Incentivizing Crowdsensing in IoT through Micropayments: a Simulation Study

Luca Serena
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
luca.serena2@unibo.it

Pietro Manzoni
Computer Engineering Department
Universitat Politècnica de València
Valencia, Spain
pmanzoni@disca.upv.es

Moreno Marzolla
Gabriele D'Angelo
Department of Computer Science and Engineering
Center for Inter-Department Industrial Research ICT
University of Bologna
Bologna, Italy
moreno.marzolla@unibo.it
g.dangelo@unibo.it

Stefano Ferretti
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
s.ferretti@unibo.it

Abstract

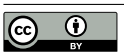
Many IoT applications rely on the timely collection and processing of data. Although the technology that allows data collection is readily available, its deployment on a large scale raises many practical issues, the main one probably being: who pays for the infrastructure? IoT deployment could significantly benefit from crowdsourcing: to ensure the successful deployment of a data collection infrastructure, it is crucial to motivate end users, as their active participation and engagement are key to the system's effectiveness and reach. This paper addresses these issues in a practical use case that deals with collecting environmental data. We propose a decentralized crowdsensing architecture where vehicles act as data collectors and transfer data from sources (e.g., sensing devices) to networked access points. The system provides economic incentives to individuals willing to act as data collectors or to operate gateways in the form of micropayments enabled by a blockchain. The proposed architecture is evaluated using a multilevel simulation model that combines existing communication, mobility, and behavioral sub-models. This reduces the time required to build a full simulator and potentially increases the accuracy of the results.

CCS Concepts

• **Computer systems organization** → **Sensor networks**; • **Computing methodologies** → **Modeling methodologies**; • **Information systems** → **Collaborative and social computing systems and tools**.

Keywords

Crowdsensing, Multilevel modeling, LoRaWAN, Blockchain, IoT



SAC '25, March 31-April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/10.1145/3672608.3707814

ACM Reference Format:
Luca Serena, Pietro Manzoni, Moreno Marzolla, Gabriele D'Angelo, and Stefano Ferretti. 2025. Incentivizing Crowdsensing in IoT through Micropayments: a Simulation Study. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31-April 4, 2025, Catania, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3672608.3707814

1 Introduction

Modern society is based on data: financial transactions, environmental monitoring, and social interactions produce huge amounts of data that need to be collected, stored, and analyzed. In this paper, we focus on environmental monitoring due to its increasing importance for reducing resource usage and improving yields in agriculture, reducing pollution, addressing the challenges posed by climate change, and improving the quality of the environment.

Environmental monitoring relies on different kinds of data, such as the concentration of pollutants in the air/land, environmental parameters such as temperature and humidity, precipitation rate, noise levels, and so on. Furthermore, this information should be collected not only within urban environments but also in the countryside, including remote areas where network connections are poor or non-existent. Although large-scale data gathering can sometimes be achieved through Earth-orbiting satellites, this requires substantial upfront investments; furthermore, some types of environmental data (e.g., noise level) can not be collected by satellite.

Luckily, technology provides some building blocks that can be used to deploy a large-scale environmental monitoring infrastructure with limited investment. Indeed, cheap sensors combined with communication-enabled micro-controllers are at the heart of the Internet of Things (IoT) that enables environmental data to be collected and transmitted [1].

A key challenge in developing IoT systems lies in efficiently retrieving and transmitting the information generated by the sensors and securely storing such data. Thus, with the growing popularity of IoT, there is a rising demand for robust and scalable solutions capable of managing vast amounts of information with particular regard for transmit energy consumption, transmission

delay, and network traffic. To address these challenges, several technologies have been adopted, such as 6G networks [9], edge computing [7] and LPWAN networks like Long Range Wide Area Network (LoRaWAN).

LoRaWAN is a wireless communication protocol designed for long-range communication and low-power operations in IoT devices. It operates on the LoRa physical medium, enabling devices to communicate over considerable distances while consuming minimal power. Besides these motivations, LoRa is particularly useful in areas where traditional communication infrastructures (such as Wi-Fi or LTE) are unavailable, since it does not rely on Internet connectivity or mobile phone towers. In addition, LoRa requires fewer gateways to cover large areas, as a single LoRaWAN gateway can manage thousands of sensors, reducing overall infrastructure costs. Finally, LoRa is optimised for intermittent transmission of small data packets, making it ideal for IoT applications that require periodic transmission of sensor readings.

Despite the availability of the building blocks above, an important issue must be addressed: who pays for the infrastructure? This question arises because it is usually assumed that the stakeholders themselves own the IoT devices.

Economic rewards may be put in place to encourage individuals and companies to participate in the process of data collection [8]. This approach is known as *crowdsensing*, and allows the burden of deploying the sensors and managing the infrastructure to be distributed across many volunteers that, e.g., deploy sensors or operate communication gateways upon compensation for their effort. Defining the details of the economic incentives is beyond the scope of this paper. However, they likely involve frequent micropayments, possibly in the form of "tokens" that may be converted into real money later. For example, users who deploy sensors might be compensated depending on the amount of data they contribute to the system. In contrast, those who operate relay nodes might be compensated depending on the amount of data transferred.

