45]. Hence, the sources used towards the end of the learning process could be less influential than those used earlier. Therefore, to accommodate these attributes of learning properties in the decentralized model training paradigm of FL, we need to define new and efficient ways to compute SV. In this regard, based on the neutrality of FL, the SV for FL (FSV) could be computed in two different strategies. The first method (called Single-Cal) reconstructs models by updating the initial global model $\mathcal{M}$ in FL with the gradients in different rounds and calculates the FSV by the performance of these reconstructed models. For example, if we want to reconstruct the model of $\mathcal{M}_{(i,j)}$ trained on the datasets of $\mathcal{D}_i$ and $\mathcal{D}_j$ of corresponding users, use the gradients information from sellers i and j in each round to update the initial global model $\mathcal{M}$ generated by the buyer. Then, the contribution is calculated using (D.2). The second method (called Multi-Cal) calculates FSV in each training round by updating the global model $\mathcal{M}$ from the previous training round with the value of gradients in the current training round. Next, the FSVs are aggregated from multiple rounds to get the final result. Therefore, there is no extra training process needed; these methods are considered efficient. The main difference between these two strategies is that the first method approximates models through complete global iterations and only evaluates them to find SV afterwards. The second one approximates and evaluates models for every global iteration and calculates the marginal contribution for each global iteration. So that makes the second method more computationally expensive than the first one. To address this issue, we propose a new algorithm `AFS` based on the first approach with the use of Truncated Monte-Carlo (TMC) [36] in Algorithm 2. In principle, `AFS` is an engineered derivative of the TMC algorithm to characterize clients' contributions with their available data samples in the collaborative training framework. In doing so, `AFS` evaluates the marginal contribution of each client instead of a subset of training

data points, unlike the TMC method; thus, allowing clients to engage in the ML model trading with the offered incentive signals based on their data contributions.

Specifically, the first part of the algorithm shows the operation of the distributed ledger, lines 1–8 and lines 10–19. In line 1, the global model $\mathcal{M}^0$ and reconstructed models based on different chosen subsets $\mathcal{S} \subseteq \mathcal{N} = \{1, 2, \ldots, N\}$ are initialized. Next, the distributed ledgers broadcast the global model $\mathcal{M}^t$ to n selected clients in each global training round t in line 3, and then receive the updates $\mathcal{M}_i^t$ from these clients in line 4 and the gradients of clients $\delta_i^t, \forall i \in \mathcal{S}$ for model aggregation are computed. After that, the global model is updated in line 6 as

$$\mathcal{M}^{t+1} \leftarrow \mathcal{M}^t + \sum_{i=1}^{n} \frac{D_i}{\sum_{i=1}^{n} D_i} \cdot \delta_i^{t+1}. \tag{D.3}$$

Next, instead of updating all the local models $\mathcal{M}_i^t, \forall i \in \mathcal{S}$ in every global interactions $t = \{0, 1, \ldots, T\}$, we can update only n models $\widetilde{\mathcal{M}}_i^t$ and compute $\mathcal{M}_{\mathcal{S}}^T$ directly as a weighted average at the end, as $\widetilde{\mathcal{M}}_i^{t+1} \leftarrow \widetilde{\mathcal{M}}_i^t + \frac{D_i}{\sum_{i=1}^{n} D_i} \delta_i^{t+1}$.

We observe that evaluation of a model incurs a considerable cost in terms of time, especially if the test set is large. And with the basic idea of a single-round algorithm, we must reconstruct and evaluate 2^n models. Hence, we applied the method of TMC to decrease the computation cost and developed a tailored variant of TMC, i.e., the `AFS` algorithm, to address this issue. The details of the adapted TMC method is as follows. First, we sample a random permutation of clients π^m with their data samples used to train the global model [43]. After that, we scan from the first clients to the last client and calculate the marginal contribution of every new client's data in the training process. By repeating this process over multiple permutations, the approximation of SV is the average of all the calculated marginal contributions. The while loop is run until certain convergence criteria are met. In this work, we stop the loop when the average percentage change after a TMC iteration m is less than a certain performance threshold (PT). For example, PT can be varied from 1%-2% from $U(\mathcal{M}_{\mathcal{S}})$.

In line 21, the trained federated model $\mathcal{M}^T$ and the SVs are finally obtained. The local training part for the clients (lines 22–25) show how the clients use private data to train the model received from the distributed ledger. The clients use the classical gradient descent algorithm and report their updated local models $\mathcal{M}_{i|i=\{1,2,3,\ldots,n\}}$ to the distributed ledger.

4.4 Performance bound on `AFS` algorithm

An analytical bound on the `AFS` algorithm can be derived by taking the properties of TMC sampling approach into account [36]. We consider `AFS` estimates the contribution of the individual client in the federated setting for a supervised learning task with probability at least $(1 - \alpha)$ that our estimator error is ϵ. Then, we are interested in evaluating the general performance bound on `AFS` such that

$$\Pr(|\phi^F - \phi^S| \geq \epsilon) \leq \alpha, \tag{D.4}$$

where $\phi^S =< \phi_1^S, \phi_2^S, \ldots, \phi_n^S >$ is the vector of Shapley contributions generated by the standard SV and $\phi^F =< \phi_1^F, \phi_2^F, \ldots, \phi_n^F >$ is the approximation FSV using the proposed `AFS` method. Assume that we know the data distribution of clients to evaluate its marginal contribution. Then, the sampling $|\mathcal{S}|$ made during the evaluation process reflects the bound on the obtained approximation of `AFS`. Without loss of generality, we assume it is possible to quantify

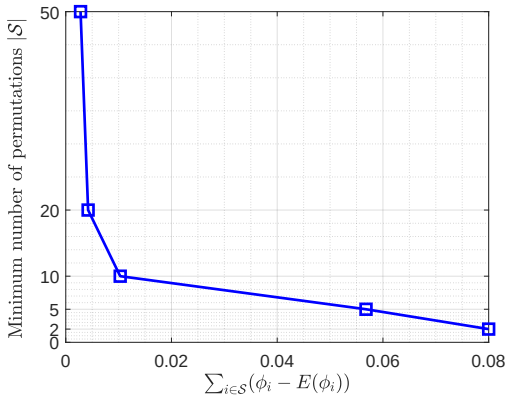


Fig. D.5: Performance analysis of AFS algorithm.

the range v of the client's data marginal contribution in improving the global model. Then, we have the following Lemma 1.

Lemma 1. *Considering the TMC sampling approach [36] for evaluating the Shapley value, we have a minimum permutation $|\mathcal{S}|$ for a known range of the client's marginal contribution v defining an upper bound of $\mathcal{O}\left(\frac{v}{|\mathcal{S}|}\right)$ on AFS such that $\alpha \geq 2\exp\left(\frac{-2|\mathcal{S}|\epsilon^2}{v^2}\right)$ satisfies for $0 < \epsilon, \alpha \leq 1$.*

Proof. The proof can be derived using Hoeffding theorem [46] for a known range of marginal contribution of clients. In practice, the distributed ledger can reuse the average of marginal contributions of clients, with known range v, to derive a sampling permutation $|\mathcal{S}|$. Then, we have $\Pr\left(\sum_{i\in\mathcal{S}\subseteq N}(\phi_i - \mathbb{E}(\phi_i)) \geq \Delta\right) \leq 2\exp\left(\frac{-2|\mathcal{S}|\Delta^2}{v^2}\right)$. Taking the average of marginal contributions on the left-hand side of the inequality, and combining it with (D.4), we get $\Pr(|\phi^F - \phi^S| \geq \epsilon) \leq 2\exp\left(\frac{-2|\mathcal{S}|\epsilon^2}{v^2}\right)$. This concludes the proof. □

In Fig. D.5, we show the results on performance analysis of AFS algorithm. We observed variability in the minimum permutation $|\mathcal{S}|$ required to ensure a defined deviation between the average value of contributions across clients, as measured using AFS, and the standard SV. For tighter bounds, the number of required permutations is large. This is intuitive, as the distributed ledger expects a larger sampling value $|\mathcal{S}|$ to define better confidence bound on the performance that minimizes the approximation error using AFS.

5 Performance Evaluation

5.1 Experimental Settings

To demonstrate the applicability of our proposed system, we implement a proof-of-concept for the trading model in an IoT network. In this section, we introduce enabling technologies

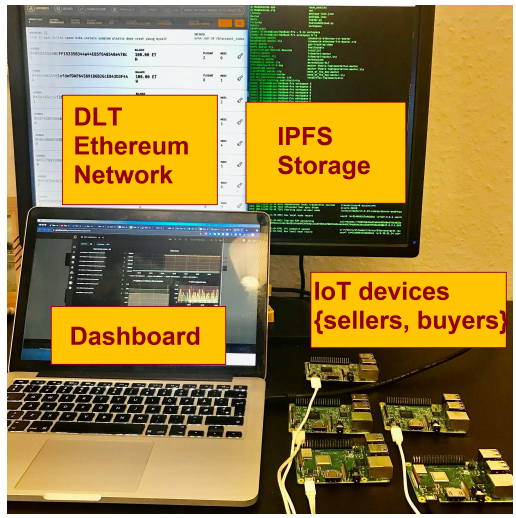


Fig. D.6: Blockchain-enabled model trading testbed. The testbed includes DLT Ethereum network running over Ganache, IPFS storage to address scalability issue, monitor dashboard and 5 IoT raspberry devices standing for marketplace participants as well as DLT clients.

involved with the prototype.

Distributed Ledger

In this study, we implement Ethereum platform for the experimental. Ethereum[4] is a distributed public blockchain network that focuses on running programming code of any decentralized application. Specifically, Ethereum is a platform for sharing information across the globe that cannot be manipulated or changed. Ethereum has its own cryptocurrency, called *Ether* (ETH), and its own programming language, called *Solidity*. The decentralized applications on the network is called *Đapps*. Practically, Ethereum provides a convenient platform for development and smart contracts system to integrate with FL. We run Ethereum network via Ganache[5] which is a personal blockchain for rapid Ethereum distributed application development.

Datasets

In the scope of this study, we conducted the experiments on the MNIST data set [47]. The dataset contains around 60,000 training images and over 10,000 testing images. Each client holds a part of dataset locally depending on the scenarios.

IoT Devices and Workstation

We use Raspberry Pi 3 with the following configurations: Pytorch, OS Raspbian GNU/Linux 10, and Python version 3.7. We note that CUDA is not available for the model. The workstation has the system configurations as CPU i7-7700HQ, GPU GTX, Pytorch, OS Linux Ubuntu 20.04

[4]https://ethereum.org/
[5]https://www.trufflesuite.com/ganache

, Python version 3.7.8 using Anaconda, and the CUDA version 11. These IoT devices are connected via WiFi access point.

Evaluation Metrics.

We consider several performance metrics for comparison.

- **Cost of Smart Contract:** We study the fees made by users to compensate for the computing energy required to process and validate transactions on the Ethereum.
- **Incentive per worker:** The amount of tokens delivered to sellers based on their contributions of training the model.
- **Maximum Different**: The performance score function $U(\cdot)$ is chosen to be the accuracy function. The SVs are then calculated according to the different schemes. For comparison of the accuracy of the SV, all SVs calculated are first standardized by scaling them by a common factor such that $\sum_{i=1}^{n} \phi_i = 1$.

This is appropriate because profit distribution will likely be based on the percentage contribution. Then, the **maximum different** $D_{\max}$ measures the maximum difference that a data provider should be allocated by the definition and by approximated calculation. The calculation is shown as below:

$$D_{\max} = \max_{i \in \{1,\ldots,n\}} |\phi_i^F - \phi_i^S|. \tag{D.5}$$

Scenarios

- **(S1)**. In Scenario 1, the compared algorithms have same distribution with same dataset size, i.e., each client dataset $\mathcal{D}_i, \forall i \in \mathcal{N}$ has the same amount of training image samples.
- **(S2)**. In Scenario 2, we introduce the case with same distribution but different dataset size. The training set is divided randomly into 5 parts with the same ratio of data size.
- **(S3)**. In Scenario 3, we use different distribution with same dataset size. Each client's dataset $\mathcal{D}_i, \forall i \in \{1, 2, 3, 4, 5\}$ has the same size, but the training images are not equally divided for each digit.
- **(S4)**. In Scenario 4, we consider the case having an added noise feature with same dataset. First, we split the training set in a similar manner as **(S1)**. Afterwards, we generate Gaussian noise for the dataset. This is done by adjusting the standard deviation of the normal distribution.

