Elsevier required licence: © <2021>. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/
The definitive publisher version is available online at https://doi.org/10.1016/j.ipm.2021.102492

ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/348419398

# A Novel Dual-Blockchained Structure for Contract-Theoretic LoRa-based Information Systems

**Article** *in* Information Processing & Management · January 2021

DOI: 10.1016/j.ipm.2021.102492

CITATIONS
0

READS
45

**7 authors**, including:

Guangsheng Yu
University of Technology Sydney

**13** PUBLICATIONS **42** CITATIONS

SEE PROFILE

Litianyi Zhang
The University of Sydney

**1** PUBLICATION **0** CITATIONS

SEE PROFILE

Xu Wang
University of Technology Sydney

**21** PUBLICATIONS **205** CITATIONS

SEE PROFILE

Kan Yu
La Trobe University

**25** PUBLICATIONS **233** CITATIONS

SEE PROFILE

**Some of the authors of this publication are also working on these related projects:**

Project An intelligent system uncovering malicious network packet payloads View project

Project Environmental Localization and Mapping View project

All content following this page was uploaded by Guangsheng Yu on 13 January 2021.

The user has requested enhancement of the downloaded file.

# A Novel Dual-Blockchained Structure for Contract-Theoretic LoRa-based Information Systems

Guangsheng Yu[a,b,*], Litianyi Zhang[c], Xu Wang[a,b], Kan Yu[e], Wei Ni[d], J. Andrew Zhang[a], Ren Ping Liu[a,b]

[a] *Global Big Data Technologies Centre, University of Technology Sydney, Australia*
[b] *Food Agility CRC Ltd, 81 Broadway, Ultimo, NSW, Australia, 2007*
[c] *Centre of Excellence in Telecommunications, the University of Sydney, Australia*
[d] *Data61, CSIRO, Australia*
[e] *The Department of Computer Science and Information Technology, La Trobe University, Bendigo, Victoria, Australia*

## Abstract

LoRa serves as one of the most deployed technologies in Internet-of-Things-based information systems (IoT-IS), and self-motivated deployment is the key to the rollout of LoRa. Proper incentive can play an important role in encouraging the private deployment of LoRa, increasing coverage and promoting effective management of IoT-IS. However, existing incentive mechanisms have the vulnerabilities of insecure centralized architecture and excessive utility loss of LoRa Controllers and Gateways, due to asymmetric information between private owners of gateways and centralized controller (or service providers). Blockchain-based LoRa networks, as a promising solution, have not been comprehensively studied to address the vulnerabilities, let alone the other issues of security, scalability, and flexibility. In this paper, we propose a novel Dual-Chained LoRa-based information system (LoRa-IS) to provide globally cross-validated security. Behaviors, including state-of-the-art contract-theoretic incentive mechanism and new flow control protocol, can be secured with the tamper-resistance of Blockchains. Being part of the proposed incentive mechanism, the new self-

[*] Corresponding author

*Email addresses:* Guangsheng.Yu@uts.edu.au (Guangsheng Yu), litianyi.zhang@sydney.edu.au (Litianyi Zhang), Xu.Wang-1@uts.edu.au (Xu Wang), k.yu@latrobe.edu.au (Kan Yu), Wei.Ni@data61.csiro.au (Wei Ni), Andrew.Zhang@uts.edu.au (J. Andrew Zhang), Renping.Liu@uts.edu.au (Ren Ping Liu)

driven flow control allows both the Dual-Chain system and the LoRa network to scale. To the best of our knowledge, the proposed system is the first comprehensive Blockchain-based LoRa-IS combined with contract theory. We also provide analysis and simulations, showing that our system can pay fair incentives under information asymmetry. With the new flow control, the system can optimize network coverage while improving the Blockchain scalability and flexibility.

*Keywords:* Information system, Internet-of-Things (IoT), Incentive mechanism, Contract theory, LoRa, Blockchain

## 1. Introduction

The Internet of Things (IoT) are widely applied in information systems (IS) with various requirements [1], including smart home management systems [2], security policy in sensing network systems [3], and automation and connectivity of smart vehicle systems [4]. The pervasiveness of IoT-IS leads to a large scale of data transmission and a significantly improved capability of network communication protocols. LoRa (Long Range), one of the most popular low-power wide-area network (LPWAN) technologies for IoT-IS, has been reported to outperform cellular-based LPWAN (e.g., Narrow-Band IoT) in corporate or private realms, due to its self-driven public participation and comprehensive open-source community [5, 6]. Operating in an unlicensed band, several open-source protocols are presented (e.g., LoRaWAN [7]), where a star-of-stars network is established between end-devices and a LoRa Gateway, and between LoRa Gateways and a LoRa Controller (i.e., LoRa network server). It is reported that LoRa has held the highest market share in some countries and accounted for the highest annual unit shipments along with an increasing projection of the entire IoT-IS market [8, 9, 10].

The use of unlicensed bands and open-source platforms compel LoRa to stay on top of improving the scalability and flexibility among self-deployed gateways (e.g., relieving traffic congestion resulting from the channel limit) [11, 12], while incentivizing more private owners to increase coverage. However, LoRa is in lack of secure, effective, and fair incentive mechanisms to incentivize the private deployment of LoRa [13]. This makes it difficult to densify the deployment and increase coverage and spectrum utilization [14]. Existing incentive mechanisms, such as linear pricing or Stackelberg game pricing [15], are ineffective in the absence of a supervising entity. Such an entity is essential, and regulates all behaviors and conditions between LoRa Gateways and LoRa Controllers in LoRa networks. This is because a malicious gateway may try to mislead its controller by overstating its performance, usually known as information asymmetry [15].

This paper proposes a new incentive mechanism by using contract theory [16] in coupling with a new flow control protocol. The new contract-theoretic incentive mechanism maximizes the utility of both the controllers and gateways at the same time, even under information asymmetry. Different from the existing approaches listed in [13], our new incentive mechanism improves scalability and flexibility by motivating the self-deployed gateway.

Existing contract-theoretic designs neither consider a malicious controller which censors the gateways by paying less reward, nor support a reliable mechanism to prove the validity of the incentive processing. Blockchain is suitable to avoid this issues of centralization by taking advantage of the decentralized architecture and tamper-resistant validation [17, 18]. However, traditional Blockchain technologies fail to handle massive data streams, incurring significant latency and low throughput [19, 20]. In addition, the one-device-to-many-gateway property (which is helpful for redundancy) of the most popular protocol, LoRaWAN, may compromise the scalability of Blockchain and the scalability of LoRa networks.

The above issues remain and a holistic solution is in demand. Specifically, the remaining challenges are: 1) how the contract-theoretic incentive mechanism can be integrated into LoRa to maximize each entity's profit in the presence of information asymmetry, in which a new self-driven flow control protocol can relieve the traffic congestion and throughput loss resulting from duplicated data uploaded to the Blockchain; and 2) how the Blockchain can be integrated into the proposed contract-theoretic LoRa-based information system (LoRa-IS) to provide secure and scalable data storage services.

In this paper, we develop a new Dual-Chain-based LoRa-IS by taking advantage of the Directed Acyclic Graph (DAG) structure. Such DAG is set along with an identity chain that provides identity registration and protocol monitoring in smart contracts. By interacting with both the DAG and identity chain and leveraging decentralized global cross-validation, a new contract-theoretic incentive mechanism can be secured. The contract-theoretic incentive mechanism significantly relieves the negative effect of information asymmetry, and improves the utility of both LoRa Controllers and Gateways than existing incentive mechanisms. Being part of the proposed incentive mechanism, the new self-driven flow control protocol contributes to the scalability of the LoRa network by relieving traffic congestion, and improves the throughput of the Blockchain. As a result, the system enables: 1) strong compatibility with typical LoRaWAN protocols; 2) the Proof-of-Task-Overhead (PoTO), a new spam protection dedicated for LoRa networks to reduce resource waste; 3) efficient and flexible data storage services at any time with high throughput; and 4) the transparency and fairness of incentives and data storage services because of the tamper-resistance property of the decentralized cross-validation protocol.

As revealed by our analysis and simulation results, the proposed contract-theoretic incentive mechanism outperforms existing incentive mechanisms in terms of the utility of the LoRa Controllers under information asymmetry. It is also revealed that the proposed Dual-Chain-based LoRa-IS can significantly improve the throughput of the Blockchain, while maintaining a high area utilization.

The rest of this paper is organized as follows. In Section 2, we present the proposed system model for the new Dual-Chain-based structure and the new flow control protocol. The implementation of contract theory in LoRa-IS is presented in Section 3, followed by numerical simulations in Section 4. Section 5 reviews the related works regarding Blockchain-based ISs, Blockchain-based LoRa technologies, and the implementation of contract theory in LoRa technologies. Section 6 concludes the paper.

## 2. System Model

As shown in Fig. 1, a number of LoRa Controllers are distributed, each of which runs a regional LoRa network supported by several LoRa Gateways. Many end-devices are distributed in each of the regions. Each region is covered by a LoRa Gateway which forwards data between the end-devices and the corresponding LoRa Controller. There could be overlapping coverage areas among the gateways. Data sent from the end-devices situated in the overlapped area (the yellow, blue, and red regions in Fig. 1) is likely to be received by multiple gateways that have overlapping coverage.

To prevent these gateways from being congested or transmitting excessive duplicated data to their corresponding controller, a self-driven flow control protocol is applied to the gateways that have a large overlapping coverage area, i.e., the yellow and blue regions. Here, the term "self-driven" indicates that the flow control is not aimed to control traffic by cutting off the transmission. Instead, it is to incentivize the gateways to maximize their profit by complying with any preconcerted rules. A single epoch comprises several time-frames assigned to each gateway. Each authorized gateway is assigned to transmit within its own short time-frame based on the flow control protocol running on the controller.

A controller publishes a contract-theoretic task with a random timeout (i.e., the controller decides the timeout point which is identical to a single epoch of the flow control) to all of its corresponding gateways. Based on the uploading data size from the gateways to their controller, the gateways are classified into different types in a tree structure managed by the controller, as shown in the type-classification in the red box at the bottom-right corner of Fig. 1. The discounted uploading data size as the penalty is applied to the unauthorized gateways transmitting beyond its time-frame. The larger data size a gateway contributes within the epoch, the higher likelihood the gateway is classified into a type that is closer to the root of the tree, and the more incentive can be granted for this task.