Traditional payment systems (e.g. bank transfers, e-wallets) may not be suitable for this scope due to the high fees and strict technical requirements imposed on the end nodes. Blockchain, originally intended as the technology for cryptocurrencies, allows financial transactions to be stored in a decentralized manner. Blockchains can be developed in various ways, but in all the implementations, data are inserted into containers known as blocks, which are logically linked together to form a chain. In all the blockchains the various active participants agree on the state of the distributed ledger through a consensus protocol, which allows them to determine the validity of the transactions. Examples of consensus protocols are Proof of Work (PoW) where participants solve complex computational puzzles to validate transactions and add new blocks to the blockchain, Proof of Stake (PoS) where the chances for the nodes to be the next validators depend on their stake, and Proof of Authority (PoA) where block validators are pre-selected and approved by a central authority or consortium [14]. Interesting properties of blockchain systems are transparency, traceability, data immutability, non-repudiation of transactions, and lack of bottlenecks or single points of failure. They allow users to reduce the costs for exchanging money and tokens, as third-party intermediaries such as banks are not involved [10]. Furthermore, blockchain can raise the trust among the involved parties, providing verifiable proof that a transaction has occurred at a specific time in the network [15].

While the first generation of blockchains only allowed users to exchange cryptocurrencies, blockchains 2.0s are well-suited for IoT applications thanks to the presence of smart contracts, which are actual contracts written in code [17], that enable automatized payments triggered by user-defined conditions. Therefore, blockchain systems can form the basis for a marketplace where customers can subscribe to data flows.

In this paper, we consider a specific use case for the scenario above, where vehicles equipped with sensors record local parameters such as air temperature, humidity, concentration of pollutants, and so on. This is motivated by the need to have a wider data coverage compared to static sensors placed in fixed locations. Moreover, by outsourcing sensor maintenance to drivers, the need for large-scale upkeep in the field is avoided, reducing the risks of damage, theft, or malfunction. The location/time-referenced data are periodically sent to a central repository using LoRaWAN technologies so that customers can access the collected information for further analysis. Incentives to people installing sensors on the vehicles, the communication infrastructure (e.g., the LoRaWAN access points), and the central data store are provided in the form of virtual tokens that are stored in a blockchain. The ultimate source of these virtual tokens is the customers, which are requested to pay a fee to access the information.

The proposed architecture is evaluated through simulation. Given that the system includes several components (e.g., mobility, distributed ledger, LoRaWAN) each of which is quite complex, the development of an accurate simulation model from scratch is time-consuming. A further contribution of this paper is to show how to integrate existing simulators that describe certain aspects of interest, thus enabling the comprehensive simulation of the entire system. The proposed simulation architecture is based on the concept of *multilevel modeling*, a M&S paradigm where a complex model is built upon existing sub-models. In the scenario considered in this work, the building blocks are sub-models that simulate the movement of sensor-equipped vehicles, LoRa physical medium, and a blockchain system. Multilevel modeling involves more than the well-known decomposition principle that is widely applied in software engineering. Indeed, it allows parts of the system to be represented at different levels of detail either in time or space [22, 23].

This paper is structured as follows: in Section 2 we review the state of art on integrating LoRa with blockchain technologies and related simulation studies; Section 3 describes the IoT architecture for crowdsensing proposed in this paper; Section 4 describes the multilevel simulator and the software integration process; Section 5 discusses the experiments carried out with the simulator, and Section 6 provides some conclusive remarks.

2 Related Works

The design of IoT architectures that incorporate LoRaWAN and blockchain have been the subject of intense research activity. The integration of LoRaWAN and the blockchain can be carried out through various approaches, depending on the diverse roles that LoRaWAN devices may assume within the distributed ledger [19].

In [15], the authors propose to build the blockchain system within the network servers layer, since gateways are in general resource-constrained and deployed outdoors, and so they are unsuited for storing data and performing the computations required by blockchain activities. On the other hand, in [5], the authors propose to build the blockchain within the gateway level, claiming that modern platforms such as Raspberry Pis have enough computing power to cope with blockchain functionalities. In [6] the authors design a four-layered blockchain platform to collect and trade weather data. In detail, the *governance layer* is responsible for deploying and maintaining validators (i.e., those who evaluate the quality of data) and creating bounties; the *data storage layer* stores the hash of the data and their score on-chain; the *oracle layer* retrieves the weather data, managing the encryption and the validation of the data; finally, the *marketplace layer* manages payments from customers to the publishers. In [3] the authors describe LoRaCHainCare, a LoRa-based blockchain system for healthcare monitoring. In the proposed system, LoRa is used to transmit data from the sensors/edge layer to the fog layer, consisting of a distributed network of LoRa gateways. The gateways in turn communicate with the cloud layer, comprising the LoRa server, Join server, and application server, through the Internet. In [18] the authors propose a smart city data marketplace where the distributed ledger employed is IOTA[1], thus allowing for lowering the costs and increasing the scalability. IOTA is based on a directed acyclic graph (DAG), where each transaction references and validates previous transactions, thus resulting in higher scalability and faster confirmation times as the network grows. In this scenario, there is no need for nodes acting solely as validators, therefore eliminating the need for fees to confirm transactions. The application proposed in the paper provides functions to i) find the devices connected to the marketplace, ii) retrieve the data of the specific devices a client is subscribed to, iii) list all the transactions from the clients to the accounts that provide data, iv) receive periodically the data from the sensors whose clients are subscribed.

Some studies explore the use of LoRa using mobile gateways instead of fixed ones. One of the advantages of mobile gateways is that the network coverage can be extended to remote rural areas. In [11], the authors designed and simulated a Smart Livestock Monitoring System where a single mobile gateway moves along the livestock area in order to cover the whole space to monitor. The experiments proved how the mobile gateway can lower capital expenditure with respect to a configuration where multiple static gateways are used. Mobility also plays an important role in emergency management systems, particularly in regions where cellular connectivity is limited or completely missing. In [20], the authors proposed a multi-hop strategy where survivors relay messages among peers through LoRa technology until they reach the rescue personnel. In this setup no gateways exist, and the P2P overlay is supposed to enlarge the communication range.