5.2 Results

Smart Contract Execution Cost

In this part, the proof-of-concept of proposed model trading platform is deployed in a private Ethereum Blockchain called *Ganache*[6]. In distributed application *Dapps*, the smart contract

[6]https://www.trufflesuite.com/ganache

Table D.2: Execution cost of smart contracts

Smart Contracts	From	Gas	Ether	USD
Contract Registry	0x283D382F	1459430	$15.9{\cdot}10^{-5}$	0.0723
AddWorker	0x283D382F	452467	$45.2{\cdot}10^{-5}$	0.0692
AddWorker	0x283D382F	452545	$45.2{\cdot}10^{-5}$	0.0692
AddWorker	0x283D382F	452436	$45.2{\cdot}10^{-5}$	0.0692
AddWorker	0x283D382F	452545	$45.2{\cdot}10^{-5}$	0.0692
AddWorker	0x283D382F	452436	$45.2{\cdot}10^{-5}$	0.0692
ModelTransmission	0x5846F427	19374	$19.3{\cdot}10^{-5}$	0.1621
ModelTransmission	0x9dD8Fd06	243482	$24.3{\cdot}10^{-5}$	0.0902
ModelTransmission	0x98HF8F94	228779	$22.3{\cdot}10^{-5}$	0.1121
ModelTransmission	0x8H9FH780	253924	$25.3{\cdot}10^{-5}$	0.0951
ModelTransmission	0x0932FD99	263924	$19.3{\cdot}10^{-5}$	0.0571
ModelTraining	0x5846F427	223924	$22.3{\cdot}10^{-5}$	0.1021
ModelTraining	0x9DD8Fd06	253924	$25.3{\cdot}10^{-5}$	0.0951
ModelTraining	0x98HF8F94	193924	$19.3{\cdot}10^{-5}$	0.0571
ModelTraining	0x8H9FH780	253924	$25.3{\cdot}10^{-5}$	0.0951
ModelTraining	0x0932FD99	253924	$19.3{\cdot}10^{-5}$	0.0571
ModelAggregation	0x5846F427	324942	$32.4{\cdot}10^{-5}$	0.0766
ModelAggregation	0x9dD8Fd06	283445	$22.4{\cdot}10^{-5}$	0.0408
ModelAggregation	0x98HF8F94	214939	$21.4{\cdot}10^{-5}$	0.0709
ModelAggregation	0x8H9FH780	253924	$25.3{\cdot}10^{-5}$	0.0951
ModelAggregation	0x0932FD99	193924	$19.3{\cdot}10^{-5}$	0.0571
Settlement	0x283D382F	212559	$21.3{\cdot}10^{-5}$	0.0712
PayChannelExecute	0x283D382F	212538	$21.2{\cdot}10^{-5}$	0.0702

* 1 Ether = 10^9 Gwei; 1 USD = 246,940.5627 Gwei

plays as key role in controlling and autonomously executing pre-defined agreements between the participants. We implemented and tested smart contracts using Remix IDE[7]. In Ethereum network, there is a fee called *gas*, needed to pay for any operation or transaction execution that changes the DLT states, which guarantees that smart contracts running in Ethereum Virtual Machine (EVM) [41] will be terminated eventually. In the scope of this research, we used Gwei[8] to evaluate the cost of different operations, for example, *AddWorker, ModelTransmisson, ModelTraining,* or *Settlement* in the model trading process. The result is demonstrated in Table D.2.

Incentive per client

In Fig. D.8, we show the comparison of received incentives by each training client based on their contribution to the global model training. The incentive is equivalent to tokens clients receive. As expected, in Fig. D.8a, where the MNIST dataset is divided equally with a ratio of 2:2:2:2:2 for five involved clients, the amount of tokens they receive are almost similar as

[7]https://remix.ethereum.org/
[8]https://www.cryps.info/

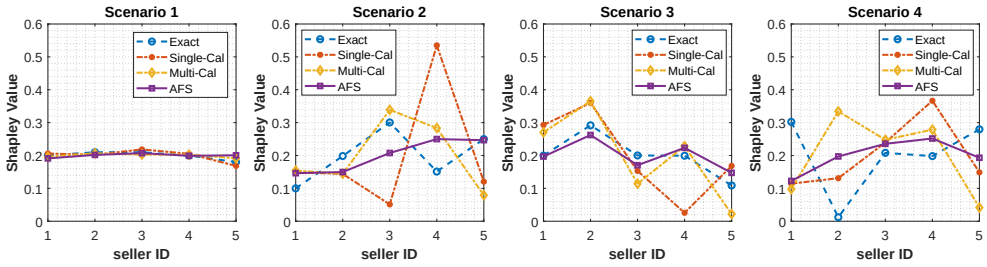
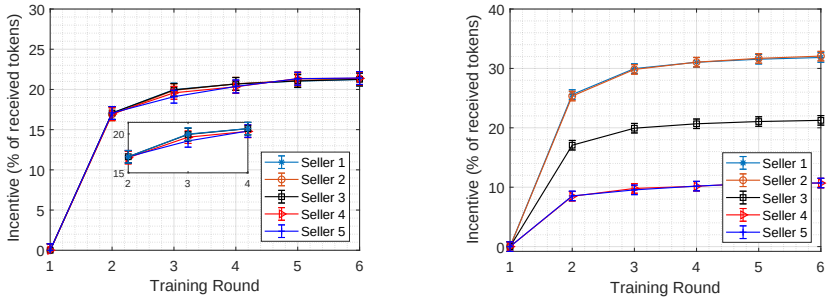


Fig. D.7: Shapley Value of each seller for different methods, namely Exact, Single-Cal, Multi-Cal, and AFS in four scenarios. The application of SV in FL is not efficient due to increasing of communication in distributed systems in comparison with centralized one, and the unbalance of data source distribution.



(a) Received incentive per client with the dataset distribution ratio 2:2:2:2:2.

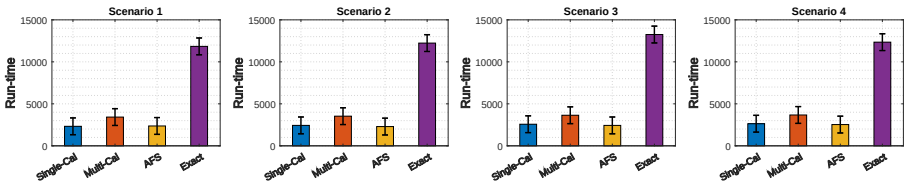
(b) Received incentive per client with the dataset distribution ratio 3:3:2:1:1.

Fig. D.8: Incentive of clients received in tokens for their contribution efforts.

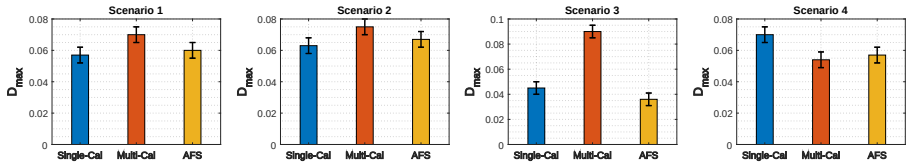
expected. In Fig. D.8b, we show the comparison of the received percentage of tokens that clients can achieve with the dataset ratio of 3:3:2:1:1. We observe that client 1 and client 2 has the same amount of dataset, so they receive the same amount of tokens for their contribution, similar to the case for clients 4 and 5. Note that the sellers can train the models with poor quality, which, in fact, reduces the stability and performance of the global models. In this regard, there exist several mechanisms to handle such dishonest reporting of parameters in the FL setting, such as [13, 29, 48]. Similar to this, the DLT keeps track of the contribution of devices and the gradient information and the size of data samples to regularly infer (check) the relationship between the expected model quality, reported data samples, and the obtained SV as Fig. D.7; hence, dealing untruthful reporting. However, the detailed study of this mechanism is out of scope for this work. In Fig. 8, the AFS shows a better performance while other methods turns out quite random SVs, especially in scenario 2 and 4 where the size of dataset is random and noise added.

Execution time and maximum different comparison

In Fig. D.9, we show the time performance of exact FL, Single-Cal, Multi-Cal, and AFS protocol. The Multi-Cal algorithm is more computational expensive than the Single-Cal algorithm. The standard exact method is the slowest one because the standard SV is naturally not compati-



(a) Runtime comparison of strategies.



(b) Comparison of $D_{\max}$.

Fig. D.9: Comparison of execution time and $D_{\max}$ between algorithms, Exact, Single-Cal, Multi-Cal, and AFS, in four different scenarios. AFS is outperform compared with standard Exact, and approximately faster 15% compared with Multi-Cal and Single-Cal for measuring contribution of participants.

ble with the FL. In the Scenario 1, each worker has same quality and quantity of dataset, so that we expect each worker has same contribution and receive equally the amount of incentive. The results show that the Single-Cal and AFS algorithm have higher efficiency in execution time. The exact method is around 5 times slower than the rest of methods because of frequent model retrain process. Meanwhile, the $D_{\max}$ of methods are relatively low, around 0.05. Similar in Scenario 2 with the same size of dataset and different distribution, AFS and Single-Cal have better performance in running time and the accuracy. In Scenario 3, we observe the Single-Cal method performs better in the setting with same data size but different distributions, and further, it also requires fewer permutation coverage as compared to Multi-Cal, and nominally higher than the AFS. However, in Scenario 4 with more noisy data, the Multi-Cal shows better results, $\approx 10\%$ in run-time but and $\approx 15\%$ in terms of maximum different value.

6 Conclusion

In this paper, we proposed a DLT-based marketplace for trading ML models, which helps companies and organizations train their learning models in a scalable and efficient manner. An incentive mechanism exists to stimulate participants in joining and training the learning models on the marketplace, which pays participants based on their contributions to train the model. To that end, an extended Data Shapley Value (DSV) for the federated environment is proposed to measure each participant's contribution in the model training process. Finally, with extensive experimental evaluations with Ethereum Blockchain to build a marketplace for model trading using smart contracts and IoT devices acting as participants, we demonstrated the design and performance of the proposed ecosystem.

7 Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957218 (Project IntellIoT).

References

[1] I. Report, "The growth in connected iot devices is expected to generate 79.4zb of data in 2025, according to a new idc forecast," [Online]https://www.idc.com, 2019, (Accessed on 12/04/2020).

[2] Z. Huang, X. Su, Y. Zhang, C. Shi, H. Zhang, and L. Xie, "A decentralized solution for iot data trusted exchange based-on blockchain," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 1180–1184.

[3] C. Perera, "Sensing as a service (s2aas): Buying and selling iot data," *arXiv preprint arXiv:1702.02380*, 2017.

[4] W. Mao, Z. Zheng, and F. Wu, "Pricing for revenue maximization in iot data markets: An information design perspective," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1837–1845.

[5] B. Bishoi, A. Prakash, V. Jain *et al.*, "A comparative study of air quality index based on factor analysis and us-epa methods for an urban environment," *Aerosol and Air Quality Research*, vol. 9, no. 1, pp. 1–17, 2009.

[6] J. Jo, B. Jo, J. Kim, S. Kim, and W. Han, "Development of an iot-based indoor air quality monitoring platform," *Journal of Sensors*, vol. 2020, 2020.

[7] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated learning in the sky: Aerial-ground air quality sensing framework with uav swarms," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9827–9837, 2021.

[8] S. Moltchanov, I. Levy, Y. Etzion, U. Lerner, D. M. Broday, and B. Fishbain, "On the feasibility of measuring urban air pollution by wireless distributed sensor networks," *Science of The Total Environment*, vol. 502, pp. 537–547, 2015.

[9] N. Gruschka, V. Mavroeidis, K. Vishi, and M. Jensen, "Privacy issues and data protection in big data: a case study analysis under gdpr," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 5027–5033.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[11] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in iot-based information system for emergency medical services," *IEEE Transactions on Industrial informatics*, vol. 10, no. 2, pp. 1578–1586, 2014.

[12] R. Radhakrishnan and B. Krishnamachari, "Streaming data payment protocol (sdpp) for the internet of things," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1679–1684.

[13] C. Niu, Z. Zheng, F. Wu, X. Gao, and G. Chen, "Achieving data truthfulness and privacy preservation in data markets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 1, pp. 105–119, 2018.

[14] D. C. Langevoort, "Fraud and insider trading in american securities regulation: Its scope and philosophy in a global marketplace," *Hastings Int'l & Comp. L. Rev.*, vol. 16, p. 175, 1992.

[15] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3241–3256, 2020.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2008.

[17] L. D. Nguyen, I. Leyva-Mayorga, A. N. Lewis, and P. Popovski, "Modeling and analysis of data trading on blockchain-based market in iot networks," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6487–6497, 2021.

[18] B. Chen, D. He, N. Kumar, H. Wang, and K.-K. R. Choo, "A blockchain-based proxy re-encryption with equality test for vehicular communication systems," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2048–2059, 2021.

[19] L. D. Nguyen, A. E. Kalor, I. Leyva-Mayorga, and P. Popovski, "Trusted wireless monitoring based on distributed ledgers over nb-iot connectivity," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 77–83, 2020.

[20] T. Wang, C. Zhao, Q. Yang, S. Zhang, and S. C. Liew, "Ethna: Analyzing the underlying peer-to-peer network of ethereum blockchain," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2131–2146, 2021.

[21] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[22] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[23] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2019.

[24] A. E. Roth, *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[25] P. Gupta, S. Kanhere, and R. Jurdak, "A decentralized iot data marketplace," *arXiv preprint arXiv:1906.01799*, 2019.

[26] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang, "Towards practical differentially private convex optimization," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 299–316.

[27] S. Bajoudah, C. Dong, and P. Missier, "Toward a decentralized, trust-less marketplace for brokered iot data trading using blockchain," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 339–346.

[28] P. Missier, S. Bajoudah, A. Capossele, A. Gaglione, and M. Nati, "Mind my value: a decentralized infrastructure for fair and trusted iot data trading," in *Proceedings of the Seventh International Conference on the Internet of Things*, 2017, pp. 1–8.

[29] W. Xiong and L. Xiong, "Smart contract based data trading mode using blockchain and machine learning," *IEEE Access*, vol. 7, pp. 102 331–102 344, 2019.

[30] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song, "A principled approach to data valuation for federated learning," in *Federated Learning*. Springer, 2020, pp. 153–167.

[31] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6341–6345.

[32] P. Upadhyaya, M. Balazinska, and D. Suciu, "Price-optimal querying with data apis," *Proceedings of the VLDB Endowment*, vol. 9, no. 14, pp. 1695–1706, 2016.

[33] J. R. Heckman, E. L. Boehmer, E. H. Peters, M. Davaloo, and N. G. Kurup, "A pricing model for data markets," *iConference 2015 Proceedings*, 2015.

[34] M. Mihailescu and Y. M. Teo, "Dynamic resource pricing on federated clouds," in *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE, 2010, pp. 513–517.

[35] J.-S. Lee and B. Hoh, "Sell your experiences: a market mechanism based incentive for participatory sensing," in *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2010, pp. 60–68.

[36] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos, "Towards efficient data valuation based on the shapley value," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1167–1176.