Each controller is powerful and robust, and responsible for multiple gateways

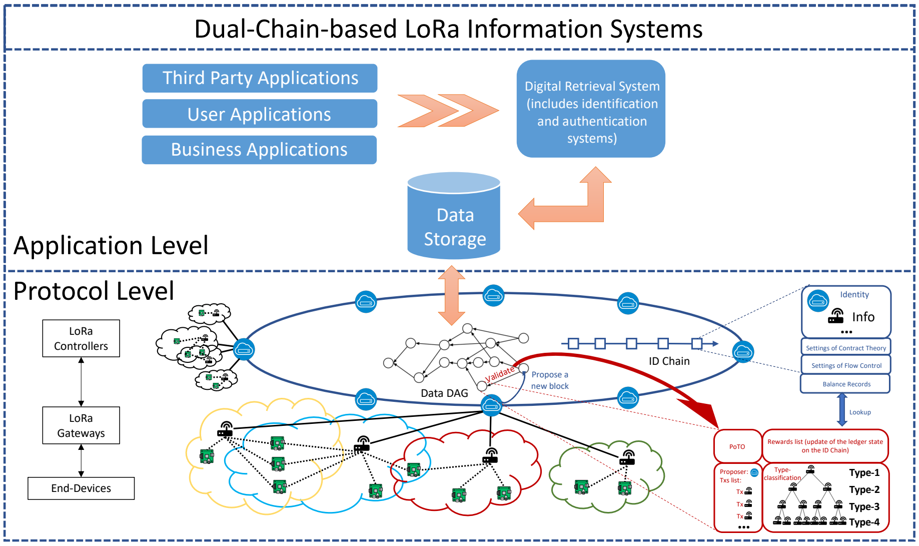


Figure 1: The system architecture of the proposed Dual-Chain-based LoRa information system with contract-theoretic incentive mechanism. It is composed of application level and protocol level. The application level maintains the LoRa-IS control process. Data from multiple types of applications is accessed by digital retrieval systems to guarantee the security and privacy, then uploaded to the data storage platform. In the protocol level, a star-of-stars network is established between end-devices and a LoRa Gateway, and LoRa Gateways and a LoRa Controller. A group of controllers constitute a committee and maintain a Dual-Chain structure, i.e., the *ID Chain* providing the registration, monitoring service, and balance record, and *Data DAG* providing the data storage service. Each controller as a task publisher collects data from its corresponding gateways and pays incentives on the *ID Chain*. Such behaviors are included in a DAG block to be injected to the *Data DAG*, and cross-validated by other controllers. LoRa data stored on the *Data DAG* serves the applications in the LoRa-IS.

in a geographical region, and the controllers cooperate and interact with each other[1]. Specifically, all controllers maintain a Dual-Chain-based structure, i.e., an *ID Chain* and a *Data DAG*, based on which the data can be secured by conducting a tamper-resistant cross-validation among the controllers. Critical information is uploaded to the *ID Chain* as smart contracts by controllers during the registration phase, and updated periodically for the cross-validation, as shown in the blue box at the top-right corner of Fig. 1. Such information includes the settings and status of published contract-theoretic tasks, the flow control processing, and each gateway's serial number and public key (from which the Blockchain address can be derived).

The *ID Chain* is a regular Blockchain only maintained by the controllers, while a DAG-based structure is used for data storage, i.e., the *Data DAG* managed by both the controllers and gateways. In a DAG-based structure inspired by [24], blocks can be proposed in parallel. Each proposed block needs to validate other pending blocks as a part of contribution to be proved. Such pending blocks are usually those previously proposed by other controllers and yet not validated by sufficient subsequent blocks. To introduce a DAG-based structure instead of a classical chain-based one for data storage is due to its high scalability by supporting parallel blocks/transactions. With its IOTA-like DAG-based solution, the *Data DAG* is particularly suitable for storing large amounts of data in parallel. A LoRa Controller does not need to wait until its previous

[1] The LoRa Controllers are typically owned and maintained by the service providers [21]. We consider a decentralized league of hundred-scale controllers in different controller platforms or different geographical regions, while each controller can be much powerful than a single end-device or gateway machine, such as enterprise data centres (e.g., The Things Network, TTN) and smaller-sized organizations using open-source platform [22, 23]/Software-as-a-service (SAAS)/Platform-as-a-service (PAAS) implemented at the cloud-based distributed cluster. The LoRa data originating from densely distributed gateways in the same region ends up at a single controller platform that manages this region. The physically damaged individual controller would be destructive to the regional LoRa service, but would not affect the data retrieval service thanks to the faulty tolerance of the dual-chain structure.

block has been accepted or others have finalized their blocks, it can publish tasks anytime with any timeout based on its requirements. However, a single *Data DAG* is not as powerful as a chain-based solution in terms of the support for smart contracts [25, 26]. Thus, we propose a separate *ID Chain* which is superior in terms of handling this issue. The *ID Chain*, in spite of its relatively weak support for parallel blocks compared to a *Data DAG*, can be as scalable as a *Data DAG* in terms of transactions per second by implementing advanced technologies, such as the HotStuff consensus algorithm [27] that enables a high transaction rate among large communities, and the sharding technology [28] that enables horizontal scalability.

By combining these two primitives, we are able to deliver a controllable and flexible dual-chain system. In particular, different functionalities are split between a *Data DAG* and *ID Chain*. The *Data DAG* enables the parallel data storage service, where LoRa Controllers can initiate contract-theoretic tasks and collect the LoRa data with no need of compliance with the sequence of the consensus process. The *Data DAG* also provides a flexible validation process for businesses that requires lower security levels and latency. The *ID Chain* is adopted to conduct devices registration/monitoring and contract in initialization/status records. It is also responsible for the incentive payment and balance records of the whole system. Therefore, splitting the data storage and balance records between the *Data DAG* and *ID Chain* enables the independence among controllers. Each controller is only responsible for the validity of on-chain LoRa data stored on the *Data DAG*, and not for the validity of the metadata, i.e., the physical meaning of the data stored on the *Data DAG*. This allows for improved controllability and flexibility.

By looking up the information of each gateway in the *ID Chain* and validating each transaction size, hash value, and timestamp in the tree, the other controllers can confirm a contribution of a particular gateway is valid. With the same method, the type-classification conducted by a controller and the transfer of the incentive can also be validated on the *ID Chain* by the other controllers, as shown in the red box at the bottom-right corner of Fig. 1. Thus, the cross-validation can be conducted among the LoRa Controllers to prevent malicious behaviors in the *Data DAG*, such as an invalid process of the PoTO protocol, invalid published contract-theoretic task, invalid rewards transfer, and invalid source data from some unknown controllers or gateways.

The proposed system focuses on the uplink transmission, because 1) the uplink transmission is strongly favored in LoRa networks [29]; 2) the downlink transmission can only happen after a successful uplink transmission [30]; and 3) the duty cycle of uplink transmission is regulated by several government agencies and departments due to the default ALOHA access and limited channel resource, thus limiting the usage of LoRa networks [31].

### *2.1. Dual-Chain Structure and Cross-Validation*

The proposed Dual-Chain structure consists of two types of Blockchains, *ID Chain* for gateways registration, contract initialization, and status updating, and *Data DAG* for efficient data storage service, as shown in Fig. 1. The Dual-Chain structure is the foundation of the cross-validation conducted among all participating LoRa Controllers, for instance, validating the behaviors of controllers.

**Registration and Initialization on ID Chain -** The *ID Chain* with a typical chain-based structure is maintained by all participating controllers to conduct the registration of the gateways, contract initialization services, and subsequent status updating. Specifically, the *ID Chain* consists of two operations (see Lines 1 and 2 of Algo. 1):

- *Devices Registration:* Each gateway needs to register its identity to its corresponding controller, including its unique serial number and public key (from which the Blockchain address can be derived). Thus, the controller collects a list of gateways it is responsible for, and can subsequently upload the list to a smart contract on the *ID Chain*. Note that we consider a controller that owns the sufficient performance to maintain multiple Blockchains and manipulate a large amount of data among the gateways and the other controllers.

- *Contract Initialization:* Besides the gateways registration, the smart contract also records the rule of flow control and type-classification for each registered controller based on their own requirements; see Section 3. The rule includes but not limited to 1) the number of types; 2) the number of gateways in each type; and 3) the amount of incentive paid to each type.

By interacting with such an *ID Chain*, each controller fetches and updates the information of the gateways and types secured by the tamper-resistance property, and thus a reliable reference can be provided to the controllers during the cross-validation phase.

**Data Uploading to Data DAG -** A typical chain-based structure incurs poor flexibility for LoRa networks due to its sequential generation of blocks one after another. Different from the typical chain-based *ID Chain*, a *Data DAG* features a DAG-based structure enabling parallel blocks/transactions to replace the traditional single block. In this paper, the *Data DAG* features a general DAG-based structure where a block can be injected into the network at any time with an upper-bounded frequency and PoTO certified. Such a block is designed to verify a certain number of pending blocks (e.g., two blocks in IOTA [24]), and will be accepted by the network with a high likelihood if the block has been verified by a sufficient number of forthcoming blocks. Note that the *Data DAG* is compatible with the common tools used in typical DAG-based structures (e.g., the trunk/branch transaction process for bundling in IOTA). It is generic and not limited to any specific tools, protocols, or algorithms. A weighting factor is applied to encourage blocks to verify the most recent pending blocks (i.e., the tips of the DAG), and verify pending blocks proposed by the other controllers (i.e., the cross-validation). Consequently, the system can achieve:

- high throughput of the *Data DAG* (transactions/second) as processing parallel blocks/transactions are now allowed simultaneously;
- instead of a single block generator per slot, a controller does not have to wait until its own previous blocks has been accepted or the other controllers have finalized their own blocks.

Based on the above, high throughput can be achieved in the network in which the controllers can publish a contract-theoretic incentive task for their gateways, and upload the result at any time with an upper-bounded frequency[2], as well as a timeout parameter on demand. This significantly contributes to an effective mechanism for LoRa networks in which high scalability and flexibility are important.

Data uploading is conducted for the controllers to upload the transactions sent from the corresponding gateways to the *Data DAG*. It consists of the following steps; see Lines 3-11 of Algo. 1).

*Data collection at LoRa Controllers:* Once a controller makes a decision, it publishes a task to inquire transactions consisting of LoRa data from its corresponding gateways. The task is subject to a pre-defined timeout and the allocation of the flow control if enabled. The controller marks full contribution only for transactions received from assigned gateways during a time-frame based on the allocation of the flow control (i.e., a discount is applied as a penalty for those breaching the rule; see Section 2.2). Note that collecting data from end-devices on the gateways' sides is independent of the flow control. A gateway can suspend the data forwarding while continuing to buffer the received LoRa data, if the gateway is unauthorized. The transaction sent from a gateway is secured by using its private key associated with the public key stored on the *ID Chain*, and also specifically contains the data size and data hash. Any received transactions which 1) belong to unregistered gateways, 2) present unmatched data size or data hash, or 3) display incorrect data format (e.g., not comply with the pre-defined upper-bounded size of uploading data; see Section 3), are considered to be invalid and discarded by the controller.

[2] Strictly, there should be a lock-phase between each task, i.e., an upper-bounded frequency. The upper-bounded frequency ensures that the previous blocks have been accepted by the *Data DAG* with high likelihood. This prevents the out-of-order of transactions [24], and unexpected rollbacks or forks by leveraging a block-ordering in the case where the data uploading is conducted in a very high rate. In this paper, the flexibility of publishing tasks without having to limited by a certain period is our focus.