Many simulators of LoRa and blockchain technologies exist. Tools like OMNeT++[2] and ns-3[3] provide modules to simulate LoRa, allowing users to customize physical layer attributes like bandwidth and transmission power. However, these tools are quite complex and heavy-weight and are primarily intended to be used in stand-alone models rather than to cooperate with other tools. Fortunately, more lightweight LoRa simulators also exist; for example, LoRaSIM [26] has been used to test various LoRaWAN configurations with optimized gateway placement, while LoRaWANsim [12] allows users to set the locations of both nodes and gateways. When it comes to the blockchain, most simulation packages are primarily concerned with the analysis of peer-to-peer overlays, the use of smart contracts, and sometimes with the security aspects. Examples include Simblock [2], which utilizes an event-driven approach to model the selection of neighbor nodes in the peer-to-peer overlay, VIBES [24] enabling large-scale peer-to-peer network simulation for exploring network metrics, and SIMBA [4] that can accurately reproduce block validation dynamics.

[1]https://www.iota.org/
[2]https://omnetpp.org/
[3]https://www.nsnam.org/

3 System Architecture

The system discussed in this paper is based on the LoRaWAN communication protocol for transmitting sensor data, and a blockchain for realizing a marketplace. The use of these technologies aims to facilitate data trading within smart city frameworks, with a focus on enhancing energy efficiency and reducing infrastructure costs. LoRaWAN has emerged as a standardized solution for sensor data exchange in IoT scenarios, providing an efficient, scalable, and secure approach to enable connectivity and data transmission across various IoT deployments. On the other hand, private blockchains enable the automation of frequent micropayments common in publish-subscribe patterns, significantly reducing infrastructure costs compared to traditional payment channels. Moreover, private blockchains enable advanced and customized management of various data operations through the use of smart contracts and promote decentralization while allowing organizations to retain complete control over the system.

The case study considered in this paper involves a marketplace for the distribution of environmental data to paying customers, although we allow different marketplaces to coexist. Environmental data are collected opportunistically by vehicles that are equipped with LoRa-capable sensors. The data are uploaded periodically through LoRaWAN to a central repository so that it can be accessed by customers upon payment of some fee. Customers may include researchers, meteorologists, public entities in charge of monitoring air quality and pollution, and so forth.

Part of the infrastructure, specifically the sensor nodes and LoRaWAN access points (details will follow), are paid for and managed by individuals who voluntarily decide to equip their vehicles with sensors or manage an access point at home. These individuals are remunerated for their efforts by means of some virtual currency in the form of tokens that are credited by the service provider(s). Similarly, customers pay for accessing the data in the form of the same tokens; the details of how these tokens can be converted from/to "real" money is beyond the scope of this paper. All financial transactions happen in a blockchain.

We envision a scenario (Figure 1) where sensors are installed on vehicles, and gateways are placed at fixed locations. While static sensors may provide continuous location-specific information, mobile IoT devices enable data gathering on a larger area, possibly

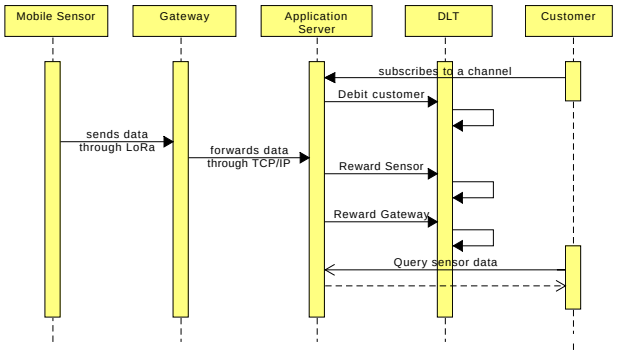


Figure 1: Interactions in the proposed architecture.

including remote places where stable network connectivity would be impractical or too expensive.

A typical LoRaWAN architecture consists of the following components:

- *Sensor Nodes*, which are the devices that collect the data of interest over the environment.
- *Gateways*, which are the bridges between IoT devices and the central network, as they relay the data received by the sensors to the Network Server. The link between the nodes and the gateway is based on LoRa.
- *Join Server*, which handles the device join procedure, security key management, and end-device authentication.
- *Application Server* focuses on application-specific data processing, user interfaces, and integration with external services. It may implement business logic, trigger actions based on received data, and interface with external databases or distributed ledgers.
- *Network Server*, which manages network-level functions like packet routing, quality of service, and authentication and authorization of the devices involved. Network servers also manage the deduplication of messages: if multiple gateways receive the same data, the policy is to consider the channel with the best signal strength [16], to maximize the success of downlink transmissions.

For what concerns the distributed ledger, both permissionless (open networks where anybody can participate) and permissioned (closed networks where only authorized users can join) blockchains can be employed in IoT applications. Permissioned blockchains provide significant scalability benefits by enabling the customization of the consensus protocol and block capacity, which can be set to get an optimal configuration. Additionally, they offer an extra layer of security since all involved parties may be considered trusted, and therefore are assumed to behave according to the protocols unless technical failures occur.

The integration of LoRaWAN components in the blockchain can vary based on the design choices of the application. Gateways can be either *full nodes* or *thin clients*, depending on whether they contribute to verifying the integrity of all the received data or merely store data fragments of interest. On the other hand, end-devices could either be *regular sensors* (i.e., low power devices that only broadcast data), *server-trusting clients* (i.e., they rely on APIs to interact with the blockchain, so no storage or computing capability is required), or *thin clients*.