[37] S. Cohen, G. Dror, and E. Ruppin, "Feature selection via coalitional game theory," *Neural Computation*, vol. 19, no. 7, pp. 1939–1961, 2007.

[38] E. Strumbelj and I. Kononenko, "An efficient explanation of individual classifications using game theory," *The Journal of Machine Learning Research*, vol. 11, pp. 1–18, 2010.

[39] R. J. Aumann and L. S. Shapley, *Values of non-atomic games*. Princeton University Press, 2015.

[40] S. Tang, A. Ghorbani, R. Yamashita, S. Rehman, J. A. Dunnmon, J. Zou, and D. L. Rubin, "Data valuation for medical imaging using shapley value and application to a large-scale chest x-ray dataset," *Scientific reports*, vol. 11, no. 1, pp. 1–9, 2021.

[41] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[42] X. Ding, J. Guo, D. Li, and W. Wu, "An incentive mechanism for building a secure blockchain-based internet of things," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 477–487, 2021.

[43] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2242–2251.

[44] Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5171–5183, 2020.

[45] Z. Charles and J. Konečnỳ, "On the outsized importance of learning rates in local update methods," *arXiv preprint arXiv:2007.00878*, 2020.

[46] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The collected works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.

[47] "Mnist handwritten digit database, yann lecun, corinna cortes and chris burges," http://yann.lecun.com/exdb/mnist/, (Accessed on 03/02/2022).

[48] A. Agarwal, M. Dahleh, and T. Sarkar, "A marketplace for data: An algorithmic solution," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 701–726.

Paper E

B-ETS: A Trusted Blockchain-based Emissions Trading System for Vehicle-to-Vehicle Networks

Authors:
Duc-Lam Nguyen, Amari N. Lewis, Israel Leyva-Mayorga,
Amelia Regan, and Petar Popovski

The paper has been published in the
7th International Conference on Vehicle Technology and Intelligent Transport Systems, pp. 171-179. SCITEPRESS Digital Library, 2021.
⋆ **Best Paper Award** ⋆

The layout has been revised.

Abstract

Urban areas are negatively impacted by Carbon Dioxide (CO_2) and Nitrogen Oxide (NO_x) emissions. In order to achieve a cost-effective reduction of greenhouse gas emissions and to combat climate change, the European Union (EU) introduced an Emissions Trading System (ETS) where organizations can buy or receive emission allowances as needed. The current ETS is a centralized one, consisting of a set of complex rules. It is currently administered at the organizational level and is used for fixed-point sources of pollution such as factories, power plants, and refineries. However, the current ETS cannot efficiently cope with vehicle mobility, even though vehicles are one of the primary sources of CO_2 and NO_x emissions. In this study, we propose a new distributed Blockchain-based emissions allowance trading system called B-ETS. This system enables transparent and trustworthy data exchange as well as trading of allowances among vehicles, relying on vehicle-to-vehicle communication. In addition, we introduce an economic incentive-based mechanism that appeals to individual drivers and leads them to modify their driving behavior in order to reduce emissions. The efficiency of the proposed system is studied through extensive simulations, showing how increased vehicle connectivity can lead to reduction of the emissions generated from those vehicles. We demonstrate that our method can be used for full life-cycle monitoring and fuel economy reporting. This leads us to conjecture that the proposed system could lead to important behavioural changes among the drivers.

1 Introduction

Typical passenger vehicles emit about 4.6 metric tons of carbon dioxide CO_2 per year. The European Union's Emission Trading System (EU-ETS) is the world's first major carbon trading market with the main goal to combat climate change and reduce Greenhouse Gas (GHG) emissions in a cost effective way. The EU-ETS works on a Cap-and-Trade (CAP) principle which allows companies that generate point source emissions to receive or buy emission allowances, which can be traded as needed [1]. The process of our B-ETS CAP program is described in Figure E.1, where it is seen that it is based on a complex centralized method of trading among the organizations involved. The first step in CAP is to make a centralized decision (by a regulatory agency or some other collective entity) on the aggregate quantity of emissions allowed. Allowances are then written in accordance with this quantity, after which they are distributed among the sources responsible for the emissions.

Since 2018, the EU-ETS began penalizing vehicle manufacturers for exceeding the targets for fleet-wide emissions for new vehicles sold in any given year. The manufacturers are required to pay an excess emissions premium for each newly registered car. A penalty of €95 must be paid for each gram per km above the target [1] and the target of CO_2 for the 2020-2021 period is set to 95 grams per km. In this work, we address the need for a new trusted and distributed system which can audit emissions at the vehicle-level.

The emerging Distributed Ledger Technologies (DLTs) brought a new era of distributed peer-to-peer applications and guarantees trust among involved parties. The terms DLT and Blockchain will be used interchangeably throughout this paper, Blockchains are a type of DLTs, where chains of blocks are made up of digital pieces of information called transactions and every node maintains a copy of the ledger. In DLTs, the authentication process relies on consensus among multiple nodes in the network [2]. Each record has a timestamp and cryptographic sig-

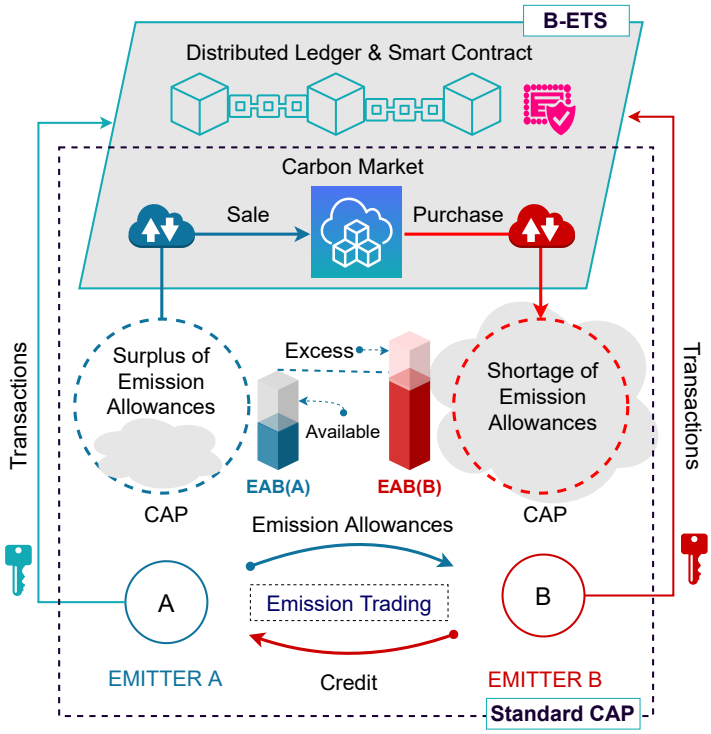


Fig. E.1: B-ETS general architecture.

nature; the system is secure and maintains a transaction ledger that is immutable and traceable. Ultimately, the goal of applying Blockchain technology to the transportation industry is to provide a fully distributed ETS system that can encourage direct communication between producers and consumers. A primary reason to embrace new DLTs is to bypass the administrative pitfalls that have plagued current emissions monitoring systems. Security is another aspect that motivates this approach. For instance, data pollution attacks are incredibly dangerous, these attacks typically occur in centralized systems and involve an adversary trying to modify the content of the packets and then forward the corrupted messages to neighboring nodes. The integration of Blockchain in individual carbon trading will accelerate the involvement of the public in carbon trading and sensitize society to individual level carbon footprints.

Current V2V approaches have limitations such as: the need for trusted third-party entities, security hardware, higher communication and storage overhead, high implementation costs, and issues related to the confidentiality of data. Studies [3], [4], [5] have strictly considered Vehicle-to-Infrastructure (V2I) approaches incorporating additional resources such as On-board Units (OBUs) and Roadside Units (RSUs). Eckert et al. develop a carbon Blockchain framework for Smart Mobility Data-Market as a trading system for CO_2 in the form of carbon tokens in [6]. The evaluation is done on the user and vehicular levels. Pan et al. outlined some advantages of the use of Blockchain in ETS namely safety and reliability, efficiency, convenience, openness and inclusiveness [7]. That work was not concerned with V2V networks or mobile carbon emissions trading, but it did introduce the concept of personal carbon emissions trading which

could be applied in vehicular networks.

In this study, we first tackle the challenges of the current EU-ETS system by proposing a distributed emissions allowance trading system called B-ETS. The system creates an account for the emissions generated from each vehicle and allows exchanges among vehicles in a trusted manner based on Blockchain and Smart Contracts. In B-ETS, each vehicle acts as a light client in the global Blockchain network and manages its own Emission Allowance Balance (EAB) which is reset at the beginning of each day. The EAB data is recorded transparently and immutably in the distributed ledger. It should be noted that we use one day as our unit of time without loss of generality. Any other unit (a week, a month) could be used if that seemed more suitable.

Then, we introduce an economic incentive-based mechanism which attracts drivers to change their driving behavior in order to reduce emissions. Each vehicle's generated emissions are calculated and the data are recorded immutably in the distributed ledger. If the emission level is higher than the defined threshold, the EAB will be reduced. If the EAB goes to zero, the driver needs to buy credits in the form of EAB from others.

The proposed V2V-based allowance trading system would not replace the in-service fleet-wide monitoring required by the EU-ETS plan. Rather, it would complement that plan by making it the responsibility of drivers to meet personal emissions targets. That is, without individualized feedback, drivers cannot measure the environmental impacts of their actions. Furthermore, without incentives, they might not be willing to contribute to environmental sustainability.

Given the proposed B-ETS system, vehicles participating in the program will be influenced by the economic incentive. Drivers are more prone to behave better when their EAB and driving privileges are at stake. If drivers contribute to lower emissions (i.e., demonstrate healthy driving habits), their EAB will increase or remain positive. Essentially, drivers want to avoid having to purchase credits from others or having a negative EAB balance as this could lead to driving restrictions.

Our mechanism can be compared to the traffic point penalty system in the U.S., Canada and other countries. As punishment for committing traffic violations, the drivers risk the suspension or revocation of their license based on a point-record mechanism in place. As a result, the Department of Motor Vehicles (DMV) can revoke the driver's license of that person and they are not allowed to drive any motor vehicle. In order to mitigate the social cost of license suspensions, point-removal systems exist for most point-record drivers licenses [8]. In contrast, our system proposes a daily (or weekly or other period as appropriate) record of associated driving behaviors with vehicle emissions data and individual accounts.

The execution of the smart contract guarantees trust among vehicles and driving habits, (e.g, avoid idling, speeding, etc) and CO_2 levels. Vehicles in the system are alerted via rules defined in the smart contract to reduce emissions [9] [3].

Our solution to reducing vehicle CO_2 emissions involves the use of DLT-enabled emissions monitoring, which could be applicable to any market worldwide. In this work, we focus on the EU, but, our method can comply with regulations in China and could be implemented in the US to measure life-cycle Corporate Average Fuel Efficiency (CAFE) standards.

The contributions of this study are described as follows:

- First, we propose a distributed Blockchain-based emission trading system named B-ETS that will meet the requirements of the EU-ETS plan for reducing vehicular emissions.

Table E.1: Nomenclature

Symbols	Descriptions
T	Considered system period [hours]
$\mathcal{V}$	Set of vehicles
i	Vehicle $i \in \mathcal{V}$
T_s	CO_2 sampling period
$\epsilon_i(t)$	Average CO_2 emissions per km for vehicle i at time t
$B_i(t)$	Emission allowance balance of vehicle i at time $t \in [0, T)$
$p_i(t)$	Penalty/tax for vehicle i at time t
$s_i(t)$	Incentive (subsidy) for vehicle i at time t
L_{total}	Total allowed latency
L_{trans}	Communication latency
L_{comp}	Blockchain verification latency
R	Communication data rate [packets/s]
$\mathbf{v}_i(t)$	Speed of vehicle i at time t [km/h]
S_B	Blockchain block size in bits
$\mathbf{v}_{ij}(t)$	Relative speed between i and j at time t
r_{ij}	Communication Range between i and j
$e_{i,j}(t)$	Allowances sold by j to i at time t
$\mathcal{T}$	Maximum allowed CO_2 emissions generated by vehicles per km.

B-ETS overcomes the disadvantages of current centralized ETS systems and provides a trustworthy approach for exchanging data in vehicle-to-vehicle networks.

- Second, we introduce an economic incentive-based system which motivates drivers to reduce fuel consumption and pollution. Based on the autonomous execution of smart contracts, the incentive mechanism is guaranteed to work in a trusted and distributed manner.
- Third, realizing the lack of communication and computation analysis in Blockchain-enabled vehicle networks, we present a theoretical model to derive the communication efficiency of the proposed system B-ETS.

The remainder of this paper is organized as follows. In the next section, we present the system model and analysis. In section III, the performance evaluation is outlined including our results. Finally, in section IV, we provide our conclusion and plan future work.

2 System Model and Analysis

2.1 Blockchain as a Ledger for VANET

The system operates within periods of duration T. In this section, we describe the two major system components: the vehicles and the distributed ledger, followed by the selected model for CO_2 emissions. Table E.1 presents the nomenclature used throughout the paper.

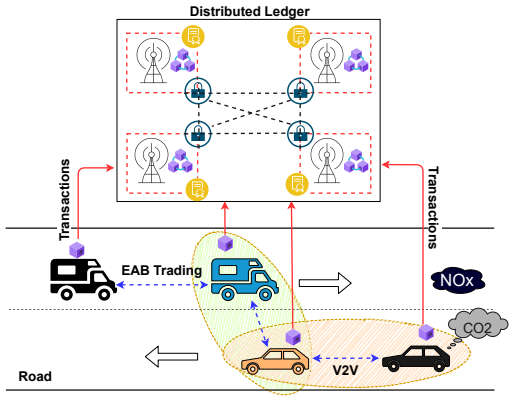


Fig. E.2: Blockchain-enabled vehicular emission trading system.