**Algorithm 1:** Decentralized data uploading mechanism

```
▷ Define
    Controller$_i$ and Gateway$_{i,j}$; // The $i$-th LoRa Controller and its corresponding $j$-th LoRa Gateway.
    Receiver $\leftarrow$ Sender.***Send***(Message); // A sender sends msgs to a receiver.
    Blockchain $\Leftarrow$ Sender.***Upload***(Message); // A sender uploads msgs to the chain.
    Gateway$_{i,j}$ $\leftarrow$ Controller$_i$.***Incentive***(data$_{i,j,k}$); // The $i$-th LoRa Controller pays incentive to $j$-th LoRa Gateway on the ID Chain upon the data uploaded in task$_k$.
    FlowControl // Enabling the flow control
▷ Registration
    Controller$_i$ $\leftarrow$ Gateway$_{i,j}$.***Send***(info);// where info contains the information of Gateway$_{i,j}$ (e.g., Blockchain address).
    Blockchain$_{ID}$ $\Leftarrow$ Controller$_i$.***Upload***(info);
▷ Uploading data
    Controller$_i$ publishes $Task_{i,k}$ with $Timeout_{i,k}$, records $TimeStamp_{start}$.
    if FlowControl is enabled then
        Invoke Algo. 3.Defining Congestion
    while $Timeout_{i,k}$ not yet reached do
        Controller$_i$ $\leftarrow$ Gateway$_{i,j}$.***Send***(LoRa_data); // data_size and data_hash are included in LoRa_data apart from the data itself. LoRa_data is secured by the asymmetric encryption.
        if the sending data from Gateway$_{i,j}$ is invalid then
            alarm and drop
        else
            Controller$_i$ records $TimeStamp_{j,t}$, i.e., the actual time that Gateway$_{i,j}$ sends LoRa_data.
    // The controller ranks and pays Gateway$_{i,j}$ with different types in $Tree_{i,k}$ based on the inbound data size |data$_{i,j,k}$| for $Task_{i,k}$.
    if FlowControl is enabled then
        Invoke Algo. 3.Incentive Allocation
    else
        Gateway$_{i,j}$ $\leftarrow$Controller$_i$.***Incentive***(|data$_{i,j,k}$|)
    // Invoke Algo. 2 where $i' \neq i$, to validate a pending block proposed by another controller.
    block$_{i,k}$ $\leftarrow$Controller$_i$.***Validate***(block$_{i',k}$).$Result$;
    Data DAG $\Leftarrow$ Controller$_i$.***Upload***(block$_{i,k}$);// where $Tree_{i,k}$ is contained in the block$_{i,k}$.
    The uploading succeeds only if Algo. 2 returns TRUE during the cross-validation of other forthcoming blocks.
```

**Algorithm 2:** Decentralized cross-validation mechanism

```
Output: TRUE or FALSE
▷ Define
    Result ← Validator.Validate(Message); // A validator validates some messages and return a bool result.
▷ Cross-Validation
    Result ← $\text{Controller}_i$.Validate($\text{block}_{i',k}.Tree$),   $i' \neq i$.
{
    if $block_{i',k}$ self-validating or lazy-validating then
        return FALSE
    for each $\text{LoRa\_data}_{i',j,k}$ in $\text{block}_{i',k}.Tree$ do
        Result ← $\text{Controller}_i$.Validate($\text{LoRa\_data}_{i',j,k}$) {
            // Asymmetric-decrypting $\text{LoRa\_data}_{i',j,k}$
            if $Gateway_{i',j}$ info $\notin$ $Blockchain_{ID}$ then
                return FALSE
            // Validate the PoTO
            if $|data_{i,j,k}| <$ lower-bound OR $|data_{i,j,k}| \neq data\_size_{i',j,k}$ then
                return FALSE
            if $Hash(data_{i,j,k}) \neq data\_hash_{i',j,k}$ then
                return FALSE
        }
    end for each
    // Validate the type-classification
    if FlowControl is enabled then
        Invoke Algo. 4.Cross-Validation for Flow Control
    else
        if Line 16 of Algo. 1 is NOT properly executed then
            return FALSE
    otherwise return TRUE
}
```

*Data uploading from LoRa Controllers to Data DAG:* A controller first classifies the gateways which have completed the task into different types based on the data size that gateways have forwarded. The types are designed to fit in a tree structure, implying that type-1 contains the smallest number of gateways, and the number of gateways increases with the type index. The larger amount of data is received by the controller, the higher probability the gateway is classified in a type that is closer to the root of the tree. With a closer position to the root of the tree, the gateways are offered a higher incentive by the controller. The incentive is subsequently applied in a block which the controller has generated and broadcast. The block verifies a certain number of pending blocks in the *Data DAG* by pointing to their block hash values. The typical Proof-of-Work (PoW) is replaced by a more efficient protocol, PoTO; see cross-validation in the following.

**Cross-Validation -** The data uploading succeeds only if the pending block passes the cross-validation by all the participating controllers (or more specifically, validated by sufficient numbers of forthcoming blocks proposed by other controllers); see Algo. 2. By introducing a weighting factor to encourage blocks to cross-validate the tips of the DAG, the risk of a malicious controller conducting self-validation (i.e., validating its own previous blocks) and lazy-validation (i.e., validating any ancient blocks) can be reduced. Also, a minimum size of payloads is compulsory to prevent distributed denial-of-service (DDoS) attacks and Sybil attacks, thus providing spam protection. Each transaction of the tree in a pending block can be associated with the LoRa data forwarded from each gateway to a specific controller for a published task. In regards to each transaction, other controllers can retrieve information, including the data size, data hash, and data payload after identifying the transaction, if the information of the gateway has been recorded on the *ID Chain*. Thus, the cross-validation can be conducted associated with the following aspects.

*Identity* (Lines 6-7 in Algo. 2): The information of each gateway and controller is secured by the tamper-resistance of the *ID Chain*. The missing information of a gateway on the *ID Chain* leads to a possibility of the controller colluding with an unregistered gateway by packetizing its transaction into a pending block.

*Proof-of-Task-Overhead (PoTO)* (Lines 8-11 in Algo. 2): The PoW used in typical IOTA is a simple computational operation dedicated for spam protection and the defense to DDoS attacks [32]. Our proposed PoTO protocol, featuring the integration of Blockchain-LoRa, can delivery the same protection without need of additional computational operations required by the PoW used in IOTA. In particular, the computational operations can be omitted, as the limited data source transmitted in a physical channel, the limited data size, and the limited number of gateways inherently extend a period of time for every task. Even the malicious controllers would have to comply with the system restriction, thus leading to a controllable growth rate of the *Data DAG*. Based on this property, the amount of data received by the controllers (the data refers to the size of data payload of each transaction in a pending block on the *Data DAG*) are enough to be the amount of "work" done during the data collection and validation process. As a result, the PoTO protocol compromises the motivation to launch attacks. This is because controllers are not responsible for the validity of the metadata based on our Dual-Chain-based structure. Malicious controllers spending time and consuming communication (downloading/uploading) and computation (verifying/smart contract operations) resources to upload local corrupted LoRa data do not affect the interests of others and reap no profits for itself (See Section 4.3.3 for more details on the security analysis).

*Type-classification* (Lines 13-17 in Algo. 2): Recall that the rule of type-classification for each registered controller is recorded in the smart contract on the *ID Chain*. Global cross-validation regarding the type-classification can also be conducted to avoid any cheating, for example accepting bribes from a gateway and assigning it with a higher type than it deserves. Based on the data size from the last validation item, the other controllers can verify whether a specific controller has complied with the type-classification rule stored on the *ID Chain*. Thus, the incentive offered to each type of the gateways can subsequently be cross-validated by checking the balance of the gateways claimed by the pending block.

### *2.2. Flow Control Protocol in a self-driven way*

In a typical LoRaWAN network, end-devices are connected to multiple LoRa Gateways, and hence the end-devices transmit data to multiple connected gateways [33]. The gateways transmit transactions to the corresponding controller, resulting in duplicated data at the controller. The duplicated data needs to be handled specifically because: 1) the *Data DAG* incurs a throughput loss and a data storage waste; and 2) the duplicated data still accounts for the uploading data size of the gateway during the type-classification, which can result in unfair competition. Thus, a discount applied to the duplicated data as the penalty is introduced to reduce unfair competition.

We propose a new flow control protocol which is designed to effectively allocate a fair and proper amount of incentive to each contributing gateway based on its uploading data size. Being part of the proposed contract-theoretic incentive mechanism (see Section 3) to reduce the unfairness, the flow control is not aimed to control the transmission in a compulsory way (e.g., cutting off the transmission). Instead, it is a management tool to incentivize the gateways and make them comply with any preconcerted rules in a self-driven way, thus achieving the scalability as expected. Exceptions (e.g., sending urgent messages) are allowed. In other words, a gateway can still upload data anytime without caring about the flow control if the gateway has a strong wish to do so. The controller maintains an internal clock which determines a time-frame for every single gateway, while the controller can still receive the transmission from all the gateways. This motivates the gateway to transmit as much as possible in their own time-frames, in order to reduce the ratio of duplicated data and maintain the participation of an adequate number of gateways in a specific region.

#### *2.2.1. Congested LoRa Gateways*

(Lines 1-7 of Algo. 3) Each gateway has a specific coverage area to serve end-devices. Multiple gateways in a region may have some overlapping coverage.

**Algorithm 3:** Flow Control: Congested LoRa Gateways and Incentive Allocation

```
▷ Defining Congestion
    Controller$_i$ uploads randomness for $Task_{i,k+1}$ to the
    *ID Chain* using Blockchain-based randomness generator.
    while $[Gateway_{i,j}, \dots, Gateway_{i,n}]$ do
        // $\lambda_i$ denotes the overlapped proportion
        if Overlap($Gateway_{i,j}$, $Gateway_{i,j+1}$)>$\lambda_i$ then
            Controller$_i$ records Gateway$_{i,j}$.$DAGAddr$,
            count the total number of congested gateways $n$.
    for each $j$ of $Gateway_{i,j}$ do
        $TimeStamp_j = TimeStamp_{start} + Timeout_{i,k} \times \frac{j}{n}$
        Scheduler$_{i,k}$ ← Controller$_i$.Scheduler(
        Gateway$_{i,j}$.$DAGAddr$, $TimeStamp_j$)
▷ Incentive Allocation
    if $TimeStamp_{j-1} < TimeStamp_{j,t} \leq TimeStamp_{j+1}$ then
        Gateway$_{i,j}$ ←Controller$_i$.Incentive($|\mathrm{data}_{i,j,k}|$)
    else
        Gateway$_{i,j}$ ←Controller$_i$.Incentive($\frac{|\mathrm{data}_{i,j,k}|}{n}$)
```

Within the overlapping coverage, the end-devices connect to multiple gateways and are most likely to transmit duplicated data. Thus, we define the congested gateways as the gateways which have a specific common overlapping coverage area. A gateway registered on the *ID Chain* leads to the essential information, including the geographical location, device model, coverage area stored in a smart contract. The overlapped proportion is also defined in the smart contract, defined as the overlapped area divided by the total coverage area of a single gateway.

Through the overlapped proportion and the coverage area of each gateway recorded on the *ID Chain*, the corresponding controller is able to determine the congested gateways as a list $[Gateway_{i,j}$, $Gateway_{i,j+1}$, $\dots$ ,$Gateway_{i,n}]$ where $i$ indicates the $i$-th controller; $j$ indicates the $j$-th gateway; and $n$ indicates the number of congested gateways within a specific overlapping coverage). Also, a scheduler is defined in a smart contract and secured by the *ID Chain*. The scheduling for $Task_{i,k}$ is based on the pre-defined randomness announced on the *ID Chain* by using the Blockchain-based randomness generator such as RANDAO [34]. The scheduler records the identity and the assigned time-frame of each congested gateway. The controller refers to the scheduler determining whether a gateway transmits data within its own time-frame hence marked as a full contribution.