For the sake of simplicity, we assume that all actors are trusted and have already followed the join procedure required by the LoRaWAN specifications. Mobile sensors send the environmental data to the gateways, which in turn forward the packets to the network server through TCP/IP. After the network server has filtered the messages, the information is sent to the application server (data brokers) that performs two actions: first, it stores the measurements into a database, and then credits the gateway, the LoRaWAN provider and the sensor with some amount of tokens representing some form of virtual currency.

The system allows multiple independent service providers (i.e., data brokers) to coexist. A permissioned blockchain might therefore be maintained cooperatively by the service providers, and is justified by the assumption that all the participating entities are identifiable and trustworthy. The information collected and organized by the data brokers can be accessed by customers upon payment of some amount of virtual currency. An appropriate balance between the inflow of virtual currency from customers and the outflow towards LoRaWAN administrators and sensor owners is required in order for the service providers to break even and generate profit for themselves.

All monetary transactions are managed by the blockchain, which automatizes the micropayments at a lower cost compared to traditional channels. In this context, the intended beneficiaries of the rewards are the blockchain nodes linked to the physical devices, potentially enabling a single actor within the distributed ledger to own multiple gateways or sensors. This allows sensor nodes, that are resource-constrained and have sporadic network connectivity, to be excluded from any interaction with the blockchain. Of course, all involved entities (customers, service providers, sensor owners) can check their balance at any time using personal devices (smartphones, PCs, ...).

To summarize, four categories of nodes are defined in the proposed architectures:

- *Full nodes*, which could be the nodes of the providers, with the task of gathering the transactions and inserting them into blocks that will be committed into the blockchain. Since we allow the presence of multiple providers, there will be at least one full node for each of them. This justifies the use of the blockchain, as it enables various providers to create a shared marketplace, sharing the costs for maintaining the infrastructure and offering customers a common platform to purchase sensor data. In Figure 1 the Application Server might be a full node of the distributed ledger.
- *Sensors*, which generate data and are rewarded by service providers through micropayments. As explained above, sensor owners can check the flow of micropayments by accessing the blockchain using their own devices, and withdraw virtual currency in order to use it. Thus, no blockchain functionality has to be embedded in the sensors, allowing them to save energy by avoiding complex computations.

- *Gateways*, which act as middlemen allowing sensors to upload data to the application servers. Gateways are placed at fixed locations where good network connectivity is assumed to be available. Similarly to sensors, no blockchain functionality is required on gateways, since their owners can access the blockchain by other means in order to check their balance.
- *Customers*, which are the final users of the data. They are subscribed to one or more flows of data upon payment of some fees in virtual currency. The fee might be calculated based on the amount of data accessed, the number of queries, or any other metric that is deemed appropriate.

4 Performance Modeling

The system described in Section 3 has been evaluated through simulation. This poses a significant challenge due to the complexity of the model, which includes a permissioned blockchain, mobile IoT entities, LoRaWAN technologies, and micropayments. These components add layers of complexity that require a sophisticated modeling framework. To address this issue, we have thus created a simulator [4] following the principles of *multilevel M&S* [23].

Multilevel modeling is a methodology that allows a complex model to be realized by assembling multiple independent submodels. This speeds up the development process and simplifies verification and validation by leveraging existing sub-models. Submodels can be of different types (continuous or discrete); additionally, the decomposition is not required to be static; indeed, submodels might be instantiated on-demand, and different realizations of the same sub-model may be active at different simulated times. For example, in a mobile scenario, a sparsely populated area might be described using a simple analytical model, switching to a more detailed but slower agent-based model if the population density increases and more accuracy is required.

To choose the simulators to integrate into the multilevel architecture, it is necessary to consider not only purely technical aspects (e.g., which simulator can more accurately reproduce certain features of interest) but also how well a model can be integrated within a complex environment. In fact, it is often necessary to make modifications to existing models, to adapt them to the specific scenarios of interest, and to enable interaction with other components. To carry out these operations, it is very useful to leverage open-source software, where the source code is available and can be inspected and modified.

The building blocks of our model are:

- A mobility simulator for modeling the movements of vehicles. We have chosen SUMO, an agent-based time-stepped tool for microscopic traffic simulation [13]. SUMO provides a platform for modeling complex traffic scenarios, including road networks, vehicles, pedestrians, traffic signals, and various control strategies. Furthermore, through a provided script it is possible to select the geographical area of interest and the intended traffic configuration.
- A LoRa simulator, capable of estimating transmit energy consumption, packet delivery ratio, and network throughput. We used simlorasf [27], a lightweight simulator written in Python that can be more easily integrated into our model than other network simulators like OMNeT++ or ns-3.
- A blockchain simulator to model the various dynamics of the P2P protocol. We used LUNES-blockchain [21], a time-stepped agent-based simulator capable of reproducing the behaviour of all actors involved in blockchain activities with high accuracy and efficiency. LUNES-blockchain can be easily customized to study different types of scenarios and blockchain policies.

SUMO can import geographic and road data from OpenStreetMap and configure vehicle flow within the geographic area of interest. Two parameters contribute to the setup: *Through Traffic Factors* controls the likelihood of selecting an edge positioned on the boundary compared to an edge completely contained within the simulation area. A higher value involves a greater influx and outflow of vehicles at the network borders, displaying increased movement of cars entering and exiting the simulation area. *Count* instead determines the number of vehicles per Km of lane that are generated every hour. The execution of SUMO can be triggered through the TraCI Python library[5] which facilitates the interoperability of SUMO with external software. Some minor changes to the source code were necessary to make the tools above suitable to our use case; this was possible thanks to their open-source nature.