Vehicles

Let $\mathcal{V}$ be the set of vehicles in the system. An On-Board-Unit (OBU) is installed in each vehicle $i \in \mathcal{V}$ in the Blockchain-based VANET. The OBU performs light tasks, including collection and transmission data to other vehicles according to the IEEE 802.11p communication standard, and provides support to passengers and drivers.

Within each system period of duration T, the CO_2 emission monitoring system takes samples of the average CO_2 emissions per km in each vehicle i and updates the ledger. The CO_2 is sampled at fixed intervals of duration $T_s < T$ hours. The sample taken by vehicle i at time $t \in \{0, T_s, 2T_s, \ldots, T\}$ hours is denoted as $\epsilon_i(t)$ and consists of the taken measurement, the vehicle ID i, and a timestamp, generated as a function of t. The amount of CO_2 generated at the vehicles is reset to zero at the beginning of each period of duration T, hence, $\epsilon_i(0) = 0$.

Distributed Ledger

The distributed ledger records the data exchange history grouped into blocks and linked together chronologically. To minimize the cost of storage, the sensing data could be hashed and stored at more powerful nodes, and only the hash of data is recorded to the blockchain. Next, a confirmation message is sent back to confirm that the data has been added to the ledger as presented in Figure E.4. We assume that the data services (e.g., data storage, trading and task dispatching) are implemented on top of a permissionless Blockchain [10].

In a permissionless blockchain, any peer can join and leave the network at any time as a reader or writer. Permissionless Blockchains are open and decentralized with no central authority. Bitcoin and Ethereum are instances of permissionless Blockchains. In contrast, in permissioned Blockchains a central authority decides and attributes the right to individual peers to participate in the write or read operations of the blockchain. Examples of these include Hyperledger Fabric and R3 Corda [11].

The sensing data are formatted into transactions of fixed size. To enhance efficiency, only the digest of each transaction is stored on the chain, and the content of the transactions are stored by each consensus node off-chain or at the IPFS storage.

Emissions Model

The amount of CO_2 generated from vehicles depends on various factors such as: national average age distributions, vehicle activity speeds, operating modes, vehicle-miles traveled, starts and idling, temperatures, maintenance, anti-tampering programs, and average gasoline fuel properties in that calendar year [12]. The calculation of emissions in our simulations are based on the Handbook Emission Factors for Road Transport V3.1 (HBEFA), the model was implemented by extracting the data from HBEFA and fitting them to a continuous function obtained by simplifying the function of the power the vehicle engine must produce to overcome the driving resistance force [13].

2.2 Emission Allowances Trading

Traditional Cap-and-Trade

Traditionally, cap-and-trade commonly refers to governmental regulations and programs in place to limit the levels of CO_2 emissions as a result of industry activity. As briefly mentioned, the EU-ETS works on a cap and trade principle, where the cap is a dynamic limitation, set on the total amount of GHG emitted by installations covered by the system. Within the system, companies receive or buy emission allowances which can be traded. Although, vehicular emissions were not initially considered, in 2006, researchers at MIT joint program on the science and policy of global change introduced the implementation of a cap-and-trade policy for vehicles. Their central conclusion indicated that there are important efficiency gains to be realized by including transport emissions under the CAP and by integrating pre-existing programs, such as CAFE, and cap-and-trade systems [14].

B-ETS Framework

Our B-ETS framework considers an economy where vehicles produce goods over a system period $[0, T]$ hours. Therefore, each vehicle i acts as a wallet in the Blockchain network and its EAB at time t is denoted as $B_i(t) \in \mathbb{R}$. In the system, the updates to the EAB are triggered by the sampling of the CO_2 emissions of the vehicles, hence, the system operates at specific times $t \in \{0, T_s, 2T_s, \dots\}$. At the beginning of each period of duration T, the EAB of each vehicle i is reset to a pre-defined value $B_i(0)$. So, the EAB cannot be accumulated between subsequent periods. However, if i were to hold on to this initial allowance endowment until the end of the period, it would be able to offset the system's *cap* by up to $B_i(0)$ units of emissions credits. This is the *cap* aspect in our B-ETS scheme.

The EAB pertains to an individual account in which the allowances are used and exchanged amongst vehicles for environmental sustainability. In order to offset penalties, the vehicles with low balances may engage in buying allowances from vehicles that expect to meet demand with fewer emissions than their own cap. This is our *trade* aspect of B-ETS framework.

Remark 1: *A CAP program is only feasible in scenarios where the vehicles have a positive allowance balance at the beginning of the periods. Hence, the following inequality must hold:*

$$B_i(0) > 0 \qquad \text{for all } i \in \mathcal{V}. \tag{E.1}$$

The maximum allowed CO_2 emissions generated by vehicles per km is denoted as $\mathcal{T}$ (which is defined as a rule in smart contract). If $\epsilon_i(t) > \mathcal{T}$, then our initial smart contract is executed to

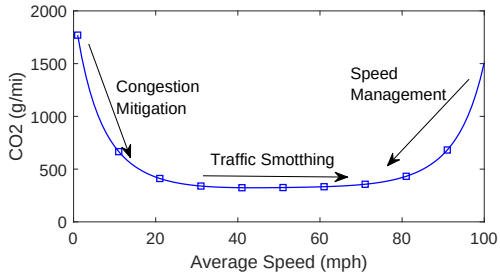


Fig. E.3: CO_2 emissions (grams/mile) as a function of average speed (mph) [15]

generate an alert to i to reduce vehicle speed as a direct solution to reduce amount of generated CO_2, and the fine $p_i(t)$ will be deducted from its EAB. In contrast, the subsidy $s_i(t)$ will be endowed to i for maintaining the CO_2 emissions below $\mathcal{T}$.

The values of $p_i(t)$ and $s_i(t)$ are considered as taxes and subsidies for vehicle i that depend on their behavior. The incentive may help encourage the driver to control their driving behavior to avoid generating CO_2 higher than the allowed standard. The driver needs to choose between receiving an incentive by reducing amount of emissions or being fined due to overloaded generated emissions. The penalties and subsidies are computed based on the theoretical model presented in [16] which depends on various vehicular factors.

In order to increase the subsidies and reduce the penalties, the drivers can follow strategies defined in smart contracts. For example, Figure E.3 shows that CO_2 is a function of average speed. First, we observe that very low average speeds generally represent stop and start driving periods, and vehicles traveling in short distances, in these cases, the emission rates are quite high. In this period, the smart contract defines rules to increase traffic speeds and reduce congestion by, for instance avoiding high traffic roads to reduce emissions. Second, when the speed of the vehicle is too high, it demands high engine loads which require more fuel, leading to higher CO_2 emission rates. The techniques to manage high speeds are implemented in the contracts which recommends the drivers to simply reduce their speeds. Consequently, moderate speeds of around 40 to 60 mph are ideal speeds which reduce emissions and will give the drivers incentive to improve their balances.

In addition, the EAB can be traded among vehicles based on predefined smart contracts. Whenever $B_i(t) < 0$, there will be a red alert issued to i for having a negative-balance. This alert is in the form of penalties, or restricted road access to zero-balance vehicles. In this cases, the vehicles can either wait until the next period for their EAB of to be reset or buy the EAB from other vehicles. We consider the case of vehicles exchanging EAB on-road via execution of smart contract and distributed ledger. For this, let $e_{i,j}(t)$ be the amount of allowances sold by vehicle j from vehicle i at time t. These operations are recorded in the distributed ledger.

Remark 2: *The vehicle j cannot sell more allowances $e_{ij}(t)$ than it actually owns. In other words, i cannot buy more than is actually available. Hence,*

$$e_{i,j}(t) \leq B_j(t), \text{for all } j \in \mathcal{V},\ t \in [0, T) \tag{E.2}$$

Operation

The operation of the vehicle's emission allowance trading is performed in the following steps:

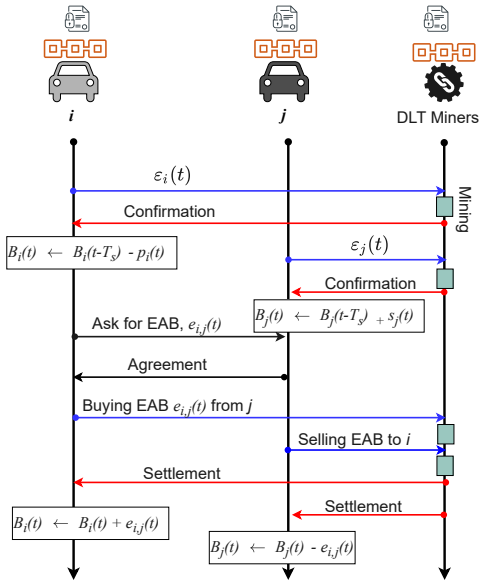


Fig. E.4: Communication System

Step 1. Publishing Data. Each vehicle $i \in \mathcal{V}$ computes its own average generated CO_2 emissions, namely, $\epsilon_i(t)$ as shown in Figure E.2 for $t \in \{0, T_s, 2T_s, \ldots, T\}$. These values are published to light ledger version of each vehicle and synchronized with the full ledger stored in DLT full nodes.

Step 2. Emission Control. The generated CO_2 emissions data is recorded in the ledger, and the smart contract with the predefined rules is executed. These rules are characterized by two categories namely maximum CO_2 emissions and actions: warnings, alerts and reminders. The published CO_2 data is formatted and arranged into blocks to be verified through a consensus process. If $\epsilon_i(t) > \mathcal{T}$, the smart contract issues an alert message to i to control its driving behavior and $p_i(t)$ is deducted from $B_i(t)$ via smart contract. Hence, the ledger is updated with the value $B_i(t) \leftarrow B_i(t - T_s) - p_i(t)$. In contrast, if j has maintained a safe speed and emitted reasonable amounts of CO_2, it received an incentive $s_j(t)$ to its balance. Hence, the ledger is updated with $B_j(t) \leftarrow B_j(t - T_s) + s_j(t)$.

Step 3. Emission Allowance Trading. After receiving a confirmation with the required action from the smart contract, if $B_i(t) < 0$, then i needs to re-charge its EAB by buying emission allowances from other vehicles. For example, i makes an agreement with j to buy an amount of emission allowances $e_{i,j}(t)$. Then, i sends the buying request for the amount $e_{i,j}(t)$ to execute a smart contract. Next, j updates the smart contract with a selling request and $e_{i,j}(t)$.

Step 4. Settlement. Finally, the EAB of each vehicle is updated and settled as $B_i(t) \leftarrow B_i(t) + e_{i,j}(t)$ and $B_j(t) \leftarrow B_j(t) - e_{i,j}(t)$.

In this paper, we focus on the efficiency of V2V communication between vehicles for exchanging data and trading EAB. We study these in terms of end-to-end latency which includes the transmission latency among vehicles and computation latency of Blockchain validation processes.

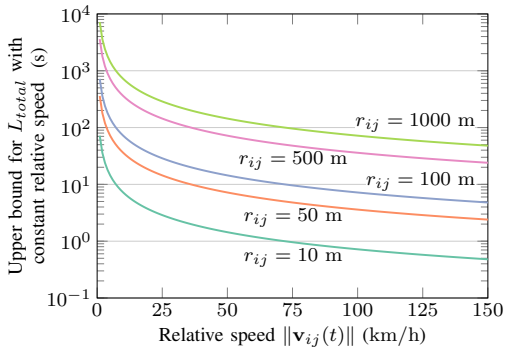


Fig. E.5: Upper bound of total latency L_{total} for communication between vehicles.

2.3 Joint Communication and Computation Model

In this section, we define the total available time for communication between two vehicles and the impact of the Blockchain computation latency.

Let $(x_i(t), y_i(t))$ denote the position of vehicle i at time t. If communication is initiated at time t, the time in which two vehicles, namely i and j, are available for communication is defined by 1) their communication range r_{ij} 2) their positions $(x_i(t), y_i(t))$ and $(x_j(t), y_j(t))$, 3) their relative speed, given by vector $\mathbf{v}_{ij}(t) = \mathbf{v}_i(t) - \mathbf{v}_j(t)$ km/h. Clearly, to initiate communication at time t, the distance between the vehicles must be

$$d_{i,j}(t) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} \leq r_{ij}. \tag{E.3}$$

Then, the total time for V2V communication between vehicles i and j at time t is given as

$$L_{total}(t) = \max_{\ell \in \mathbb{R}} \{\ell \,|\, d_{i,j}(\ell) \leq r_{ij}\} - t. \tag{E.4}$$

It is immediate to see that $L_{total} \to \infty$ when $\|\mathbf{v}_{ij}(t')\| \to 0$ for all $t' \in [t, \ell]$. This implies that whenever both vehicles move in the same direction and with near equal speed, they will have a long time L_{total} to communicate and exchange messages. Furthermore, it can be seen that, the upper bound for L_{total} seconds for the case where the relative speed $\mathbf{v}_{ij}(t')$ km/h remains constant for all $t' \in [t, \ell]$ is

$$L'_{total} \leq \frac{r_{ij}}{1.8\|\mathbf{v}_{ij}(t)\|} \tag{E.5}$$

Figure E.5 illustrates the upper bound for L_{total} with several values of $\|\mathbf{v}_{ij}(t)\|$.