The contract-theoretic task-$k$ published by the controller-$i$ includes a time-out $Timeout_{i,k}$ which is a single epoch of the flow control. At the same time of publishing a task, a local timer is triggered on the controller and the initial timestamp $TimeStamp_{start}$ is recorded. Meanwhile, the controller checks the congested gateways based on the information (e.g., the overlapped proportion) of the gateways in the smart contract. The period of an epoch, $Timeout_{i,k}$, is divided into $n$ time-frames which equals to the number of congested gateways. Therein, each boundary point is defined as the unique timestamp for each gateway-$j$ $TimeStamp_j$ (Line 6 of Algo. 3)[3]. Finally, the controller updates ev-

[3] Generally, multiple gateways as a group can be allocated in one time-frame based on the

ery single timestamp $TimeStamp_j$ to the scheduler based on the identity of the congested gateway $Gateway_{i,j}.DAGAddr$. Note that using a single randomness for multiple tasks (i.e., multiple epochs) is permitted to prevent updating smart contracts for every task, and reduce the transaction load on the *ID Chain* which is typically poor at scalability.

### *2.2.2. Incentive allocation*

(Lines 8-10 of Algo 3) Recall that the incentive allocated to each gateway is based on the uploading data size as the contribution. The flow control is an internal management tool for controllers to adjust the contribution of each gateway during a task, in order to allocate a proper incentive to the gateways and provide a fair type-based classification. The gateway can either check the *Data DAG* before transmitting transactions to reduce the ratio of duplicated data, or transmit without checking due to reasons such as saving query resources, enhance the data redundancy, or the willingness for sharing the coverage. Thus, transmitting data within its own time-frame is strongly encouraged with a high incentive, thus motivating gateways to reduce the ratio of duplicated data. In other words, if the transaction is uploaded by a gateway within its own time-frame, a full incentive is offered to the gateway based on its uploading data size. While if the transaction is uploaded by a gateway out of its corresponding time-frame, a partial incentive is offered based on the data size as the penalty. Here, the partial incentive is calculated by the original size of the uploading data divided by the number of congested gateways, $n$.

### *2.2.3. Cross-validation of Flow Control*

The above incentive allocation will not be conducted until passing the global cross-validation of flow control among other controllers. Recall that finalizing a pending block needs to validate a sufficient number of other pending blocks. By specifying the transactions in an arbitrary pending block uploaded to the

live requirements. For simplicity, this paper considers a one-gateway-one-timeframe scheme.

**Algorithm 4:** Flow control: decentralized cross-validation mechanism

```
  Output: TRUE or FALSE
  ▷ Cross-Validation for Flow Control
1     Result ← Controller$_i$.Validate($Scheduler_{i',k}$),   $i' \neq i$.
  {
2     if $Gateway_{i,j}.DAGAddr \notin Scheduler_{i,k}$ then
3         return FALSE
4     if the pre-defined randomness is NOT matched with $Scheduler_{i,k}$ then
5         return FALSE
6     if Algo. 3.Incentive Allocation is NOT properly executed then
7         return FALSE
8     otherwise return TRUE
  }
```

*Data DAG*, all the other controllers can validate the transactions via their own proposed pending blocks, through the following aspects:

- *Identity of gateway* (Lines 2-3 of Algo. 4): The controllers validate whether the identity (e.g. Blockchain Address) belongs to one of the congested gateways based on the scheduler.
- *Scheduler* (Lines 4-5 of Algo. 4): The controllers validate whether the presented scheduler for a task is matched with the publicly pre-defined randomness.
- *Timestamp and Incentive* (Lines 6-7 of Algo. 4): The controllers compare the timestamp of the transaction with the time-frame of the gateway that is recorded in the scheduler. After that, the controllers calculate the amount of incentive and compare it with the intended amount.

## 3. Incentive Mechanism for LoRa Gateways Using Contract Theory

An efficient LoRa-IS requires high coverage. In regard to motivating more to participate in the deployment of LoRa Gateways to expand the coverage, we aim to design an incentive mechanism where the corresponding LoRa Controller will offer the token reward based on the amount of each gateway has contributed. However, a controller does not have any prior knowledge about the performance of gateways (e.g., the hardware performance) and the amount of LoRa data each gateway is willing to forward as a contribution. The information asymmetry between the controller and gateways needs to be tackled to reduce the cost of the incentive while maximizing the utility of both controllers and gateways. In this paper, we adopt contract theory [16] in our incentive mechanism design that can be integrated into a Blockchain-based LoRa system. In other words, the reward can be matched with how much each gateway should deserve in terms of the contribution without any bias, which guarantees efficiency and fairness. The whole process of the adoption of the contract theory is conducted during the contract initialization in the smart contract of *ID Chain* (referred to Section 2.1), and is discussed in the following.

During packetizing the block with task index $k$ by a controller $i$ chosen by the consensus process, a monopoly market [16] is considered. Therein, the market consists of a controller acting as the task publisher and a set of gateways $\mathbb{N} = \{\mathcal{N}_1, \dots, \mathcal{N}_j, \dots, \mathcal{N}_N\}, 1 \leq j \leq N$. For a gateway $\mathcal{N}_j$, the total amount of receiving data forwarded from the end-devices to the controller is denoted as $\mathbb{Q}_j^{i,k}$ for task-$k$ published by controller-$i$. Any $\mathbb{Q}$ is resource-intensive in terms of the bandwidth, the performance of receiver, the number of supported bands, etc. To be specific, $\mathbb{Q}$ can be abstracted into $\mathbb{R}$ and $t$, denoting the single channel bandwidth (bps) and the usage (second) of this channel on this bandwidth during the maximum task period of task-$k$, respectively. Note that, (1) represents the vector of the logic channel characterized by a pair $BW$ and $SF$ (denoting the bandwidth in Hertz and the spreading factor), while $CR$ denotes the code rate defining the level of tolerance to signal interference[33].

$$\mathbb{R} = \mathbb{R}(SF, BW, CR) = SF \times \frac{BW}{2^{SF}} \times CR. \tag{1}$$

Thus, we define $F$ types based on the heterogeneous willingness in terms of the total uploading data size $B$. We assume that the gateways are sorted in an ascending order of the contributed data: $\theta_1 < \cdots < \theta_f < \cdots < \theta_F$. The greater

$\theta_f, f \in \{1, \ldots, F\}$ implies the more LoRa data that this type of gateways have forwarded from the end-devices to the controller[35, 36]. Thus, each task $\mathbb{Q}_j^{i,k}$ is denoted by a two-tuple $(\mathbb{R}_j^{i,k}, t_j^{i,k})$.

Each controller needs to encounter the information asymmetry while they aim to minimize the economic loss. For all $F$ types $\theta_f$ customized by a controller in a pre-defined contract by a controller, the contract is a series of uplink-data-reward bundles denoted by $(T_f(B_f), B_f)$. $B_f$ denotes the total uploading data size $B$ of the type-$\theta_f$ gateways and $T_f(B_f)$ is the incentive offered to the gateways. Also note that, the gateways which forward more data from the end-devices to the corresponding controller under a valid threshold upper-bounding the uploading data size $B$ (i.e., $B_{max}$) can be rewarded more.

### *3.1. Uploading Data Size in LoRa Gateways*

We define the uploading data size $B$ as the *significant* data size. Here, the term "*significant*" indicates that the received LoRa data is de-duplicated (i.e., repeated data being discounted as the penalty) after decoded at the controller side. This is due to the fact that multiple gateways may receive and forward identical messages sent from a single end-device either with the same or different logic channels [33]. In addition, we define $B_f$ taking the following factors into consideration:

- the number of end-devices served by the type-$\theta_f$ gateways;
- the latency for the type-$\theta_f$ gateways to complete the task (receiving data and forwarding out) associated with both inbound and outbound bandwidth;
- the coverage area and the overlapping coverage area subject to the proposed flow control protocol (see Section 2.2);
- the amount of duplicated data.

We consider a specific region. For the $j$-th gateway completing the task-$k$, the significant data size $B_f(\mathbb{R}_j^k, t_j^k)$ for data forwarding task $\mathbb{Q}_j^k$ can be given by

$$B_f(\mathbb{R}_j^k, t_j^k) = \sum^{l} (\mathbb{R}_j^k t_j^k)_l, \tag{2}$$

where $\mathbb{R}_j^k$ and $t_j^k$ denote the vector of logic channels and the time of transmission on each discrete logical channel out of total $l$ channels during the task $\mathbb{Q}_j^k$, respectively. $B_f$ denotes the size of uploading data from the end-devices to the gateways and should be upper-bounded by $B_{max} = \sum \mathbb{R}_j^k \mathbb{T}_{max}$, where $\mathbb{T}_{max}$ is the task period (i.e., an epoch period). In (2), $t_j^k \leq \mathbb{T}_{max}$ and $B_f \leq B_{max}$. Note that the actual significant data size $B_f > B_{max}$ may happen due to maliciously forwarding accumulated data to the controller, which is detectable at the controller side by comparing $B_{max}$ with the actual inbound uploading data size from gateways. This is because the link between gateways and controllers can be a fixed, wired, or backbone network. Thus, the total size of the accumulated data can be much bigger than $B_{max}$.

### *3.2. Utility of LoRa Controllers*

Based on the signed contract $(T_f, B_f)$ between a controller and the type-$\theta_f$ gateways, the utility of the controller earned through the task $\mathbb{Q}^k$ from the type-$\theta_f$ gateways is given by

$$U_c(\theta_f) = \mu(B_f) - \omega T_f, \tag{3}$$

where

$$\mu(B_f) = \begin{cases} \frac{\sigma}{(B_{max} - B_f)} & \text{if } B_{max} \geq B_f, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

There is a penalty applied to the gateways uploading excessive data to the controller (i.e., $T_f(B_{max} < B_f) = 0$). Note that $\sigma > 0$ is a pre-defined parameter and (4) implies $B_f$ (closer to $B_{max}$) can attain a larger $\mu(B_f)$. Thus, the goal of the controller is to maximize its total utility all through the $F$ types of gateways, as given by

$$\max_{(T_f, B_f)} \quad U_c = \sum_{f=1}^{F} p_f N(\mu(B_f) - \omega T_f), \forall f \in \{1, \ldots, F\}, \tag{5}$$

where $N$ is the total number of gateways that the controller is in charge of; $\omega$ is a pre-defined parameter; and $p_f$ is the prior probability of the type-$\theta_f$ gateways with $\sum_{f=1}^{F} p_f = 1$. Note that the controller can attain the distribution based on the historical statistics, and we assume a uniform distribution among the $F$ types gateways which the controller is aware of [35, 36, 15]. The controller is also aware of the value of $N$, as a pre-registration on the *ID Chain* is needed to activate gateways in the coverage.

### *3.3. Utility of LoRa Gateways*

For the type-$\theta_f$ gateways completing the task-$k$ based on the signed contract $(T_f, B_f)$, the utility function is given by

$$U_f = \theta_f \nu(T_f) - \phi B_f, \forall f \in \{1, \ldots, F\}, \tag{6}$$

where $\phi$ is the unit resource cost of data forwarding; $\nu(T_f)$ is the evaluation function of the type-$\theta_f$ gateways in terms of the incentive $T_f$. The evaluation function $\nu(T_f)$ monotonically increases with the following properties [16]

- $\frac{\partial \nu}{\partial T_f} > 0$, monotonically increasing;
- $\frac{\partial^2 \nu}{\partial T_f^2} < 0$, concavity;
- $\frac{\partial \nu}{\partial \theta_f} > 0$, positive correlation of data contribution; and
- $\nu(0) = 0$, scheme for non-incentive.

The goal of all $F$ types of gateways is to maximize the utility earned by data forwarding, as given by

$$\max_{(T_f, B_f)} \quad U_f = \theta_f \nu(T_f) - \phi B_f, \forall f \in \{1, \ldots, F\}. \tag{7}$$

Based on (5) and (7), our objective is to maximize the utility of the controllers and the utility of the gateways at the same time, while they are, in fact, contradictory. To solve the conflicting problem, the contract theory is used to design a series of optimal type specific contract ($T_f^*$,$B_f^*$).