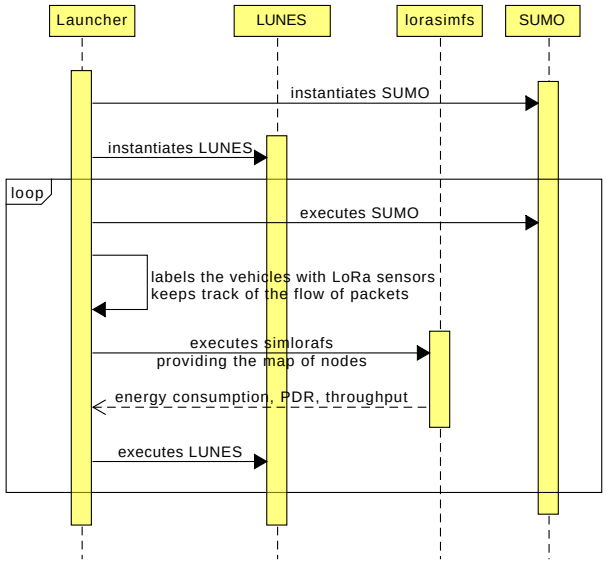


Figure 2: Sequence diagram of the multilevel simulator.

The integration of different sub-models requires the definition of orchestration and data exchange patterns, that specify how the different components interact and how they exchange information [22]. The interactions are shown in the sequence diagram in Figure 2. We use the *Model's Controller* orchestration pattern [22] where an ad-hoc module (the controller, which in our case is a launcher script) sits at the top of the hierarchy of the multilevel

[4] https://github.com/luca-Serena/lora-blockchain/

[5] https://pypi.org/project/traci/

model, serving both as the user interface and as the controller for the execution of the sub-models. SUMO and LUNES-blockchain are started at the beginning and remain active during the whole simulation, the reason being that they must maintain state information (i.e., position, velocity and direction of vehicles, the content of the blockchain) across the entire execution. The launcher then executes a defined number of steps: at each step, SUMO is advanced to the next timestep to update the position of vehicles. Then, simlorasf is invoked to execute data transfers for vehicles that moved within the communication range of some gateway. simlorasf also computes the energy consumption of the mobile sensors and stores the information into a log file, so that it can be analyzed at the end of the simulation run. After all communications are completed, the launcher script calls LUNES-blockchain to assign the rewards to nodes and gateways and store them in the distributed ledger.

5 Simulation Results

We considered the city of Bologna (Italy) and placed gateways in the locations shown in Figure 3. Gateways 1 and 2 are actual LoRaWAN Gateways that are installed in the city, Gateways 3, 4, and 5 are placed in strategic positions (Piazza Maggiore, Dall'Ara Stadium, and Bologna-Mazzini train station), and Gateway 6 is placed in a peripheral location. The arrangement of the sensors has been chosen to cover a significant portion of the map, particularly focusing on high-traffic areas. Most of the vehicles in fact are active in the center, while the green-shaded parts of the map represent hilly zones with the surrounding streets having lower vehicle density.

Following an initial warm-up phase required to populate the area with a suitable number of vehicles, the simulation proceeded for 500 steps, where each SUMO step represents a second of wall-clock time. We assume that every 10 seconds the sensors emit messages.LoRa-equipped vehicles lack awareness of gateway presence within their transmission range, making it very difficult to ensure that all transmitted data has been received. While occasional data losses might be acceptable in some applications, they might not be acceptable in others. There are various solutions to mitigate this issue. One approach involves storing and transmitting the last k data points, where k is the transmission window. Higher values of k increase the likelihood that all data points will eventually reach the application server at least once, at the cost of a larger message size and greater effort for identifying and removing duplicates on the receiving side. Other approaches based on explicit acknowledgments using the downlink channel might be designed, but they require sensors to wait for message receipts and the handling of retransmissions. In terms of SUMO traffic generation parameters, our testbed has a *Through Traffic Factors* value (i.e. likelihood for the vehicles of choosing a boundary edge) of 5 and a *Count* value (i.e., quantity of vehicles produced every hour per kilometers of lane) of 100; we assume that 10% of all vehicles are equipped with sensors.

Regarding financial compensation, we assume that sensors, gateways and providers are rewarded with a certain amount of tokens ("virtual coins") for every transaction validated in the blockchain.

In our experiments, the number of tokens awarded to gateways is proportional to the volume of messages they forward. If multiple gateways receive the same message, standard LoRaWAN deduplication policies are applied, and the token is assigned to the gateway

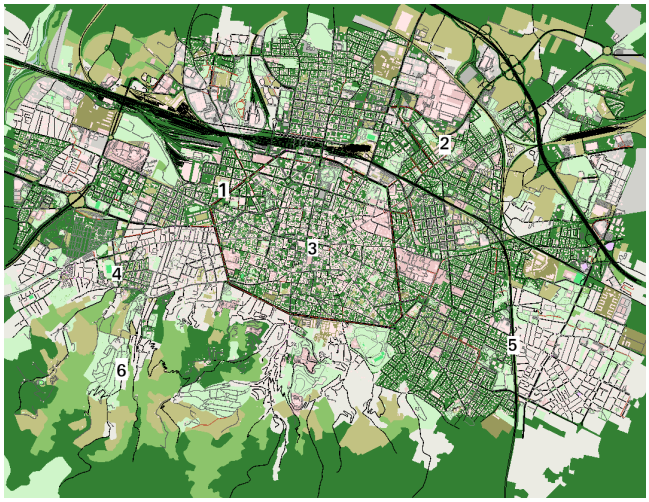


Figure 3: Placement of LoRa Gateways in the simulation scenario.