The time needed to complete a trade between two vehicles i and j in B-ETS can be divided into two parts. First, the communication between vehicles, simply denoted as L_{trans}, and, second, the time needed for the verification process in the distributed ledger, denoted as L_{comp}. Hence, a trade is completed successfully if and only if

$$L_{total} \geq L_{trans} + L_{comp}. \tag{E.6}$$

From there, we define the probability of successful data trading as

$$P_{success} = \Pr\left(L_{comp} + L_{trans} \leq L_{total}\right) \tag{E.7}$$

The latency for the communication between vehicles i and j, denoted simply as L_{trans}, is a function of the amount of data that must be exchanged and the effective data rate selected for communication R in packets per second. The data that must be exchanges is defined by the block size of the Blockchain, denoted as S_B. On the other hand, the effective data rate R is determined by the implemented protocol, the wireless conditions (e.g., fading, noise, interference, and number of active devices), and the modulation and coding scheme; where the latter determines the instantaneous data rate. The implemented protocol for communication is the IEEE 802.11p standard and the wireless environment are given in Section 3. Nevertheless, we can approximate the latency for communication by assuming that the effective data rate remains constant throughout the trade as

$$L_{trans} \approx \frac{S_B}{R}. \tag{E.8}$$

The formulations to calculate L_{comp} are presented in the following.

Blockchain computation latency

We consider a Blockchain-based VANET network that includes a subset of vehicles $\mathcal{M} \subseteq \mathcal{V}$ that work as miners. These miners start their Proof-of-work (PoW) mechanism computation at the same time and keep executing the PoW process until one of the miners completes the computational task by finding the desired hash value [17]. When a miner i executes the computational task for the POW of current block, the time period required to complete this PoW can be formulated as an exponential random variable W_i whose distribution is $f_W(w,i) = \lambda_c e^{-\lambda_c w}$, in which $\lambda_c = \lambda_0 P_c$ presents for the computing speed of a miner, P_c is power consumption for computation of a miner, and λ_0 is a constant scaling factor. Once a miner completes its PoW, it will broadcast messages to other miners, so that other miners can stop their PoW and synchronize the new block.

For the PoW computation, we are interested in finding the time in which the first miner $i*$, among all the $M = |\mathcal{M}|$ miners, finds out the desired hash value. This is the time for the fastest PoW computation among miners and denoted by the random variable W_{i*}. By assuming $\{W_i\}$ are i.i.d. random variables, we can calculate the complementary cumulative probability distribution of W_{i*} as

$$\begin{aligned}\Pr(W_{i*} > w) &= \Pr\left(\min_{i\in\mathcal{M}}(W_i) > w\right) = \prod_{i\in\mathcal{M}} \Pr(W_i > w) \\ &= (1 - \Pr(W_i < w))^M, \text{ s.t. } i \in \mathcal{M}.\end{aligned} \tag{E.9}$$

Hence, L_{comp} is the average computational latency of the fastest miner $i*$, calculated as

$$L_{comp} = \int_0^\infty (1 - \Pr(W_i \leq w))^M w = \int_0^\infty e^{-\lambda_c M w} w \tag{E.10}$$

Now we can calculate the communication latency as $L_{trans} + L_{comp}$.

Note that it can occur that the communication delay exceeds the available communication time L_{total}. In such a case, a proposed transactions with potentially valid PoW solution must be abandoned. Hence, finding a valid puzzle solution does not guarantee that the proposed transactions will be finally accepted by the network because of the propagation delay. In such cases, a Blockchain fork can only be adopted as the canonical Blockchain state when it is first

Table E.2: Smart Contract execution cost

Smart Contracts	Gas	Ether	USD
UserAuthority	159430	$15.9{\cdot}10^{-5}$	0.0723
RecordData	152443	$15.2{\cdot}10^{-5}$	0.0692
AlertControl	213924	$21.3{\cdot}10^{-5}$	0.0971
Incentive	224934	$22.4{\cdot}10^{-5}$	0.1021
RecordData	276394	$27.6{\cdot}10^{-5}$	0.1254
EABTransfer	246374	$24.6{\cdot}10^{-5}$	0.1118

* 1 Ether = 10^9 Gwei; 1 USD = 4,182,471.9949 Gwei

disseminated across the network. In scope of this research, to simplify, we do not address the problem of fork, please refer to [18] for more detail.

3 Performance Evaluation

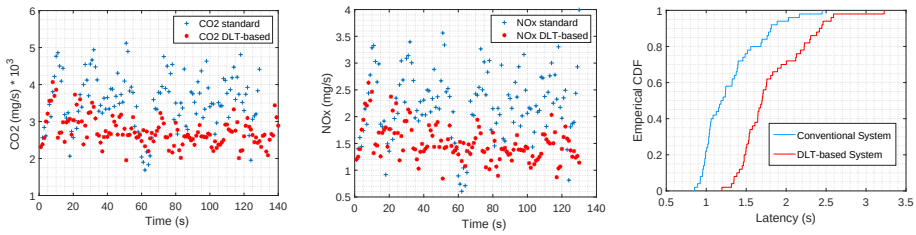
In this section, we analyze the performance of our proposed B-ETS system.

In order to emulate a realistic vehicle network as presented in Figure E.2, a combination of micro simulators, network libraries and open-source vehicular network simulators is employed. Specifically, SUMO [13], OMNET++ which runs in parallel via a proxy TCP connection, and Veins. The IEEE 802.11p standard is used for communication between vehicles and a simple path loss model is selected. In each simulation, 120 vehicles are generated and located randomly. The CO_2 emissions are calculated reading the Traffic Control Interface (TraCI) commands from SUMO. Ethereum is deployed as a ledger in the experiments by using local Ganache platform.

The computational efforts to execute smart contracts in Blockchain are measured in units of gas. The currency for Ethereum is Ether (ETH). In our simulations, the transaction costs and execution costs are converted to ETH and USD, see Table E.2. The ETH gas station was used to estimate the costs, the price is generated using a static average of 20 Gwei, where one Ether is equivalent to 10^9 Wei. The transaction costs are the costs associated with sending the contract codes to the Ethereum blockchain, dependent on the size of the contract.

The amount of CO_2 generated from vehicles is dependent upon various factors such as: speed, age of vehicles, etc. We ran two separate experiments to compare the amount of emissions generated between a standard CAP system and a Blockchain-based system when the driving behavior is controlled. Figure E.6 illustrates the generated CO_2 and NO_x, along with the V2V communication latency for the standard and the DLT-based trading. In the DLT-based trading, vehicles follow defined rules such as dropping their speed in the smart contract. In Figure E.6 we observe that the amount of CO_2 and NO_x generated from DLT-based system is lower than conventional system. These results prove that our system has the ability to reduce the overall CO_2 emitted from vehicles on the network.

In B-ETS, the transactions exchanged between vehicles are encrypted, and verified before attached in the distributed ledger. Therefore, the trusted recording and trading data is guaranteed in comparison with standard system. However, because of extra verification steps in Blockchain, the time to complete a transaction between vehicles is higher. This is a trade-off between trust



(a) CO2 Emission **(b)** NOx Emission **(c)** V2V communication latency

Fig. E.6: Performance Evaluation. (a) and (b): The CO_2 and NOx emission generated in standard and DLT based systems; (c) Communication latency between standard and Blockchain-based system.

and latency in Blockchain-based systems.

4 Conclusion

In this paper, we first proposed a Blockchain-based Emission Trading System, called B-ETS, to support the accounting and monitoring of emissions in vehicular networks. B-ETS provides a trustworthy and transparency for accounting the emissions generated from vehicles. The vehicles can exchange their emission allowances through autonomous smart contracts in a trusted manner. We introduce an economic incentive scheme based on smart contracts to encourage drivers to behave in environmentally friendly ways.

This work provides a mechanism for policy makers, vehicle manufacturers and the EU-ETS to enforce the carbon emissions regulations in a more efficient, secure manner as well as to perform full life-cycle analysis of vehicles. Using the proposed method could result in vehicle manufacturer savings, ensuring that they are not subject to excess emissions fees at the end of the year through the continuous monitoring and reporting of CO_2.

The next stage of this work involves further analysis of the current system in two ways. First, we will include the analysis of more pollutants such as Particulate Matter (PM_x), Carbon Monoxide (CO), Sulfur Dioxide (SO_2) into B-ETS. Then, we will address the limitations of this work by diversifying the vehicles on the network, thereby incorporating other types of vehicles (other than passenger vehicles), such as: buses, vans and trucks.

5 Acknowledgment

This work has been in part supported by the European Union's Horizon 2020 program under Grant 957218 IntellIoT, the Independent Research Fund Denmark (DFF) under Grants Nr. 8022- 00284B (SEMIOTIC) and Nr. 9165-00001B (GROW), and the National Science Foundation Graduate Research Fellowship under Grant DGE-1839285

References

[1] E. Commision, "Road transport: reducing CO2 emissions from vehicles," 2015, (Accessed on 10/28/2020).

[2] L. D. Nguyen, A. E. Kalor, I. Leyva-Mayorga, and P. Popovski, "Trusted wireless monitoring based on distributed ledgers over NB-IoT connectivity," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 77–83, 2020.

[3] F. Zheng, J. Li, H. van Zuylen, and C. Lu, "Influence of driver characteristics on emissions and fuel consumption," *Transportation Research Procedia*, vol. 27, pp. 624–631, 2017.

[4] M. Alsabaan, K. Naik, and T. Khalifa, "Optimization of fuel cost and emissions using V2V communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1449–1461, 2013.

[5] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.

[6] J. Eckert, D. López, C. L. Azevedo, and B. Farooq, "A blockchain-based user-centric emission monitoring and trading system for multi-modal mobility," *arXiv preprint arXiv:1908.05629*, 2019.

[7] Y. Pan, X. Zhang, Y. Wang, J. Yan, S. Zhou, G. Li, and J. Bao, "Application of blockchain in carbon trading," *Energy Procedia*, vol. 158, pp. 4286–4291, 2019.

[8] G. Dionne, J. Pinquet, M. Maurice, and C. Vanasse, "Incentive mechanisms for safe driving: a comparative analysis with dynamic data," *The review of Economics and Statistics*, vol. 93, no. 1, pp. 218–227, 2011.

[9] EEA, "Do lower speed limits on motorways reduce fuel consumption and pollutant emissions?" https://www.eea.europa.eu/themes/transport/speed-limits-fuel-consumption-and/, 4 2019, (Accessed on 10/28/2020).

[10] L. D. Nguyen, I. Leyva-Mayorga, A. N. Lewis, and P. Popovski, "Modeling and analysis of data trading on blockchain-based market in iot networks," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[11] K. Wust and A. Gervais, "Do you need a blockchain?" in *Proceedings IEEE Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2018, pp. 45–54.

[12] EPA, "Estimated U.S. average vehicle emissions rates per vehicle by vehicle type using gasoline and diesel | bureau of transportation statistics," https://www.bts.gov/content/, 6 2018, (Accessed on 10/28/2020).

[13] D. Krajzewicz, M. Behrisch, P. Wagner, R. Luz, and M. Krumnow, "Second generation of pollutant emission models for sumo," in *Modeling mobility with open data*. Springer, 2015, pp. 203–221.

[14] A. D. Ellerman, H. D. Jacoby, and M. B. Zimmerman, "Bringing transportation into a cap-and-trade regime," 2006.

[15] A. Cappiello, I. Chabini, E. K. Nam, A. Lue, and M. Abou Zeid, "A statistical model of vehicle emissions and fuel consumption," in *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*. IEEE, 2002, pp. 801–809.

[16] D. Fullerton and S. West, "Tax and subsidy combinations for the control of car pollution," National Bureau of Economic Research, Tech. Rep., 2000.

[17] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[18] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

Paper F

Analysis of Distributed Ledger Technologies for Industrial Manufacturing

Authors
Lam Duc Nguyen, Arne Bröring, Massimo Pizzol, and Petar Popovski

The paper has been submitted for publication
Nature Scientific Reports 2022.

Abstract

In recent years, industrial manufacturing has undergone massive technological changes that embrace digitalization and automation towards the vision of intelligent manufacturing plants. With the aim of maximizing efficiency and profitability in production, an important goal is to enable flexible manufacturing, both, for the customer (desiring more individualized products) and for the manufacturer (to adjust to market demands). Manufacturing-as-a-service can support this through manufacturing plants that are used by different tenants who utilize the machines in the plant, which are offered by different providers. To enable such pay-per-use business models, Distributed Ledger Technology (DLT) is a viable option to establish decentralized trust and traceability. Thus, in this paper, we study potential DLT technologies for an efficient and intelligent integration of DLT-based solutions in manufacturing environments. We propose a general framework to adapt DLT in manufacturing, then we introduce the use case of shared manufacturing*, which we utilize to study the communication and computation efficiency of selected DLTs in resource-constrained wireless IoT networks.*

1 Introduction

Industrial Internet of Things (IIoT) is a recent concept that gained traction with the emergence of the wireless 5G technology and it is already exhibiting a great impact within the manufacturing domain [1]. The general trend is to embrace digitalization and automation towards manufacturing plants that act as cyber-physical systems. This results in an increasing number of smart devices with sensors and actuators that are being integrated in industrial automation processes. In parallel, local edge computing infrastructures are being built up in manufacturing plants, which provide resources for advanced computing and henceforth the basis for next generation IIoT applications [2]. The key economic driver behind this technological evolution is the increase in the production flexibility. This allows for smaller lot sizes and more individualized products for customers. These trends are supported by business models, such as manufacturing-as-a-service, where manufacturing facilities are utilized more flexibly by numerous tenants who utilize the machines in the plant, which are offered by different providers. These economic forces drive manufacturing plants towards an increase in technological complexity and require improvements in system reliability, intelligence, and trustworthiness during operation [3]. Especially the opening of the manufacturing plant's ecosystem to a diverse set of involved parties poses many challenges for manufacturing enterprises to satisfy the trust requirements of multi-partner collaboration [4].