### *3.4. Problem Transformation and Optimization*

We introduce the contract theory in the LoRa context. Given that the utility of LoRa Gateways defined by (6), each contract item for the gateways needs to satisfy *Definitions 1-2* [16].

*Definition 1:* **Individual Rationality (IR)**. It means a non-negative utility should be attained for each gateway participating in data forwarding, i.e.,

$$\theta_f \nu(T_f) - \phi B_f \geq 0, \forall f \in \{1, \ldots, F\}. \tag{8}$$

*Definition 2:* **Incentive Compatibility (IC)**. It means only the contract $(T_f, B_f)$ dedicated for type-$\theta_f$ can maximize the utility of the type-$\theta_f$ gateways than any other contracts $(T_{f'}, B_{f'})$, i.e.,

$$\begin{aligned} &\theta_f \nu(T_f) - \phi B_f \geq \theta_f \nu(T_{f'}) - \phi B_{f'}, \\ &\forall f, f' \in \{1, \ldots, F\}, f \neq f'. \end{aligned} \tag{9}$$

Thus, the optimization problems ( 5) and ( 7) can be defined in (10), where the first two constraints refer to IR and IC, respectively.

$$\begin{aligned} \max_{(T_f, B_f)} \quad & U_c = \sum_{f=1}^{F} p_f N(\mu(B_f) - \omega T_f) \\ s.t. \quad & \theta_f \nu(T_f) - \phi B_f \geq 0, \\ & \theta_f \nu(T_f) - \phi B_f \geq \theta_f \nu(T_{f'}) - \phi B_{f'}, \\ & \forall f, f' \in \{1, \ldots, F\}, f \neq f'. \end{aligned} \tag{10}$$

To solve problem (10), we relax the complicated constraints ($F$ number of IR constraints and $F(F-1)$ number of IC constraints) and transform (10) to attain a more tractable set of constraints. After that, we need to solve the relaxed optimization problem without monotonicity (*Corollary 1*) and verify whether the monotonicity is satisfied with the solution.

Firstly, we need to prove that the utility of the gateways (referred to (6)) satisfies the Spence-Mirrlees property [16].

*Definition 3*: **Spence-Mirrlees Property**. Eq. (6) satisfies the property if and only if

$$\frac{\partial}{\partial \theta}[-\frac{\partial U/\partial T}{\partial U/\partial B}] > 0. \tag{11}$$

We need to prove that the utilities of the gateways satisfy *Definition 3*. *Proof:* See Appendix Appendix A. ■

Thus, we can prove that the utilities of the gateways satisfy the Spence-Mirrlees property based on (6). Based on the proof, we present the following corollaries.

*Corollary 1:* **Monotonicity Condition**. For contracts ($T_f$,$B_f$) and ($T_{f'}$,$B_{f'}$), if $\theta_f \geq \theta_{f'}$, $\forall f, f' \in \{1, \ldots, F\}, f \neq f'$, then $T_f \geq T_{f'}$.

This is proved if the following is true.

*Lemma 1:* $T_f \geq T_{f'}$ iff $B_f \geq B_{f'}$, $\forall f, f' \in \{1, \ldots, F\}, f \neq f'$.

*Proof:* See Appendix Appendix B. ■

*Lemma 2:* The IR constraint (*Definition 1*) of (7) can be reduced as $\theta_1 \nu(T_1) - \phi B_1 \geq 0$.

*Proof:* See Appendix Appendix C. ■

*Definition 4*: **Downward Incentive Compatibility (DIC) and Upward Incentive Compatibility (UIC).** The IC constraint can be fifty-fifty split between DIC and UIC, as given by

$$DIC : \theta_f \nu(T_f) - \phi B_i \geq \theta_f \nu(T_{f-1}) - \phi B_{f-1}, \forall f \in \{2, \ldots, F\};$$

$$UIC : \theta_f \nu(T_f) - \phi B_i \geq \theta_f \nu(T_{f+1}) - \phi B_{f+1}, \forall f \in \{1, \ldots, F-1\}. \qquad (12)$$

*Corollary 2:* The IC constraint of (7) can be transformed into the Local DIC (LDIC) by utilizing the monotonicity in *Corollary 1*, as given by

$$\begin{cases} \theta_{f+1} \nu(T_{f+1}) - \phi B_{f+1} \geq \theta_{f+1} \nu(T_f) - \phi B_f; \\ \theta_f \nu(T_f) - \phi B_f \geq \theta_1 \nu(T_{f-1}) - \phi B_{f-1}, \end{cases} \qquad (13)$$

where three continuous types of gateways, i.e., $\theta_{f-1} < \theta_f < \theta_{f+1}, \forall f \in \{2, \ldots, F-1\}$, are considered to prove DIC that can be reduced to LDIC.

*Proof:* See Appendix Appendix D. ■

With the **Monotonicity Condition** (*Corollary 1*) and LDIC (*Corollary 2*), DIC holds. On the other hand, UIC holds with similar approaches combining *Corollary 1* and LUIC, which can be omitted due to the basic two-type analysis [16].

*Lemma 3:* LDICs bind at the optimum together with the monotonicity of *Corollary 1.*

*Proof:* See Appendix Appendix E. $\blacksquare$

Based on *Lemmas 1-3*, we can convert (10) into (14)

$$
\begin{aligned}
&\max_{(T_f, B_f)} \quad U_c = \sum_{f=1}^{F} p_f N(\mu(B_f) - \omega T_f) \\
s.t. \quad & \theta_1 \nu(T_1) - \phi B_1 = 0, \quad \textcircled{1} \\
& \theta_f \nu(T_f) - \phi B_f = \theta_f \nu(T_{f-1}) - \phi B_{f-1}, \quad \textcircled{2} \\
& \forall f \in \{2, \ldots, F\}, \\
& T_f \geq T_{f-1} \geq \cdots \geq T_1, \\
& \theta_f > \theta_{f-1} > \cdots \theta_1, \\
& \text{and } B_f \leq B_{max}.
\end{aligned}
\tag{14}
$$

*3.5. Solving the Optimization Problem*

The optimization problem in (14) is solved sequentially

1. solving the reduced problem without the **Monotonicity Condition**;
2. verifying whether the solution to the reduced problem satisfies the monotonicity condition.

Recall that $\nu(\bullet)$ is a monotonically increasing and concave function because $\frac{\partial^2 \nu}{\partial T_f^2} < 0$. To conduct action 1), an iteration of ① and ② in (14) is conduct to express $T_f$, as shown below

$$
T_f = \frac{\phi B_1}{\theta_1} + \sum_{k=2}^{f} \Delta_k, \text{ where } \Delta_k = \frac{\phi B_k}{\theta_k} - \frac{\phi B_{k-1}}{\theta_k} \text{ and } \Delta_1 = 0. \tag{15}
$$

In turn, (14) can be transformed to the following, $\forall f \in \{1, \ldots, F\}$

$$\begin{aligned} U_c &= \sum_{f=1}^{F} p_f N(\mu(B_f) - \omega T_f) \\ &= \sum_{f=1}^{F} p_f N \sigma (B_{max} - B_f)^{-1} \\ &\quad - N\omega \sum_{f=1}^{F} p_f [\frac{\phi B_1}{\theta_1} + (\frac{\phi B_k}{\theta_k} - \frac{\phi B_{k-1}}{\theta_k})], \end{aligned} \tag{16}$$

where

$$\sum_{f=1}^{F} p_f [\frac{\phi B_1}{\theta_1} + (\frac{\phi B_k}{\theta_k} - \frac{\phi B_{k-1}}{\theta_k})] = \sum_{f=1}^{F} \pi_f B_f,$$

$$\pi_f = \begin{cases} (\sum_{i=f+1}^{F} p_i)(\frac{\phi}{\theta_f} - \frac{\phi}{\theta_{f+1}}) + \frac{\phi p_f}{\theta_f}, 0 < f < F \\ \frac{\phi p_f}{\theta_f} \qquad\qquad\qquad\qquad\quad f = F. \end{cases}$$

*Proof:* See Appendix Appendix F. ■

Thus,

$$\begin{aligned} \max_{B_f} \quad & U_c = \sum_{f=1}^{F} \frac{p_f N \sigma}{(B_{max} - B_f)} - N\omega \sum_{f=1}^{F} \pi_f B_f \\ s.t. \quad & 0 \le B_f \le B_{max}, \\ & N \sum_{f=1}^{F} \pi_f B_f \le \mathbb{T}_{max}, \\ & \forall f \in \{1, \ldots, F\}. \end{aligned} \tag{17}$$

We can subsequently calculate the second derivative to confirm that $U_c$ is concave in (17), and the constraints are affine.

$$\frac{\partial^2 U_c}{\partial B_f^2} = -\frac{2 p_f N \sigma}{(B_{max} - B_f)^3} < 0. \tag{18}$$

Therefore, we can obtain the optimal $B_f^*$ and the corresponding incentive $T_f^*$ by using the convex optimization tools to solve the optimization problem in (17).

### *3.6. Practical Implementation*

The following steps are conducted for the practical implementation of the contract-theoretic incentive mechanism, as shown in Fig. 2. A LoRa Controller,

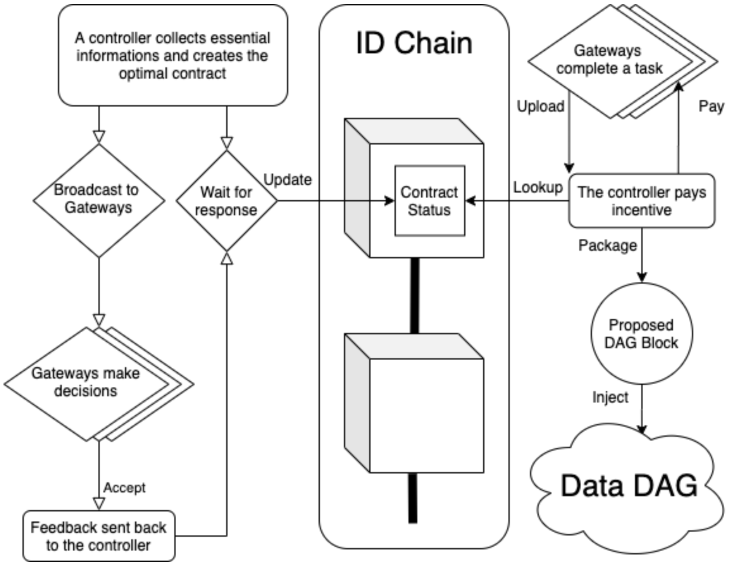


Figure 2: The flow chart of practical implementation of the contract-theoretic incentive mechanism to relieve the information asymmetry

acting as the task publisher, obtains the values of any relevant information based on the historical data. Such information includes the LoRa settings i.e., ($SF$, $BW$, and $CR$), the number of LoRa Gateways under the management, and the expected value of significant uploading data size. Thus, the controller can calculate the optimal contract along with the types, i.e., $\theta_f \nu(T_f)$, and subsequently broadcasts this term to the gateways via the internet. By evaluating the utility $U_f$ based on the contract received from the controller, each of the gateways decides whether to participate in the task, and chooses one option in the contract, i.e., $(T_f, B_f)$, by sending back feedback to claim its willingness for signing the contract with the controller. The above is recorded and updated with the status of smart contract in the *ID Chain*. Finally, after the gateways establish the task to forward the LoRa data from the end-devices to the controller with the agreed value of significant uploading data size, the controller proposes a new DAG block. Therein, the preconcerted incentive to each type of gateways can be paid based on the corresponding contractual obligation recorded in the contract of *ID Chain*. Thus, the information asymmetry can be relieved by implementing the contract-theoretic incentive mechanism.