with the strongest signal—assumed in our simulation to be the closest one to the sensor. In Figure 4 we show the reward obtained by the gateways after 10 minutes of simulated time. We compared two policies: in the flat policy every gateway is rewarded 1 token for every processed message. This policy favors gateways that are positioned in crowded areas, since they will likely forward the higher number of messages, and penalizes gateways that operate in remote locations. Since one of the main points of the proposed architecture is to provide decentralized services for the countryside, we consider a weighted policy, gateways initially earn 1.1 tokens per message. This earning decreases by 0.2 tokens for every 1000 messages, until it reaches a minimum threshold of 0.5 tokens per message. Every 10 minutes the earnings are reset to the initial value.

The gateways receiving a higher reward are either those positioned in the busiest areas (Gateways 1 and 3) or those individually covering a wide territory (Gateway 5). On the other hand, Gateway 6, which is located in a peripheral and low-traffic area of the city, is by far the one that retransmits the fewest messages.

We then evaluate the system's resilience in the event of gateway malfunctions. When all gateways are operational, the packet delivery ratio is 94.2% with 1% of the vehicles equipped with sensors. Figure 5 shows how this ratio declines if a gateway fails. Notably, Gateway 5 plays a critical role in maintaining good coverage, as it is the only gateway serving the southeast region, where traffic levels are quite high.

The fraction of cars equipped with sensors has a big influence on metrics such as transmit energy consumption and the packet delivery ratio, as shown in Figure 6. When this parameter increases, the energy consumption grows proportionally, while the packet delivery ratio decreases due to a higher amount of interference.

When managing a potentially large volume of transactions, utilizing blockchain solutions capable of handling such data flow becomes crucial. Private blockchains offer the advantage of customizing the distributed ledger's configuration in order to effectively accommodate the data rate required by the IoT application. The

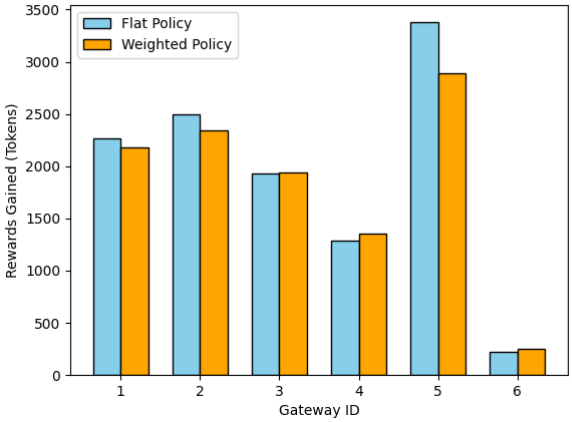


Figure 4: Rewards obtained by each gateway.

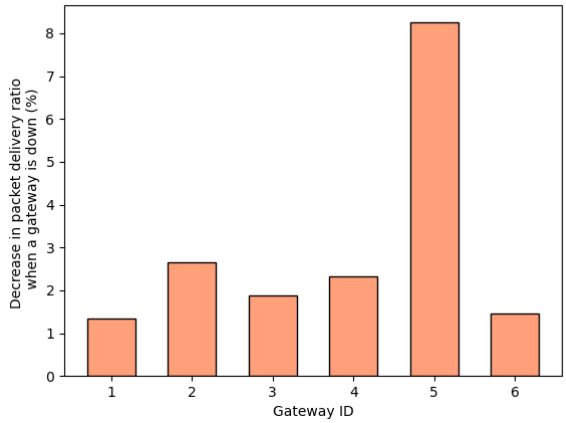


Figure 5: Percentage difference in received messages between a scenario with all gateways operational and a scenario with one inactive gateway.

capacity to support transactions within a specific timeframe is determined by two key attributes: the maximum number of transactions allowed per block and the frequency of block production. These attributes can be unified into a single metric, defined as the maximum transaction throughput. Figure 7 shows the percentage of transactions successfully inserted into a block, depending on the number of vehicles equipped with LoRa sensors and the throughput, which is expressed as the maximum number of blockchain that can be inserted in one minute. Obviously, in a real-world system it is desirable that no transaction gets lost. Therefore, it is necessary to customize the blockchain attributes in a way that 100% insertion rate is ensured. In our experiments, consensus is achieved through PoA where provider nodes, serving as trusted entities, are responsible for block production. It is worth saying that the proposed architecture is not heavily dependent on a specific

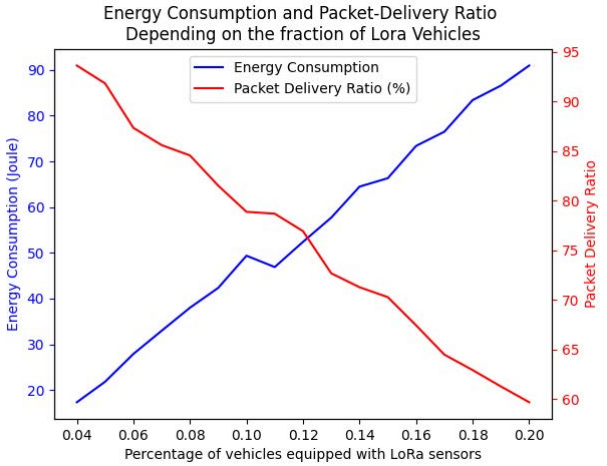


Figure 6: Increasing the number of sensors increases the transmit energy consumption and reduces the packet delivery ratio.

consensus mechanism. While PoA is a well-established choice for permissioned blockchain [25], there is no specific standard about how the PoA is actually implemented. Thus, we decided to rely on a PoS-based implementation, since it is easily supported by the simulator. However, any suitable consensus mechanism could be integrated.