Distributed Ledger Technology (DLT) can be used to address those trust and privacy challenges in the manufacturing environment of the future, e.g., to transparently store machinery's usage data as a basis for pay-per-use business models on the manufacturing shop floor. A DLT is a distributed ledger of transactions—rather than being kept in a single, centralized location, the information is held by all the nodes of a network [5]. In general, all these network nodes have copies of the same ledger. This removes the need for a third-party to assure that rules are being implemented correctly, instead, this is implicitly done through a decentralized system. Although the most widely known instance of DLT is blockchain, and, specifically, bitcoin, the transactions on a DLT do not have to be financial. In essence, a transaction simply represents a change in state for whichever data point the DLT's stakeholders want to track. DLTs are driven

by consensus: when a node or a *DLT-client* initiates a transaction, its details are broadcast to the entire network, checked by other nodes and accepted if there is consensus. *DLT-clients* are considered as lightweight devices which have limited resources and just initiate transactions, as well as transmit transactions to *DLT-managers* to validate. Once a transaction has been validated, it is bundled with other transactions into a block of data. Each block is secured via a cryptographic algorithm. This results in a unique signature for each block known as a hash. These blocks are then ordered sequentially into a chain of blocks, with each block also containing the previous block's hash [6]. This makes it extremely difficult to tamper with a block, as altering a single piece of data would result in a different hash value, making it evident to the DLT's users and causing the transaction to be rejected.

In short, DLTs allow the storage of transactions in immutable records and every record is distributed across many nodes. Thus, security in DLTs comes from the decentralized operation, but also from the use of strong public-key cryptography and cryptographic hashes. The key benefits of the integration of DLTs into manufacturing systems are: i) auditable and guaranteed immutability and transparency for stored data (e.g., machine usage data, sensing data about machine conditions, or logs about user/technician engagements), ii) no need for a third party to assure the rules between the different parties in the manufacturing ecosystem are met, iii) enabling high security and privacy of information in manufacturing networks, which is urgently needed as more than 25% of cyber attacks will involve IoT [7].

To showcase these benefits and to have a realistic use case as an example for our studies, we implement in this work the DLT-based application of *shared manufacturing*, which relates to the economic driver of 'flexible production'. Specifically, a robot arm, as part of a production cell with multiple machines, is offered by a provider, who allows different tenants of the plant the usage of the robot arm, while expecting a usage fee. In this application, the DLT is required to capture the usage times of the robot arm through the various tenants, which is then the basis for a correct billing and payment for usage time. Some parts of this process can be done automatically with smart contracts. These involve two entities turning a business contract into code that recognizes actions on the DLT. For example, a smart contract might recognize that a rental of a machine from "provider A" to "customer B" on a certain date for a specific time period should be for a specific price [8]. This simplifies processes that take significant time to check. This structure gives DLT participants confidence in their transaction without the need to trust each other. Nor do they need to agree on a trusted third party to make sure they're both following the rules. Because the ledger of transactions is consensus-based and distributed, records stored in it cannot be erased or changed.

To be able to implement the above described application and reap the described benefits, the system designs of current manufacturing plants need to be adjusted to be able to accommodate the operation of a DLT and overcome certain limitations: First, today's computation infrastructures of industrial manufacturing plants are typically designed as centralized systems, where cloud services perform data aggregation and analysis [9]. While the manufacturing infrastructure comprises a multitude of IoT devices and sensors that collect data and have only little computing power, the gathering and processing of data in a centralized cloud service may lead to network overload and single points of failure [10]. To setup a DLT network in such an environment, a sufficient amount of local computing capacity [11] needs to be available and, potentially, edge computing facilities can be integrated in the computation infrastructure. Furthermore, industrial communication systems have been traditionally designed for reliable operation in a noisy factory environment, employing mainly wired and proprietary communication

technologies to connect sensors, actuators, and controllers. Nevertheless, with the emergence of IIoT, future factories will increasingly rely on diverse communication technologies, including wireless standards, to ensure reliability, interoperability, and remote operation and control of production processes through the Internet. These wireless links are potentially less reliable and are more constrained, which needs to be considered, when operating a DLT network. From these limitations regarding the system infrastructure, we derive the key *research question* of this work: *"What is the computation and communication overhead that results from the operation of a DLT network in a manufacturing environment?"* The answer to this question will be critical to understand for future research on applications of DLT in manufacturing, as well as for practitioners who want to deploy a DLT network in a manufacturing plant.

The use of DLT in manufacturing has received attention from both academia and industry because of its promise for easing supply chain and manufacturing operation management problems due to its advantages in transparency, traceability, and security. In industry, Bosch increasingly connects their products to the IIoT in order to directly participate in the digital economy. The goal is to build an *Economy of Things*, which will be based on DLT [12]. Another example is a concrete solution by Siemens, which enables their *Mindsphere* IIoT platform to track products of the food and beverage industry transparently throughout their entire life cycle based on DLT [13]. *MindSphere* exploits all useful information before forwarding only a crucial subset to the distributed ledger. The DLT then makes sure the collected data is safe and transparently accessible to everyone who is part of the ecosystem. In academia, Li et al. [14] introduced a distributed P2P system that improves the security and scalability of the cloud-based manufacturing platform based on DLT. Danzi et al. [15] analyze the communication aspects in terms of delay and overhead between IoT devices and Blockchain network. The authors demonstrate that, if the statistics of account updates and the channel state are known, the lightweight IoT clients can construct a list of events of interest that provides a predictable average communication cost. In addition, a survey [16] about performance of different Blockchains is conducted, but the work mainly focuses on theoretical aspects, and lacks a detailed analysis in specific application areas such as manufacturing. Fu et al. [17] presented an innovative environmentally sustainable DLT-energized strategy for the fashion apparel manufacturing industry. Yu et al. [18] proposed a DLT-based service composition architecture for manufacturing. In general, the public DLT-based applications are characterized by the distinctive metric of computational trust.

In order to be able to answer our research question, we extend state-of-the-art through the following research contributions:

- General analysis of different DLTs and their capabilities when used in industrial manufacturing environments;
- System design for DLT-based IIoT manufacturing systems that can integrate and adapt multiple features and components;
- Evaluation of communication and computation overhead of different DLTs in resource-constrained IoT networks. This benchmark of different DLTs for manufacturing scenarios will help interested parties to understand the trade-offs in DLT-based systems.

The remainder of this paper is organized as follows. In the next section, we present the results of this study. First, we present a general analysis of five different DLT platforms. Second, we introduce a system design for using DLT in industrial manufacturing. Third, we implement

Table F.1: Comparison of different enterprise DLT platforms

	Hyper. Fabric [19]	**Quorum** [20]	**Ethereum** [21]	**IOTA** [22]	**Solana** [23]
DLT type	Private	Private	Public / Private	Public / Private	Public / Private
Goals	Open DLT framework	Open, based on Ethereum	Broad ecosystem	Lightweight	High scalability
Application	Enterprise DLT	Enterprise DLT	DApps	IoT	DApps
Governance	Linux Foundation	ConsenSys	Ethereum Foundation	IOTA Foundation	Solana
Currency	N/A	N/A	Ether (ETH)	MIOTA	SOL
Consensus	Pluggable	Voting Protocol	PoW	Tangle	PoH
Smart Contract	nodejs, go, java	Solidity	Java or Kotlin	Solidity	Rust
Throughput	∼2000 tps	∼100 tps	∼100 tps	1000∼1500 tps	∼1400 tps
Latency	∼250 ms	∼414 ms	∼2150 ms	∼ 258 ms	∼ 500 ms

the shared manufacturing use case and perform a performance evaluation of the five different DLTs. Finally, we discuss our findings and indicate avenues for future research.

2 Results

In this section, we first study five different DLT platforms, then we propose a general framework to integrate DLT in manufacturing. Finally, we implement the use case of shared manufacturing and conduct the evaluation.

2.1 Analysis of DLTs for Industrial Manufacturing

Although a large number of DLTs are available, within the scope of whic work we have selected five representative DLT platforms that are either already used or appear as most promising for manufacturing environments: Hyperledger Fabric, Ethereum, Quorum, Solana, and IOTA. The overview comparison of these DLTs is shown in Table F.1.

Each DLT can be categorized as public, private, or hybrid, where the latter one can support features of both public and private ones. *DLTs* allow any user to pseudo-anonymously join the DLT network and do not restrict the rights of the nodes on the network. We are investigating in this paper the public DLTs Ethereum [21], IOTA [22], and Solana [23]. However, for the implementation of our use case within a manufacturing plant such public DLTs are used in a

private deployment by installing local networks. In contrast, private (or permissioned) DLTs restrict access to their network to certain nodes and may also restrict the rights of nodes on the network. In this paper, we are investigating the private DLT platforms Hyperledger Fabric [19] and Quorum [20]. The identities of the users of a private DLT are known to the other users of that private DLT. In a Hybrid DLT, every transaction can happen quickly in its own private chain and commits to the public chain only happen as and when necessary, e.g., when public verification is required. This provides the immutable trust from the public Blockchain as well as the scaling from private DLTs. Layer 2 solutions and side-chains [24] are variations of this concept.

Besides their type, the five DLTs have different goals and applications in focus. Hyperledger Fabric and Quorum are both aiming to offer a open foundation for new components to build a broad ecosystem that supports enterprises with various functionalities to deploy their own private DLT. Ethereum has a large community of developers and already an established ecosystem that focuses on decentralized applications (DApps), e.g., for decentralized finance. Solana follows a similar application focus, while aiming for higher scalability than Ethereum. IOTA's focus is on IoT applications and therefore aims to support DLT participants with a small footprint.

IIoT applications in the manufacturing environment will involve many stakeholders with different roles, functionalities, and information with access rules, identities and security factors. An important factor to provide security is the support to validate transactions generated by participating nodes. While Hyperledger Fabric and Quorum are solely for the private setups, Ethereum, IOTA and Solana are designed for public networks, but can also be configured for private purposes. In terms of security, public networks can show certain advantages over private ones, especially if they are able to provide transparency and distributed storage. For example, in a public DLT, the data is encrypted and stored in all the devices, which makes it transparent. Besides, the more users a public/permissionless DLT has, the more secure it is. However, for enterprise use (i.e., also for typical manufacturing scenarios) public DLTs are not ideal as companies deal with highly sensitive data and cannot allow anyone to join their network. Also, private DLTs provide very low or no fees for validation and a faster consensus process. However, a private DLT can be altered by its owners, making it more vulnerable to hacking [25]. Besides, only Hyperledger Fabric supports by default data confidentially via in-band encryption and guarantees the privacy of data by creating private channels (e.g., to setup for departments within an organization). Therefore, Hyperledger Fabric allows for authorization with trusted Certificate Authority per channel. These features are vital in a trusted IoT system for enterprises.

Each DLT platform deploys a different consensus mechanism. Ethereum uses the Proof-of-Work (PoW) consensus that requires involved parties of a network to expend effort solving a mathematical puzzle to prevent anybody from gaming the system. PoW consensus consumes significant computing and energy resources, which is not suitable for resource-limited systems. Quorum, as an enterprise version of Ethereum, uses a voting-based consensus protocol. This consensus protocol achieves consensus on transactions and key network decisions by counting the number of votes cast by nodes on the networks and not consuming more energy for verification as compared to PoW. IOTA uses "little" PoW for preventing spamming attacks. In Ethereum, doing PoW is to receive the power to define the truth, the node with more power can solve the PoW faster and consume more energy. Meanwhile, IOTA use PoW with lower difficulty to prevent spamming and to allow transactions to be attached in the Tangle. Hyeperledger Fabric modularized the consensus part among distributed peers in an ordering service [19],

so that this platform allows users to choose their preferred algorithm, e.g., CFT (crash fault-tolerant) or BFT (byzantine fault-tolerant) ordering. Finally, Solana introduces a new consensus algorithm called Proof-of-History which allows timestamp field to be built into the blockchain itself instead of using values of timestamps as PoW DLTs.

Table F.1 specifies the smart contract programming language supported by the DLT. Smart contracts act as autonomous entities on the ledger that deterministically execute logic expressed as functions of the data that are written on the ledger. Therefore, smart contracts can be established to have automatic reactions from the DLT network to specific events. For example, in the use case of shared manufacturing, smart contracts can be used for restricting, tracking and payment for the usage of the rented machinery. The smart contract feature is currently supported by Ethereum, Solana and Hyperledger Fabric (called 'Chaincode'). In IOTA, a smart contract is called Quobic and its development is still in progress.

Furthermore, Table F.1 states the performance characteristics of the DLTs. These values have been acquired both through our own experiments and the data available in the literature. These measures are of vital importance for IoT applications, particularly in manufacturing, where a large number of sensors may generate millions of data points per day. This requires high efficiency of the consensus mechanism, including the way in which transactions are processed by the peers, known as *endorsing peers* in Hyperledger Fabric, validators in Solana, and full nodes (peers) in Ethereum. Specifically, we have Solana, with 600 nodes and around 1000 validators. Currently Solana is hosting around 340 apps [26]. Meanwhile, Ethereum has over 3000 Dapps running on its network. Regarding latency, the transaction confirmation time must be sufficiently short to avoid queuing in the DLT and to ensure consistency in the ledgers. The confirmation time of an Ethereum transaction in a public network is around 25 seconds in public networks. This value indicates that consensus over public networks may not be suitable for real-time IoT applications. However, other DLT platforms can achieve much lower confirmation times [27]. Note that in Table F.1 the transaction confirmation time is included in the end-to-end latency, which however does not account for the communication latency at the radio access network.

Another important performance feature of the DLTs is related to the CPU usage and resulting energy consumption. The idea that DLT technologies and crypto-assets consume an excessive amount of electricity has been at the heart of recent discussions around this technology. The energy consumption of a DLT protocol should not be equated with its environmental footprint. Indeed, many use cases related to DLT technologies and crypto-assets may even contribute to improving the environmental footprint, in particular by using the surplus of decarbonised energy in certain geographical areas where the need for electricity is lower than the level of production [28]. In the scope of this work, we study the carbon footprint of the different selected DLT platforms within the local testbed of the shared manufacturing use case.