## 4. Benchmark, Simulation, and Discussion

In this section, we first provide the utility of LoRa Controllers in other benchmark mechanisms, including the upper-bounded centralized optimization and Stackelberg optimization with symmetric/asymmetric information. Next, we conduct simulations by evaluating and comparing the proposed contract-theoretic incentive mechanism with such benchmark mechanisms. In addition, an evaluation of the impacts on the entire dual-chain system with the proposed flow control and incentive mechanism is also simulated.

### *4.1. Evaluation of the Contract-theoretic Incentive Mechanism*

Inspired by [15], we compare the proposed contract-theoretic incentive mechanism with the existing benchmark mechanisms under symmetric and asymmetric information. In such a way we can investigate the impacts of the symmetric/asymmetric information on the incentive results, and to what extent the contract-theoretic incentive mechanism can overcome the asymmetric information. Firstly, the centralized optimization mechanism under symmetric information is discussed. Subsequently, we present the discussion about the Stackelberg optimization with symmetric/asymmetric information.

#### *4.1.1. Centralized Optimization*

The controllers completely knowing the types of corresponding gateways is the most significant feature of a centralized optimization mechanism. Such feature also leads to the centralized optimization mechanism being the upper-bound of the performance among all incentive mechanisms. The centralized optimization problem is expressed as follows.

$$\begin{aligned} \max_{(T_f,B_f)} \quad & U_c = \sum_{f=1}^{F} p_f N(\mu(B_f) - \omega T_f) \\ s.t. \quad & \theta_1 \nu(T_1) - \phi B_1 \geq 0, \forall f \in \{1, \dots, F\}. \end{aligned} \tag{19}$$

Eq. (19) can subsequently result in the conversion of $\theta_f \nu(T_f^*)$ associated with the optimal prices $T_f^*$ due to the visible types of the gateways, as given by

$$\theta_f \nu(T_f^*) = \phi B_f$$

$$\nu(T_f^*) = \log(T_f^* + 1) = \frac{\phi B_f}{\theta_f}, \forall f \in \{1, \ldots, F\}, \tag{20}$$

where we consider $\nu(T_f^*) = \log(T_f^* + 1)$ which satisfies the concavity and $\nu(0) = 0$. Thus, $T_f$ can be substituted by $T_f^*$

$$\begin{aligned} \max_{(T_f, B_f)} \quad U_c &= \sum_{f=1}^{F} p_f N(\mu(B_f) - \omega T_f^*) \\ &= \sum_{f=1}^{F} p_f N(\mu(B_f) - \omega \exp{(\frac{\phi B_f}{\theta_f} - 1)}) \end{aligned} \tag{21}$$

$$s.t. \quad 0 \leq B_f \leq B_{max}, \forall f \in \{1, \ldots, F\}.$$

*4.1.2. Stackelberg Optimization with Asymmetric Information*

Different from our proposed incentive mechanism, where LoRa Controllers relieve the information asymmetry by providing every types of LoRa Gateways with the unique, limited, and properly designed contract-theoretic incentive options, a vector of prices $\boldsymbol{\lambda}= [\lambda_1, \lambda_2, \ldots, \lambda_N]^T$ are used to denote the unit price per significant throughput $B$ for $N$ gateways in the Stackelberg optimization problem. For simplicity, we use type-$\theta$ to denote every single gateway. The utility of gateway $f$ in the Stackelberg optimization problem with asymmetric information is given by

$$U_f = \theta_f \nu(T_f) - \phi B_f = \theta_f \log(T_f + 1) - \phi B_f, \tag{22}$$

where $T_f = \lambda_f B_f$ denotes the total price that the controller requires to pay. To maximize each gateway's profit, the following is calculated to attain the optimal $T_f^*$ and $B_f^*$

$$\frac{\partial U_f}{\partial B_f} = \frac{\lambda_f}{\lambda_f B_f + 1} - \phi = 0,$$

$$B_f^* = \frac{\lambda_f - \phi}{\lambda_f \phi},$$

$$T_f^* = \lambda_f B_f^* = \frac{\lambda_f - \phi}{\phi}. \tag{23}$$

Thus, we can obtain

$$U_c = \sum_{f=1}^{F} p_f N[\mu(\frac{\lambda_f - \phi}{\lambda_f \phi}) - \frac{\lambda_f - \phi}{\phi}]. \tag{24}$$

It can be observed that the information asymmetry has a more severe impact on the incentive than the proposed contract-theoretic incentive mechanism. This is due to the fact that gateways can finally transmit any amount of $B$ to match up with any price imposed by the controller, which makes it impossible to adapt to the frequent change of the instantaneous combination of the types of gateways. The restriction stops the controller to relieve the information asymmetry by a timely recognition of the types.

### *4.1.3. Stackelberg Optimization with Symmetric Information*

The Stackelberg Optimization performs much better under symmetric information because the instantaneous utility of gateways can be optimized, as given by

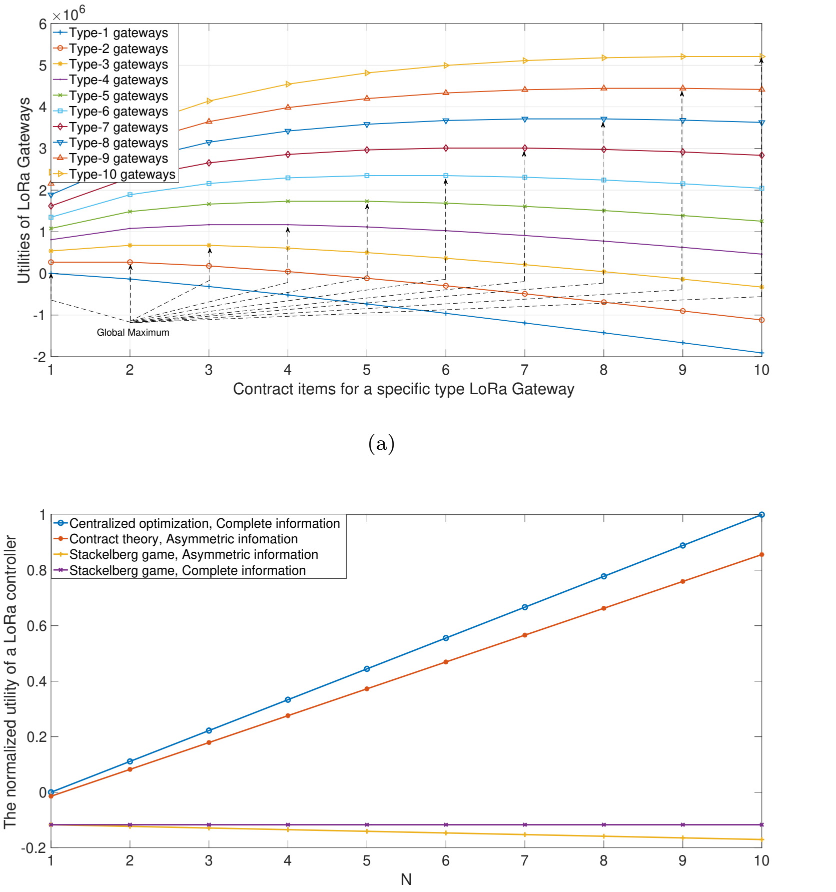
$$U_c = \sum_{j=1}^{N} (\mu(B_j) - \omega T_j), \tag{25}$$

where $B_j = \frac{\lambda_j - \phi}{\lambda_j \phi}$ and $T_j = \frac{\lambda_j - \phi}{\phi}$.

### *4.1.4. Comparison between the Contract-theoretic Incentive Mechanism and Other Benchmarks*

We conduct simulations in Matlab. Major parameters used in the simulations are given in Table 1, and the settings of $SF$, $BW$, and $CR$ refer to [33]. We consider the default parameter settings that 100 gateways participate in the transmitting tasks published by a single controller. The gateways are classified into 10 types as $\boldsymbol{\theta}$= $\{1, \dots, 10\}$, leading to the probability of a gateway belonging to a particular type being 0.1.

Fig. 3(a) shows the utilities of gateways from type-$\theta_1$ to type-$\theta_{10}$. It can be observed from each of the global maximum that a gateway can only obtain the



(a)

(b)

Figure 3: (Left) The utilities of LoRa Gateways under different contract items (different types). (Right) The utilities of LoRa Controllers under a different total number of LoRa Gateways.

Table 1: Parameter setting in the simulation

| **Parameter** | **Setting** |
|---|---|
| Spreading factor ($SF$) | $[7, 8, 9, 10, 11, 12]$ |
| Bandwidth in kHz ($BW$) | $[125, 250, 500]$ |
| Code rate ($CR$) | $[\frac{4}{5}, \frac{4}{6}, \frac{4}{7}, \frac{4}{8}]$ |
| The number of LoRa Gateways ($N$) | 100 |
| The number of types ($F$) | 10 |
| Task Period in second ($\mathbb{T}_{max}$) | 10 |
| The unit price for each LoRa Gateway in a Stackelberg game ($\lambda$) | $10^7$ |
| The pre-defined weighted parameter for $\mu(\bullet)$ ($\sigma$) | 1 |
| The pre-defined weighted parameter for contract incentive $T$ ($\omega$) | 10 |
| The pre-defined weighted parameter for the significant throughput $B$ ($\phi$) | 10 |

non-negative maximum utility (validating the IR constraints of the contract) when it chooses the option exactly designed for its type (validating the IC constraints of the contract).

We compare the utility of the controllers in the proposed contract-theoretic incentive mechanism with the existing benchmark mechanisms under symmetric and asymmetric information (a Stackelberg-based model considered in [15]). As shown in Fig. 3(b), a controller can obtain a linearly increasing utility as the number of gateways $N$ increases in the proposed contract-theoretic incentive mechanism. Moreover, the proposed mechanism significantly (red line) promotes the utility to approach the upper-bound (blue line) by relieving the information asymmetry. This results in a significant improvement even against the Stackelberg game with symmetric information (purple line). The utility of a controller in purple can be hardly improved with an increasing value of $N$. On the other hand, The information asymmetry worsens the utility of controllers in the Stackelberg model with asymmetric information (orange line) where the orange line decreases significantly with the growth of $N$. This is because the more gateways are involved in the task, the more diverse the combinations of the gateways' types are. This differs from the preconcerted types in the proposed contract-theoretic model. An excessive number of types worsen the information asymmetry as the frequent changes can make the type adjustment fails. To solve this problem, preconcerted types in the proposed model can be useful when a controller manages to obtain the information of gateways to relieve the information asymmetry. The limited options for gateways to restrict the number of types avoid needing to concern the type adjustment due to the changes of gateways (as preconcerted).

### *4.2. Evaluation of the new Dual-Chain System with the proposed Flow Control and Incentive Mechanism*

We carry out Monte-Carlo simulations in Python-3.8 to evaluate the impact on a LoRa network from the proposed flow control and incentive mechanism in a systematic view. We hereby define the following three terms:

**(Un)weighted Scoring:** Excessive duplicated data severely degrades the throughput of *Data DAG* [37]. Specifically, transmitting a message to $n$ different congested gateways results in the throughput of the *Data DAG* being reduced by $n$. In order to mitigate the loss, weighted scoring is introduced to assess the overall system by taking into account an expected loss rate associated with the overlapping coverage area among congested gateways, i.e., the total profit of the whole system. Accordingly, unweighted scoring is also introduced to represent the area utilization of a specific region managed by a single controller, i.e., the total coverage ratio.