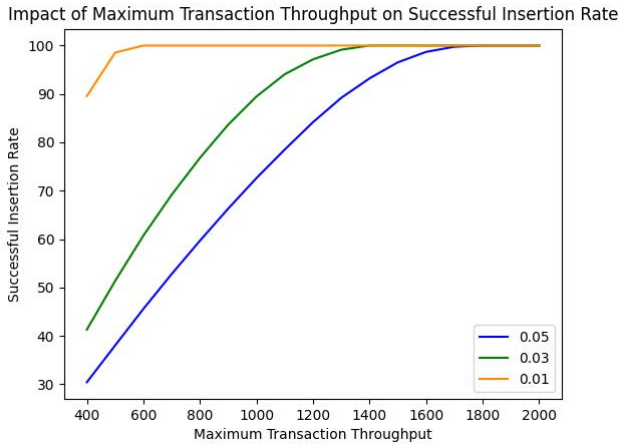


Figure 7: Transactions successfully validated in the blockchain. The three curves represent the percentage of vehicles equipped with LoRa sensors, respectively 1%, 3%, and 5% of all vehicles.

6 Conclusions

In this paper, we explore a solution that integrates the gathering and trading of sensor data. While LoRaWAN is an established standard for IoT applications, enabling efficient retrieval of data, blockchain

technology could facilitate the establishment of a marketplace. This could empower customers to subscribe to various data streams, possibly generated and managed by different entities. To simulate such a system, we have adopted a multilevel modeling approach, allowing us to combine existing models within a single framework. In fact, one of the purposes of this paper is to demonstrate the suitability of multilevel modeling techniques to simulate complex IoT scenarios involving multiple components. Other than mitigating the cost of building a complex model from scratch, we were able to reuse task-specific simulators that have been assembled following the design principles of multilevel modeling. The experiments suggested that even with a small number of strategically positioned gateways, extensive coverage of large urban areas is feasible. Private blockchains allow stakeholders to customize the attributes of the distributed ledger to meet system requirements, effectively addressing issues such as trust and scalability.

In future works, we plan to investigate similar use cases, such as smart shire scenarios where data transmission is required in locations without Internet coverage. Additionally, we will consider a more sophisticated architecture that offers features that can potentially increase the usability of certain applications, such as the possibility to negotiate service costs and better strategies to reduce message losses.

Acknowledgments

This work is partially supported by the European Union - NextGenerationEU within the framework of PNRR Mission 4 - Component 2 - Investment 1.1 under the Italian Ministry of University and Research (MUR) programme "PRIN 2022" - grant number 2022N2NH42 SmartShires – CUP: H53D23003570006. M.M. is partially supported by INdAM-GNCS and by the ICSC National Research Center for High Performance Computing, Big Data and Quantum Computing within the NextGenerationEU program - CUP: J33C22001170001. P.M. is partially supported by the European Union's Horizon Europe Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement No. 101086387. This work is also partially supported by the research project CIPROM/2023/29, funded by "Direcció General de Ciència i Investigació" Generalitat Valenciana - SPAIN.

References

[1] Zainab H Ali, Hesham A Ali, and Mahmoud M Badawy. 2015. Internet of Things (IoT): definitions, challenges and recent research directions. *International Journal of Computer Applications* 128, 1 (2015), 37–47.

[2] Yusuke Aoki, Kai Otsuki, Takeshi Kaneko, Ryohei Banno, and Kazuyuki Shudo. 2019. Simblock: A blockchain network simulator. In *INFOCOM WKSHPS*. IEEE, Paris, France, 325–329. https://doi.org/10.1109/INFCOMW.2019.8845253

[3] Bouthaina Dammak, Mariem Turki, Saoussen Cheikhrouhou, Mouna Baklouti, Rawya Mars, and Afef Dhahbi. 2022. Lorachaincare: An iot architecture integrating blockchain and lora network for personal health care data monitoring. *Sensors* 22, 4 (2022), 1497.

[4] Seyed Mehdi Fattahi, Adetokunbo Makanju, and Amin Milani Fard. 2020. SIMBA: An efficient simulator for blockchain applications. In *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*. IEEE, Valencia, Spain, 51–52.

[5] Lorenzo Felli and Romeo Giuliano. 2021. Access Control in woodland through Blockchain and LoRaWAN. In *2021 AEIT International Conference on Electrical and Electronic Technologies for Automotive*. IEEE, Torino, Italy, 1–5.

[6] Marko Grebovic, Tomo Popovic, and Radmila Sindjic Grebovic. 2023. Blockchain Technology for Weather Data Management. In *2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE, East Sarajevo, Bosnia and Herzegovina, 1–6. https://doi.org/10.1109/INFOTEH57020.2023.10094082

[7] Najmul Hassan, Saira Gillani, Ejaz Ahmed, Ibrar Yaqoob, and Muhammad Imran. 2018. The role of edge computing in internet of things. *IEEE communications magazine* 56, 11 (2018), 110–115.