The charge of fees to process the transactions, commonly known as *gas* is yet another factor to take into account to select the appropriate DLT. These may greatly increase the operational costs of the network, which negatively impacts the throughput of the DLT. On the one hand, transaction fees pose a problem in massive IIoT scenarios if the generation of a large number of transactions is essential. On the other hand, these fees may contribute to minimizing the amount of redundant transactions generated by the sensors, which in turn offloads the DLT. In industrial manufacturing domain, the required fee for generating transactions within a company or among some cooperative organizational setup may not be suitable. In addition, businesses have always required a reasonable degree of privacy as well as control over their networks, so that publishing

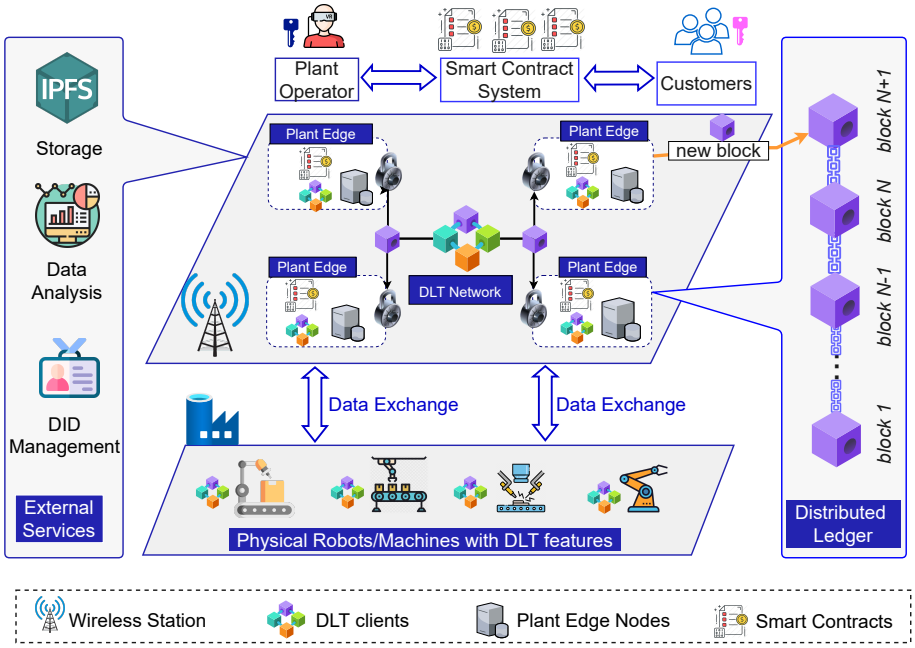


Fig. F.1: Overview of the system design

the data on a Blockchain is not reasonable and potentially unsafe. Therefore, we consider only private Blockchains for the enterprise scenario.

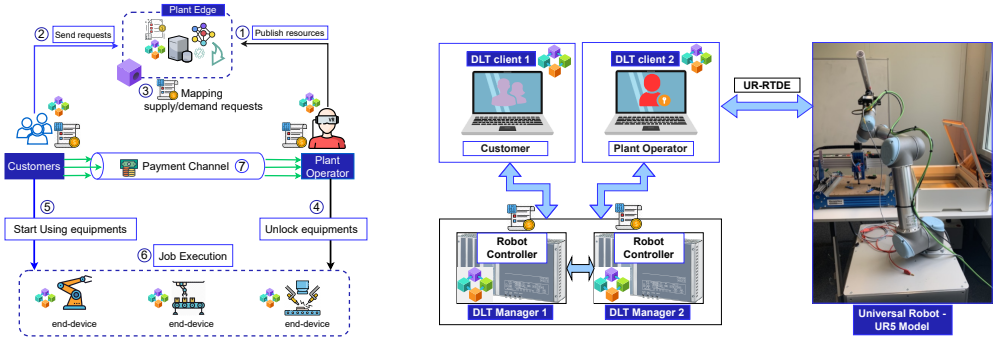
2.2 System Design for using DLT in Industrial Manufacturing

The proposed system design is described in Figure F.1. It comprises four key parts as described below.

DLT System: This component includes all modules to build various features of DLT technologies such as consensus, smart contract, data authorization, identity management, and peer-to-peer (P2P) communication. These components must ensure that every change to the ledger is reflected in all copies in seconds or minutes and provide mechanisms for the secure storage of the data generated by IoT devices and parameter configurations. There are numerous DLTs with different characteristics that may be beneficial for different target applications. The DLT nodes can be located everywhere and connected with base stations via the Internet.

Physical Machines: This component consists of physical robots, machines, and IoT sensor devices which collect the data and publish to the distributed ledger for accounting or analyzing purposes.

Plant Edge System: Even though DLT-based solutions offer significant countermeasures to secure data from tampering and support the distributed nature of the IoT, the massive amount of generated data from sensors and the high energy consumption required to verify transactions make these procedures unsuitable to execute directly on resource-limited IoT devices. Instead, edge servers with high computation resources can be used to handle real-time applications and to further increase the degree of privacy (e.g., through cloud computing) [29]. The edge network is



(a) Shared Manufacturing communication workflow

(b) Testbed set up with UR5 robot arm [30]

Fig. F.2: System model of shared manufacturing and local test-bed setup

a potential entity to cooperate with the DLT network in computationally heavy tasks and return the estimation results (e.g., from solving proof-of-work (PoW) puzzles, hashing or algorithm encryption) to the DLT network for verification.

External Services: The devices of the manufacturing environment are typically resource-constrained with limited storage space and low computation capacity. Hence, external infrastructure which operates on the edge may be incorporated to provide external services, such as storage and computing. For example, the Interplanetary File System (IPFS) is a distributed file storage system that can store data generated from IoT networks and return a hash to the ledger based on the content of the data. Since the ledger cannot handle and store the massive amount of manufacturing data collected by the sensors, machines, and robots, the service provided by the IPFS is a vital component. In addition, a Digital Identity Management (DID) could be added to support managing identity of participant devices in a distributed manner.

2.3 Performance Evaluation of DLTs in a Shared Manufacturing Use Case

In this section, we analyze the application of the Shared Manufacturing use case and study its performance. The application uses DLT to automate the management of rentals of industrial robots, where the manufacturing plant operators and their customers can make agreements without third parties and the associated delay.

Along with data sharing [31] and vehicle sharing [32], the machine sharing concept in industry manufacturing has been recently identified as a key innovation for implementing the next industrial evolutionary step [18]. Open and shared manufacturing factories are composed of a number of industrial robots and other production machines that can be rented by customers. The advantage over traditional manufacturing plants is that such plants can have a higher workload and less idle periods, which in turn can make the production cheaper. Therefore, production tasks need to be efficiently allocated on the available machine resources under consideration of system performance.

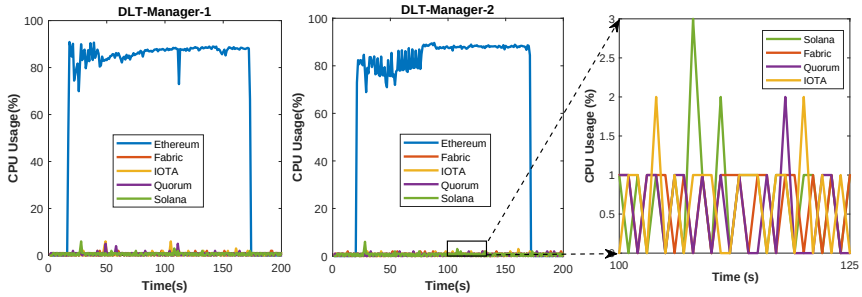
Our shared manufacturing application scenario is described in Figure F.2a. As an initial step (1), the plant operator of a factory publishes the list of machine resources which are available for rent. Thereby, each machine has a unique ID and described capabilities to perform specific jobs. A *manufacturing marketplace* running on a DLT-based network can be implemented in

Table F.2: Testbed Settings

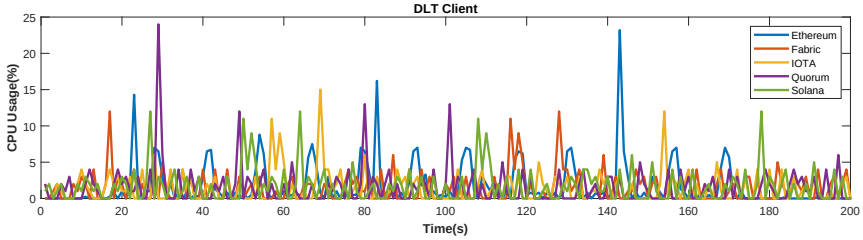
	DLT-manger 1	***DLT-manager 2***	**DLT-Client 1**	**DLT-client 2**
Devices	Siemens Microbox	Laptop	Rapsberry Pi 3+	Raspberry Pi 3+
RAM	4GB	8 GB	1 GB	1GB
Connectivity	Ethernet	Ethernet	Wifi	Ethernet
Capacity	Intel(R) Core i7-351UE CPU @ 1.70GHz x4 GHz	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.9 GHz	Quad Core 64 bit ARM cortex at 1.2 GHz	Quad Core 64 bit ARM cortex at 1.2 GHz

such an environment to offer access to those machine descriptions. In the DLT-based manufacturing network, smart contracts are running to receive requests from customers rent machines (step 2) and match them to resources offered by the plant operator (step 3). In addition, the rules and agreements, e.g., about the rent period, specific tasks, or payment methods between plant operator and customers are pre-defined in the smart contracts and executed autonomously. The customers can check the list of available machines published by the plant operator, and if the customers have a relevant job coming up, they can request the suitable machines via smart contracts. This is the first difference between the DLT-based and non-DLT shared manufacturing system. In a non-DLT based system [33], a plant operator and customers could not work directly by exchanging messages without the guarantee about the trust of contracts as well as payment. This guarantee requires a third-party to complete the deal. After DLT-based smart contracts executed and mapped the requests from plant operator and customers, the plant operator account will unlock automatically the available machines (step 4) and assign the control of the machines to customers. Then, the customers can start control and program the machines for their jobs (step 5), which are then executing these jobs (step 6). Compared to standard shared manufacturing, the second innovation in DLT-based systems is that we implemented a layer 2 *payment channel* [34] between the plant operator and customer for micro-payments (step 7).

To study the communication and computation overhead resulting from DLT in manufacturing systems, we have implemented the above described *shared manufacturing* application in a private setup as shown in Figure F.2b. The setup involves the DLT components *DLT-manager 1 and 2* and *DLT-clients*. The *DLT-manager* has a high computation capacity and enough storage for a full ledger with all the information and data. The *DLT-clients* are lightweight and are limited in terms of computation and resources. The *DLT-clients* can query and access the data from the ledger without downloading the full chain of blocks. The *DLT-managers* are implemented in two different equipments: as a Siemens Microbox [30] and a Macbook Pro. The *DLT-clients* are implemented in Raspberry Pi 3+. The specifications of these devices are found in Table F.2. *DLT-manager-1* and *DLT-manager-2* are connected via Ethernet, and communicate with DLT-clients via local WiFi. The communication method can be extended to other long-range communication or global internet depending on specific scenarios. The distributed ledger is deployed in the DLT Managers. We have implemented five types of *DLTs*, namely Ethereum,



(a) Energy Consumption of *DLT-managers* running full DLT features in separated scenarios.



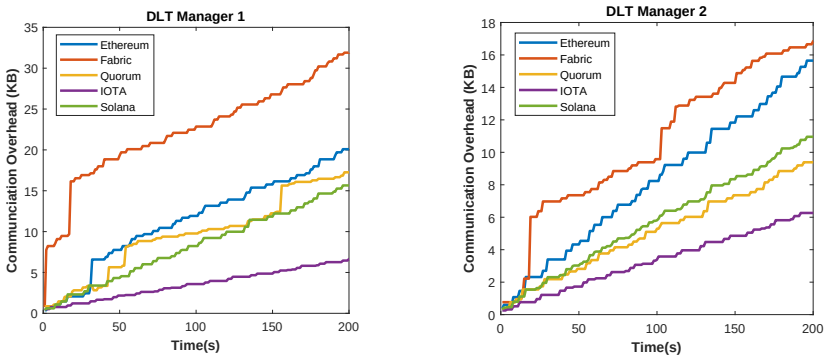
(b) Energy consumption of *DLT-clients* which transmit data to *DLT-managers* and synchronize data when needed.

Fig. F.3: Computation overhead of each network component of the 5 studied DLTs namely Ethereum, Hyperledger Fabric, IoTA, Quorum, and Solana.

Quorum, IOTA, Hyperledger Fabric, and Solana.

During our evaluation, the DLT-client sends 10 transactions per second to the ledger, which is hosted by the DLT-manager-1 and -2. By recording the CPU usage percentage of the DLT-specifc processes, we have observed the computation overhead in each case of the 5 selected DLT platforms. Looking at the DLT Managers, we have found Ethereum as an outlier, as it requires by far the most computation time of around *85%* of CPU as shown in Figure F.3a due to the usage of the Proof of Work (PoW) for the consensus and verification process. The *non-PoW* DLTs, Solana, Hyperledger Fabric and Quorum, require in our private network setting only around *1-3%* CPU usage in both *DLT Manager 1* and *DLT Manager 2*. Similarly, the IOTA platform uses *PoW* only rarely in order to prevent spam attacks, so the CPU usage of the DLT Managers is relatively low, similar to Hyperledger Fabric and Quorum. On the DLT Client component, the CPU usage is primarily the generation and transmission of transactions, so that these DLTs require around 5-10% CPU usage of the Raspberry Pi.

Figure F.4a and F.4b show the communication overhead of the five different DLTs in our shared manufacturing setup. The Hyperledger Fabric produces more network traffic than the others on the DLT Managers. The reason is that the network architecture of Hyperledger Fabric is optimized for an enterprise environment with high security requirements, where the raw data need to be formatted for signed transaction proposals, then going through the complex endorsement and validation process, before attachment to the Blockchain. This process introduces more communication overhead. IOTA produces the lowest traffic on the DLT Managers thanks to the design based on the Tangle [22]. Specifically, the interconnected Tangle infrastructure does not require total verification across the whole ledger. Instead, all parties are verifying simultaneously and, as a result, the energy and time required to complete transactions are shortened. In



(a) Communication overhead of *DLT-manager-1*.