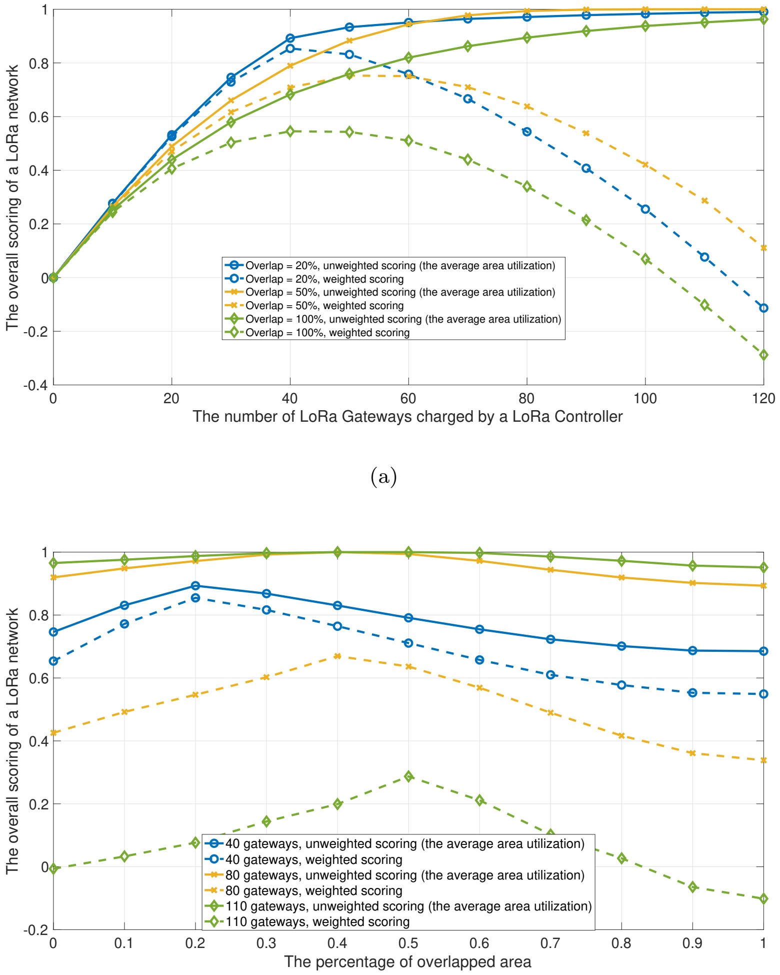
**Expected Loss Rate:** The expected loss rate is the average loss rate applied to the gateways sending duplicated data to the controller. It is used to assess the impact of the proposed incentive mechanism applied to the overlapping coverage area in order to mitigate the throughput loss on the *Data DAG*. The overlapping coverage area is multiplied by the loss rate, which reflects a rapid decline in the overall weighted scoring because of the possible overlapping coverage areas of more than two gateways. A zero loss rate indicates that the negative impacts of the overlapping coverage area are not considered[4]. A lower loss rate implies a larger impact of the incentive mechanism applied to the overlapping coverage area.

**Ideal Overlapped Proportion:** The maximum overlapping coverage area which can be accepted by a gateway being deployed in the case where there is enough leftover space within the region. Any owners will prefer to deploy their own gateways apart from an existing one in order to achieve the ideal overlapped proportion, until the gateways cannot be situated in such a position anymore due to the insufficient space. Gateways distributed in an overlapping coverage area have to comply with the flow control protocol conducted by the corresponding controller. A gateway tends to carefully maintain an overlapping coverage area with others such that it can enjoy more exclusive use of the significant throughput while still having a chance to compete with others. As such, traffic congestion can be relieved without having an excessive amount of duplicated data.

We consider an expected loss rate = 80% in the simulations. Figs. 4(a) and 4(b) show the trajectory of the overall (un)weighted scoring with the increasing number of gateways and the percentage of overlapping coverage area in a specific region, respectively. Every single scenario determined by the overlapping coverage area or the number of gateways is illustrated in a specific color, with the unweighted scoring and weighted scoring plotted as solid curve and dotted curve, respectively.

It is concluded in Fig. 4(a) that the unweighted scoring of overlap of 20% (the solid blue curve) reaches 90% area utilization the fastest at around 40

[4] An expected loss rate = 1 indeed results in the overlook of negative effects of the overlapping coverage area. This is because, intuitively, the existing LoRa network designs take the data redundancy into account while rarely considering the throughput loss in a Blockchain-based LoRa network.



(a)

(b)

Figure 4: (Left) The trajectory of the overall scoring along with the increasing number of LoRa Gateways changed by a LoRa Controller. (Right) The trajectory of the overall scoring along with the increasing percentage of the overlapping coverage area among gateways.

gateways while its weighted scoring (the dotted blue curve) remains the highest. In contrast, overlap of 100% (the solid green curve) implies the system takes no consideration about the negative effects of the overlapping coverage area. It reaches 90% area utilization at around 80 gateways and incurs the fastest decline of the weighted scoring (the dotted green curve). The unweighted scoring of overlap of 50% (the solid orange curve) reaches 90% area utilization by having only around 10 gateways more (at around 50 gateways) with the longest tail (the dotted orange curve). The longest tail implies that the scenario can afford the most number of gateways within the region while having the slowest decline of the weighted scoring (in other words, the total profit of the whole system). This encourages an energetic ecosystem where more users are willing to participate.

Fig. 4(b) reveals the same concept from another perspective. The scenario with 40 gateways exhibits the peaks on both the unweighted scoring (the solid blue curve) and weighted scoring (the dotted blue curve) at overlap of 20%. As the percentage of overlapping coverage area increases to around overlap of 50%, the peaks gradually shift to the point with overlap of 50% wherein there is an increasing number of participating gateways (from the orange curve to the green curve).

Along with the findings in Fig. 4, we can conclude that, introducing our new flow control and incentive mechanism can satisfy a Dual-Chain-based LoRa network requesting high throughput and flexibility. A smaller upper-bound of the overlapping coverage area which needs to comply with the flow control (a more strict flow control) can result in a fast convergence to high area utilization while the total profit of the whole system remains high. On the other hand, if the goal is to encourage more gateways to participate, an upper-bound of the overlapping coverage area approaching 50% can balance the area utilization and the system profit very well.

### *4.3. Security Analysis*

The security analysis focuses on the dual-chain-based structure and the proposed new PoTO protocol.

### *4.3.1. Dual-Chain Structure: Why Splitting Functions?*

We split the functions between the *Data DAG* and *ID Chain*, respectively. The *Data DAG* is only responsible for the data storage, while the *ID Chain* is responsible for contract operations and payment functions. The proposed dual-chain strategy is different from those IOTA-like DAG-based structures where the payment functions is inherently supported. The strategy is particularly important in the context of LoRa. The reason is that a malicious controller may collude with all its gateways by sharing gateways' private keys, or simply generate a set of private keys and pretend to manage some gateways. As such, the controller can bypass the preconcerted incentive rule of the contract theory, and maximize its average block rate in the *Data DAG*, compared to other honest controllers which only initiate the transmission after collecting some data. As a result, the attacker can broadcast blocks which contain double-spending payment at the highest rate to induce the growth of *Data DAG* in its favour, thus breach the transaction order finality and compromise the balance system. It is thus significant to introduce a dual-chain structure that splits the payment function (whose physical meaning matters) out from the *Data DAG* and let the *ID Chain* handle the payment.

### *4.3.2. Dual-Chain Structure: The Performance Gap*

The performance gap between a scalable DAG-based structure and an non-scalable chain-based structure may lead to network bottlenecks, data inconsistency, and corruption. Our proposed system enable the consistency between the *Data DAG* and *ID Chain*, even with an increasing number of end-devices and gateways. This can be achieved due to:

- **The scarcity of smart contract operations on the *ID Chain*:** Any smart contract operations on an *ID Chain* related to the devices registration/monitoring and contract initialization/status records are conducted much less frequently than publishing a single contract-theoretic task on an *Data DAG*. This is because these operations mostly send small data to update the status on smart contracts, and closely depend on the type-classification which can be conducted once and serve multiple tasks. Likewise, the randomness generated on the *ID Chain* can also be reused conservatively for multiple tasks during the self-driven flow control process.

- **The limited amounts of data transmitted on the *Data DAG*:** The block rate of *Data DAG* in LoRa networks is restrained by the task period, the data size of a task, and the number of gateways involved in a task.
  - **Limited task period:** The period of any smart contract operations related to incentive payment and balance records correspond to the unit period of a published task, as the functionality is in the charge of the *ID Chain*. Such period is restrained by an upper-bounded frequency of publishing new blocks to the *Data DAG* based on the anti-spam features, i.e., PoTO protocol.
  - **Limited data size:** The data size uploaded by a gateway during a task is restrained by the physical channel resource according to Equations (1) and (2), including spreading factor, bandwidth, code rate, duty cycle, etc.
  - **Limited number of gateways:** It is found in Figure 4 that the number of self-deployed gateways in a region tends to fall into a certain range and maximize the utility of the LoRa network, considering that the (un)weighted scoring takes effect for the penalty of the uploaded data size.
- **The advanced technologies implemented to improve the *ID Chain*:** The performance gap can be eliminated by implementing advanced technologies to improve the performance of the *ID Chain*, such as the HotStuff consensus algorithm that enables high transaction rate among large communities, and the sharding technology that enables the horizontal scalability. Moreover, requesting a higher security level on the *Data DAG* needs each node to download as deep *Data DAG* as possible for more secure validation. This makes the *Data DAG* no better than the improved

*ID Chain*, and stops the *Data DAG* from being scaled out. One feasible solution is to implement the sharding technology to both the *Data DAG* and *ID Chain*.

### *4.3.3. PoTO Protocol: The Improved Spam Protection*

The proposed system inherently enables the spam protection which, otherwise, would have to be achieved by a typical PoW process in IOTA. The PoW used in IOTA is not a typical Proof-of-X (PoX)-based consensus algorithm [38]. It is a comparably simple computational operation which differs from the expensive PoW conducted in many miner-based Blockchains. Instead of racing for the winner of each consensus round, the defense to DDoS attacks and spam protection are the key goals of the PoW in IOTA. Our proposed PoTO protocol can deliver the same protection without need of additional computational operations required by the PoW used in IOTA, as clarified below from the perspectives of system restriction and attack motivation.

From the perspective of system restriction,

- **Limited data source:** Each LoRa Gateway needs to be registered on the *ID Chain* and generate the unique private key for digital signature, prior to granting permission to participate in the LoRa networks. The controllers can mutually verify each of the identities of transactions in a DAG block to ensure the transactions are indeed sourced from a certain and finite gateway set.

- **Limited data size and number of gateways:** As suggested in Section 4.3.2, the data size uploaded by a gateway during a task is restrained by the physical channel resource according to Equations (1) and (2). Also, the number of self-deployed gateways in a region falls into a certain range to maximize the entire utility.