[8] Jiejun Hu, Kun Yang, Kezhi Wang, and Kai Zhang. 2020. A blockchain-based reward mechanism for mobile crowdsensing. *IEEE Transactions on Computational Social Systems* 7, 1 (2020), 178–191.

[9] Tongyi Huang, Wu Yang, Jun Wu, Jin Ma, Xiaofei Zhang, and Daoyin Zhang. 2019. A Survey on Green 6G Network: Architecture and Technologies. *IEEE Access* 7 (2019), 175758–175768. https://doi.org/10.1109/ACCESS.2019.2957648

[10] Laurie Hughes, Yogesh K. Dwivedi, Santosh K. Misra, Nripendra P. Rana, Vishnupriya Raghavan, and Viswanadh Akella. 2019. Blockchain research, practice and policy: Applications, benefits, limitations, emerging research themes and research agenda. *International Journal of Information Management* 49 (2019), 114–129. https://doi.org/10.1016/j.ijinfomgt.2019.02.005

[11] Mukhammad Gufron Ikhsan, Muhammad Yanuar Ary Saputro, Dian Abadi Arji, Ruki Harwahyu, and Riri Fitri Sari. 2018. Mobile LoRa gateway for smart livestock monitoring system. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*. IEEE, Bali, Indonesia, 46–51. https://doi.org/10.1109/IOTAIS.2018.8600842

[12] Shahzeb Javed and Dimitrios Zorbas. 2022. LoRaWAN Downlink Policies for Improved Fairness. In *IEEE Conference on Standards for Communications and Networking (CSCN '22)*. IEEE, Thessaloniki, Greece, 1–6.

[13] Daniel Krajzewicz. 2010. Traffic Simulation with SUMO – Simulation of Urban Mobility. In *Fundamentals of Traffic Simulation*, Jaume Barceló (Ed.). Springer New York, New York, NY, 269–293. https://doi.org/10.1007/978-1-4419-6142-6_7

[14] Bahareh Lashkari and Petr Musilek. 2021. A comprehensive review of blockchain consensus mechanisms. *IEEE Access* 9 (2021), 43620–43652.

[15] Jun Lin, Zhiqi Shen, Chunyan Miao, and Siyuan Liu. 2017. Using blockchain to build trusted LoRaWAN sharing server. *International Journal of Crowd Science* 1, 3 (2017), 270–280.

[16] Pierluigi Locatelli, Pietro Spadaccino, and Francesca Cuomo. 2022. Ruling out IoT devices in LoRaWAN. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops*. IEEE, New York, USA, 1–2. https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798063

[17] Pratyusa Mukherjee and Chittaranjan Pradhan. 2021. Blockchain 1.0 to Blockchain 4.0—The Evolutionary Transformation of Blockchain Technology. In *Blockchain Technology: Applications and Challenges*, Sandeep Kumar Panda, Ajay Kumar Jena, Santosh Kumar Swain, and Suresh Chandra Satapathy (Eds.). Springer International Publishing, Cham, 29–49. https://doi.org/10.1007/978-3-030-69395-4_3

[18] Stefano Musso, Guido Perboli, Mariangela Rosano, and Alessandro Manfredi. 2019. A decentralized marketplace for M2M economy for smart cities. In *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, Napoli, Italy, 27–30. https://doi.org/10.1109/WETICE.2019.00014

[19] Kazim Rifat Ozyilmaz and Arda Yurdakul. 2019. Designing a Blockchain-based IoT with Ethereum, swarm, and LoRa: The software solution to create high availability with minimal security risks. *IEEE Consumer Electronics Magazine* 8, 2 (2019), 28–34.

[20] Luca Sciullo, Angelo Trotta, and Marco Di Felice. 2020. Design and performance evaluation of a LoRa-based mobile emergency management system (LOCATE). *Ad Hoc Networks* 96 (2020), 101993.

[21] Luca Serena, Gabriele D'Angelo, and Stefano Ferretti. 2022. Security analysis of distributed ledgers and blockchains through agent-based simulation. *Simulation Modelling Practice and Theory* 114 (2022), 102413.

[22] Luca Serena, Moreno Marzolla, Gabriele D'Angelo, and Stefano Ferretti. 2023. Design Patterns for Multilevel Modeling and Simulation. In *2023 IEEE/ACM 27th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, Singapore, 48–55. https://doi.org/10.1109/DS-RT58998.2023.00015

[23] Luca Serena, Moreno Marzolla, Gabriele D'Angelo, and Stefano Ferretti. 2023. A review of multilevel modeling and simulation for human mobility and behavior. *Simulation Modelling Practice and Theory* 127 (2023), 102780. https://doi.org/10.1016/j.simpat.2023.102780

[24] Lyubomir Stoykov, Kaiwen Zhang, and Hans-Arno Jacobsen. 2017. VIBES: fast blockchain simulations for large-scale peer-to-peer networks: demo. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos* (Las Vegas, Nevada) *(Middleware '17)*. Association for Computing Machinery, New York, NY, USA, 19–20. https://doi.org/10.1145/3155016.3155020

[25] Nenad Zoran Tomić. 2021. A review of consensus protocols in permissioned blockchains. *Journal of Computer Science Research* 3, 2 (2021), 19–26.

[26] Thiemo Voigt, Martin Bor, Utz Roedig, and Juan Alonso. 2016. Mitigating inter-network interference in LoRa networks. arXiv preprint arXiv:1611.00688.

[27] Tugrul Yatagan and Sema Oktug. 2019. Smart spreading factor assignment for lorawans. In *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, Barcelona, Spain, 1–7.