(b) Communication overhead of *DLT-manager-2*.

Fig. F.4: Communication Overhead comparison of the 5 studied DLTs namely Ethereum, Hyperledger Fabric, IoTA, Quorum, and Solana.

addition, Tangle's verification process purports to ensure that there are no duplicate transactions that would lead to double-spending. On the *DLT-clients*, the communication overhead is mainly coming from publishing the collected data via formatted transactions to DLT managers. In specific, a single transaction in IOTA consists of *2673 trytes* which is equivalent to *1589 bytes*, if encoded, a Ethereum transaction includes around *109 bytes* of header, and no limited metadata , Hyperledger Fabric transaction sizes depends on the type of transactions, for example, *3.06 kB* for spend and *4.33 kB* for mint [19]. The overhead of a Solana transaction includes *64 bytes* signature and maximum *1232* bytes for given metadata.

Based on the CPU usage and the utilized computing hardware, we determined the energy consumption of the five selected DLT platforms and computed carbon footprint. We assume that electricity for running the computational operations is consumed and produced in Germany. As a measure of carbon intensity of the German energy mix, calculated in a life cycle perspective, we used data from the life cycle database ecoinvent v.3 cutoff system model [35]. In particular, the dataset *Market for electricity, low voltage, DE* was chosen, which represents an average low-voltage energy mix for Germany. We obtained the life cycle impact of producing *1 kWh* electricity according to this version of the database via the software *SimaPro* and using the default IPCC Global Warming Potential (GWP) method with a time horizon of 100 years [36]. This resulted in a value of global warming impact of *0.540 kg CO_2-eq / kWh* that represents the impact of all greenhouse gases emitted in the electricity production process and upstream activities in a life cycle perspective. This value was further used to calculate the total carbon footprint of the computation based on its energy requirements. The results are shown in Table F.3. We observe that the annual CO_2 generated through PoW consensus is significantly higher than that of non-PoW Blockchains. The private Ethereum DLT produced around *26692* $\cdot 10^{-6}$ *kg CO_2-eq/hour*, which is equivalent to the average of *4.3* charged smartphones [37]. This compares to around *203* $\cdot 10^{-6}$ *kgCO_2-eq/hour*, *211 kgCO_2-eq/hour*, and *198 kg CO_2-eq/hour* from Hyperledger Fabric, Quorum, and IOTA, and Solana respectively. Note that all results are extrapolated to the utilization of our private DLT setup for the shared manufacturing application over an operation day.

Table F.3: Carbon FootPrint of private DLT testbed with 5 *DLTs* calculuated in Germany market running per hour

Platform	**Power of Machine (kW)**	**Energy consumed on Average (kWh)**	**Avg. CPU Usage for Blockchain Operation (%)**	**Energy Consumed for Blockchain operation (kWh)**	**Greenhouse gas (GHG) emission in DE (kg CO_2-eq/kWh)****	**GHG emission per blockchain operation (kg CO_2-eq)**
Hyper.Fabric	0.06	0.06	0.625%	$375 \cdot 10^{-6}$	0.540	$203 \cdot 10^{-6}$
Ethereum	0.06	0.06	82.35%	$49392 \cdot 10^{-6}$	0.540	$26682 \cdot 10^{-6}$
Quorum	0.06	0.06	0.65%	$390 \cdot 10^{-6}$	0.540	$211 \cdot 10^{-6}$
IOTA	0.06	0.06	0.61%	$366 \cdot 10^{-6}$	0.540	$198 \cdot 10^{-6}$
Solana	0.06	0.06	0.61%	$366 \cdot 10^{-6}$	0.540	$198 \cdot 10^{-6}$

3 Discussion

We have seen that deploying a DLT in an industrial manufacturing environment allows to realize novel business cases. Choosing the right DLT platform for industrial use cases, such as the one we elaborated above, is challenging, as their are many options available. Therefore, we have conducted here an evaluation of five of the most popular and promising DLT platforms and proposed how to integrate those into a physical manufacturing system.

A clear observation is that Ethereum is an outlier in terms of CPU usage, due to its PoW consensus algorithm, which of course also results in high energy consumption. Therefore, we can conclude that Ethereum and other PoW DLTs should, in general, not be used in the envisioned manufacturing environments. In order to still be able to use many of Ethereum development tools, a plant operator can use Quorum, which is an enterprise version of Ethereum. Both Quorum and Hyperledger Fabric show a similar performance in our local evaluation regarding CPU usage. However, Hyperledger Fabric introduces higher communication overhead as compared to Quorum. Therefore, in an environment with communication restrictions, the operator could opt for Quorum out of these two by simply looking at slight performance advantages. IOTA, which is specifically designed for IoT networks, requires the lowest CPU and communication overhead and can hence be favoured by an operator that has strong requirements in this regards. However, IOTA's smart contract mechanism is still under development and also the tooling support is not as strong. Finally, Solana is a public Blockchain network with a focus on achieving high scalability. In our local network, Solana performed similarly to Quorum in terms of communication overhead as well as CPU usage.

The results measured from our local experiments can be considered as a benchmark regarding sustainability aspects in specific shared manufacturing use cases. In the scope of this research, we evaluated the greenhouse gas emission per Blockchain operation based on energy consumed by Blockchain activities. For example, IOTA foundation provided an energy benchmark for the IOTA network, which shows results that are similar to our experimental results [38]. In terms of Hyperledger Fabric, we have used Hyperledger Caliper for the benchmark evalua-

tion [39]. Referring to prior research [40], the energy consumed by beyond-PoW blockchains, such as Polkadot [41], Cardano [42], or Hedera Hashgraph [43], is within a range that is similar to the energy consumed in our experimental setup.

Looking towards the future, we see many benefits for the use of DLT in manufacturing, enabling a broad range of use cases and business models. The vision at the horizon is a truly *collaborative industrial IoT* in which *things* (such as machines in a manufacturing plant) ubiquitously and automatically interact without intervention of humans. This is fuelled by the capability to autonomously make (micro-)payments. This would empower devices, e.g., to rent cloud server capacity for additional computational capacity when required, to pay directly to other devices for access to the Internet, or automatically pay for electricity consumed. The current payment systems are not well suited for massive-scale micro-transactions due to high transaction costs and limited capacity. This calls for a vision of payments between things, which will be small per transaction, but autonomous and running efficiently at a massive scale.

4 Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957218 (Project IntellIoT)

References

[1] B. Kafle, B. S. Goraya, S. Mack, F. Feldmann, S. Nold, and J. Rentsch, "Topcon–technology options for cost efficient industrial manufacturing," *Solar Energy Materials and Solar Cells*, vol. 227, p. 111100, 2021.

[2] B. Soret, L. D. Nguyen, J. Seeger, A. Bröring, C. B. Issaid, S. Samarakoon, A. El Gabli, V. Kulkarni, M. Bennis, and P. Popovski, "Learning, computing, and trustworthiness in intelligent iot environments: Performance-energy tradeoffs," *IEEE Transactions on Green Communications and Networking*, 2021.

[3] M. Helu, K. Morris, K. Jung, K. Lyons, and S. Leong, "Identifying performance assurance challenges for smart manufacturing," *Manufacturing letters*, vol. 6, pp. 1–4, 2015.

[4] J. Al-Jaroodi and N. Mohamed, "Blockchain in industries: A survey," *IEEE Access*, vol. 7, pp. 36 500–36 515, 2019.

[5] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[6] N. Alrebdi, A. Alabdulatif, C. Iwendi, and Z. Lian, "Svbe: searchable and verifiable blockchain-based electronic medical records system," *Scientific Reports*, vol. 12, no. 1, pp. 1–11, 2022.

[7] "Top 10 security predictions 2016," https://www.gartner.com/smarterwithgartner/top-10-security-predictions-2016, (Accessed on 12/06/2021).

[8] L. D. Nguyen, S. R. Pandey, S. Beatriz, A. Broering, and P. Popovski, "A marketplace for trading ai models based on blockchain and incentives for iot data," *arXiv preprint arXiv:2112.02870*, 2021.

[9] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, "Edge computing in iot-based manufacturing," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 103–109, 2018.

[10] W. H. Hassan *et al.*, "Current research on internet of things (iot) security: A survey," *Computer networks*, vol. 148, pp. 283–294, 2019.

[11] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[12] "Economy of things | bosch global," https://www.bosch.com/research/know-how/success-stories/economy-of-things-a-technology-and-business-evolution/, (Accessed on 12/06/2021).

[13] "Blockchain iot | exclusive content for the food and beverage industry | siemens global," https://new.siemens.com/global/en/markets/food-beverage/exclusive-area/blockchain-iot.html, (Accessed on 12/06/2021).

[14] Z. Li, A. V. Barenji, and G. Q. Huang, "Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform," *Robotics and computer-integrated manufacturing*, vol. 54, pp. 133–144, 2018.

[15] P. Danzi, A. E. Kalør, C. Stefanović, and P. Popovski, "Delay and communication tradeoffs for blockchain systems with lightweight iot clients," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2354–2365, 2019.

[16] C. Fan, S. Ghaemi, H. Khazaei, and P. Musilek, "Performance evaluation of blockchain systems: A systematic survey," *IEEE Access*, vol. 8, pp. 126 927–126 950, 2020.

[17] B. Fu, Z. Shu, and X. Liu, "Blockchain enhanced emission trading framework in fashion apparel manufacturing industry," *Sustainability*, vol. 10, no. 4, p. 1105, 2018.

[18] C. Yu, L. Zhang, W. Zhao, and S. Zhang, "A blockchain-based service composition architecture in cloud manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 7, pp. 701–715, 2020.

[19] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[20] A. Baliga, I. Subhod, P. Kamat, and S. Chatterjee, "Performance evaluation of the quorum blockchain platform," *arXiv preprint arXiv:1809.03421*, 2018.

[21] D. Vujičić, D. Jagodić, and S. Ranđić, "Blockchain technology, bitcoin, and ethereum: A brief overview," in *2018 17th international symposium infoteh-jahorina (infoteh)*. IEEE, 2018, pp. 1–6.

[22] S. Popov and Q. Lu, "Iota: feeless and free," *IEEE Blockchain Technical Briefs*, 2019.

[23] A. Yakovenko, "Solana: A new architecture for a high performance blockchain v0. 8.13," *Whitepaper*, 2018.

[24] C. Sguanci, R. Spatafora, and A. M. Vergani, "Layer 2 blockchain scaling: a survey," *arXiv preprint arXiv:2107.10881*, 2021.

[25] K. Wüst and A. Gervais, "Do you need a blockchain?" in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 45–54.

[26] "Homepage | solana docs," https://docs.solana.com/, (Accessed on 01/14/2022).

[27] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[28] A. Ahl, M. Yarime, M. Goto, S. S. Chopra, N. M. Kumar, K. Tanaka, and D. Sagawa, "Exploring blockchain for the energy transition: Opportunities and challenges based on a case study in japan," *Renewable and sustainable energy reviews*, vol. 117, p. 109488, 2020.

[29] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 33–39, 2018.

[30] "Ur5 collaborative robot arm | flexible and lightweight cobot," https://www.universal-robots.com/products/ur5-robot/, (Accessed on 01/26/2022).

[31] L. D. Nguyen, I. Leyva-Mayorga, A. N. Lewis, and P. Popovski, "Modeling and analysis of data trading on blockchain-based market in iot networks," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6487–6497, 2021.

[32] D.-M. Storch, M. Timme, and M. Schröder, "Incentive-driven transition to high ride-sharing adoption," *Nature communications*, vol. 12, no. 1, pp. 1–10, 2021.

[33] "Renting machine made easy – mcpond," https://mcpond.com/, (Accessed on 12/07/2021).

[34] M. Jourenko, K. Kurazumi, M. Larangeira, and K. Tanaka, "Sok: A taxonomy for layer-2 scalability related protocols for cryptocurrencies." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 352, 2019.

[35] G. Wernet, C. Bauer, B. Steubing, J. Reinhard, E. Moreno-Ruiz, and B. Weidema, "The ecoinvent database version 3 (part i): overview and methodology," *The International Journal of Life Cycle Assessment*, vol. 21, no. 9, pp. 1218–1230, 2016.

[36] T. F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. K. Allen, J. Boschung, A. Nauels, Y. Xia, B. Bex, and B. Midgley, "Ipcc, 2013: climate change 2013: the physical science basis. contribution of working group i to the fifth assessment report of the intergovernmental panel on climate change," 2013.

[37] "Greenhouse gas equivalencies calculator | us epa," https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator, (Accessed on 12/07/2021).

[38] "Energy benchmarks for the iota network (chrysalis edition)," https://blog.iota.org/internal-energy-benchmarks-for-iota/, (Accessed on 02/11/2022).

[39] "Hyperledger caliper – hyperledger foundation," https://www.hyperledger.org/use/caliper, (Accessed on 02/11/2022).

[40] M. Platt, J. Sedlmeir, D. Platt, P. Tasca, J. Xu, N. Vadgama, and J. I. Ibañez, "Energy footprint of blockchain consensus mechanisms beyond proof-of-work," *arXiv preprint arXiv:2109.03667*, 2021.

[41] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," *White Paper*, vol. 21, pp. 2327–4662, 2016.

[42] "Cardano | home," https://cardano.org/, (Accessed on 02/11/2022).

[43] L. Baird, M. Harmon, and P. Madsen, "Hedera: A governing council & public hashgraph network," *The trust layer of the internet, whitepaper*, vol. 1, pp. 1–97, 2018.

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-894-6

AALBORG UNIVERSITY PRESS