In our system, the controllers incentivize the self-deployment of gateways. The controllers and gateways maximize the utilities, i.e., Equations (5) and (7). A controller would not be able to spam the network unless it colludes with all its gateways. In the worst-case scenario where the collusion happens, the system restriction can extend a period of time before a task is finalized, thus significantly reducing the risk of DDoS attacks with a compulsory minimum size of payloads. This leads to a controllable growth rate of the DAG , since the block rate depends on the data source, data size, and the number of gateways.

From the perspective of attack motivation, as suggested in Section 2, the function of incentive payment is conducted on the *ID Chain* in our proposed dual-chain system. This leads to independence among controllers, and the controllers are only responsible for the validity of on-chain LoRa data stored on the *Data DAG*, and not for the validity of the metadata. Each individual controller aims to enhance the quality of local LoRa business. Spending time and consuming communication (downloading/uploading) and computation (verifying/smart contract operations) resources to upload local corrupted LoRa data does not affect the interests of others, and reaps no profits for itself. Therefore, for LoRa Controllers, the communication and computation overhead is enough to be the amount of "work" done during the data collection and validation process, i.e., PoTO, without need of additional computation operations.

## 5. Related Work

Blockchain-based systems have been proposed to improve the data integrity verification of current ISs [39, 40, 41, 42, 43]. The authors of [39] propose an efficient integrity check scheme for IoT-IS with no need of trusted third parties, based on Lifted EC-ElGamal cryptosystem and bilinear pairing. The authors of [40] propose a public auditing scheme for data verification in cloud storage which is only accessible to data owners and cloud service providers. The authors of [41] develop a decentralized application, namely, EtherTwin, for digital twins data sharing in an industrial context. [42] proposes a task scheduling technique for fog computing to guarantee the privacy of user information. [43] provides an overall latency analysis of hyperledger fabric Blockchain networks. However, these studies either directly utilize existing Blockchain technologies to improve the data security and privacy in IoT-IS without optimizations of the Blockchains[39, 40, 41, 42]; or only provide a general analytical model of Blockchains, which could hardly capture the specialty of IS [43]. In practice, both the data communication protocol in IoT networks and the design of Blockchain technologies have non-negligible influence on data security and transaction performance. It is important to design jointly the IoT communication protocol and Blockchains. In this paper, we optimize the design of Blockchains specially tailored for LoRa-IS.

Recent studies have investigated the integration between the Blockchain and LoRa technologies [44, 45, 46, 47] to enhance the security of Blockchain-based LoRa-IS. They integrate an existing traditional Blockchain platform (e.g., Ethereum) with a LoRa network, aiming to take advantage of smart contracts for data storage. [44] introduces an Ethereum Blockchain for data storage and access, where either end-devices or gateways are in charge of the block generator. [45] reveals the end-devices incur large overhead with the design in [44], and thus introduces a separate Ethereum Blockchain in which multiple agent nodes can provide data storage and access services for either gateways and controllers. [46, 47] employ an Ethereum Blockchain as a decentralized database providing data storage and access services for all nodes in the network. However, an Ethereum Blockchain or other traditional Blockchain technologies have been revealed to suffer from the vulnerable scalability in the context of LPWAN [19], and the one-device-to-many-gateway property of LoRa networks with ALOHA access even compromises the scalability. Again, an incentive mechanism needed to motivate the LoRa network deployment still lacks a reliable Blockchain-based solution. None of the existing technologies take advantage of both technologies, and are able to deliver a Blockchain-based solution to LoRa-IS where the Blockchain and LoRa technologies can complement each other to address the above issues. By using our new Dual-Chain-based LoRa-IS, the scalability issue (throughput loss and huge storage) of Blockchains can be relieved without modifying the original LoRa transmission protocol, and the self-motivation to deploy the LoRa networks can be fairly incentivized and secured by Blockchains.

The contract theory is first formalized in [16] to relieve negative effects of information asymmetry, and subsequently introduced in a number of areas, such as delayed traffic offloading [48], device-to-device communications [49], wireless energy harvesting [15], fog computing [36], and vehicle-to-vehicle communications [35]. However, none of the above studies combine with Blockchain to prevent malicious task publishers nor leverage a feasible global cross-validation protocol. To the best of our knowledge, this is the first paper introducing the contract theory for data collection in LoRa-IS under information asymmetry, and also secure the process by combining with a Dual-Blockchain system in an efficient and scalable way.

## 6. Conclusions and Future Work

In this article, we proposed a new Dual-Chain-based LoRa-IS, where the state-of-the-art contract-theoretic incentive mechanism is applied to motivate self-driven deployment, and data storage service can be secured by a decentralized global cross-validation with the tamper-resistance of Blockchain. The contract-theoretic incentive mechanism provides a healthy and fair credit system by effectively relieving the impact of information asymmetry and maximizing the profits of both controllers and gateways. Being part of the proposed incentive mechanism, the new self-driven flow control protocol mitigates the traffic congestion and throughput loss of the Blockchain by avoiding duplicated data. As a result, each gateway would be preferably installed at the location where the entire system throughput can be maximized, while reducing the data replication without the need of modifying existing LoRaWAN protocols.

In the future, we will apply our system in a practical LoRa-IS and evaluate its performance. We will also extend our system to more generalized LPWAN networks, and address computation, communication, and storage complexity by using sharding technologies [28].

## Acknowledgments

This project was partially supported by funding from Food Agility CRC Ltd, funded under the Commonwealth Government CRC Program. The CRC Program supports industry-led collaborations between industry, researchers and the community.

## References

## References

[1] D. Berdik, S. Otoum, N. Schmidt, D. Porter, Y. Jararweh, A survey on blockchain for information systems management and security, Information Processing  Management 58 (1) (2021) 102397. doi:https://doi.org/10.1016/j.ipm.2020.102397.
URL http://www.sciencedirect.com/science/article/pii/S030645732030892X

[2] H. Ko, J. H. Kim, K. An, L. Mesicek, G. Marreiros, S. B. Pan, P. Kim, Smart home energy strategy based on human behaviour patterns for transformative computing, Information Processing  Management 57 (5) (2020) 102256. doi:https://doi.org/10.1016/j.ipm.2020.102256.
URL http://www.sciencedirect.com/science/article/pii/S0306457320300662

[3] C. Esposito, O. Tamburis, X. Su, C. Choi, Robust decentralised trust management for the internet of things by using game theory, Information Processing  Management 57 (6) (2020) 102308. doi:https://doi.org/10.1016/j.ipm.2020.102308.
URL http://www.sciencedirect.com/science/article/pii/S0306457320308037

[4] C. Oham, R. A. Michelin, R. Jurdak, S. S. Kanhere, S. Jha, B-ferl: Blockchain based framework for securing smart vehicles, Information Processing  Management 58 (1) (2021) 102426.

doi:https://doi.org/10.1016/j.ipm.2020.102426.
URL http://www.sciencedirect.com/science/article/pii/S0306457320309183

[5] R. S. Sinha, Y. Wei, S.-H. Hwang, A survey on lpwa technology: Lora and nb-iot, ICT Express 3 (1) (2017) 14 – 21. doi:https://doi.org/10.1016/j.icte.2017.03.004.
URL http://www.sciencedirect.com/science/article/pii/S2405959517300061

[6] D. Poluektov, M. Polovov, P. Kharin, M. Stusek, K. Zeman, P. Masek, I. Gudkova, J. Hosek, K. Samouylov, On the performance of lorawan in smart city: End-device design and communication coverage, in: V. M. Vishnevskiy, K. E. Samouylov, D. V. Kozyrev (Eds.), Distributed Computer and Communication Networks, Springer International Publishing, Cham, 2019, pp. 15–29.

[7] L. Alliance, LoRaWAN Specification v1.0, accessed on 16.04.2020 (2015).
URL https://lora-alliance.org/sites/default/files/2018-05/2015_-_lorawan_specification_1r0_611_1.pdf

[8] A. Bhutani, P. Wadhwani, Global LPWAN Market Size worth over $65 Bn by 2025, accessed on 16.04.2020 (2019).
URL https://www.gminsights.com/pressrelease/lpwan-market

[9] R. Merritt, LoRa Taps New Chips, Smart Homes, accessed on 16.04.2020 (2018).
URL https://www.eetimes.com/lora-taps-new-chips-smart-homes/

[10] I. Analytics, LPWAN Market Report 2018-2023, accessed on 16.04.2020 (2018).
URL https://iot-analytics.com/product/lpwan-market-report-2018-2023/

[11] S. Delbruel, N. Small, E. Aras, J. Oostvogels, D. Hughes, Tackling contention through cooperation: A distributed federation in lorawan space, in: Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks, Junction Publishing, 2020, pp. 13–24.

[12] M. N. Ochoa, A. Guizar, M. Maman, A. Duda, Toward a self-deployment of lora networks: Link and topology adaptation, in: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2019, pp. 1–7.

[13] E. Yang, The challenges of implementing LoRa and LoRaWAN in industries worldwide, accessed on 15.05.2020 (Oct. 2019).
URL `https://www.asmag.com/showpost/30700.aspx`

[14] D. Ismail, M. Rahman, A. Saifullah, Low-power wide-area networks: Opportunities, challenges, and directions, in: Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking, Workshops ICDCN '18, Association for Computing Machinery, New York, NY, USA, 2018. `doi:10.1145/3170521.3170529`.
URL `https://doi.org/10.1145/3170521.3170529`

[15] Z. Hou, H. Chen, Y. Li, B. Vucetic, Incentive mechanism design for wireless energy harvesting-based internet of things, IEEE Internet of Things Journal 5 (4) (2018) 2620–2632.

[16] P. Bolton, M. Dewatripont, et al., Contract theory, MIT press, 2005.

[17] O. Novo, Blockchain meets iot: An architecture for scalable access management in iot, IEEE Internet of Things Journal 5 (2) (2018) 1184–1195.

[18] H. Dai, Z. Zheng, Y. Zhang, Blockchain for internet of things: A survey, IEEE Internet of Things Journal 6 (5) (2019) 8076–8094.

[19] M. Salimitari, M. Chatterjee, Y. Fallah, A survey on consensus methods in blockchain for resource-constrained iot networks (Apr 2020).

doi:10.36227/techrxiv.12152142.v1.
URL https://www.techrxiv.org/articles/A_Survey_on_Consensus_Methods_in_Blockchain_for_Resource-constrained_IoT_Networks/12152142/1

[20] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, K. Zheng, Survey on blockchain for internet of things, Computer Communications 136 (2019) 10 – 29. doi:https://doi.org/10.1016/j.comcom.2019.01.006.
URL http://www.sciencedirect.com/science/article/pii/S0140366418306881

[21] I. Factory, Best LORAWAN Network Servers, accessed on 16.04.2020 (2020).
URL https://iotfactory.eu/products/software-platform/best-lorawan-network-servers/

[22] P. Gotthard, Compact server for private LoRaWAN networks, accessed on 16.04.2020 (2020).
URL https://github.com/gotthardp/lorawan-server

[23] O. Brocaar, ChirpStack Network Server, accessed on 16.04.2020 (2020).
URL https://github.com/brocaar/chirpstack-network-server

[24] S. Popov, The tangle, cit. on (2016) 131.

[25] M. Zichichi, S. Ferretti, G. D'angelo, A framework based on distributed ledger technologies for data management and services in intelligent transportation systems, IEEE Access 8 (2020) 100384–100402. doi:10.1109/ACCESS.2020.2998012.

[26] D. M, N. B. Biradar, Iota-next generation block chain, International Journal of Engineering and Computer Science 7 (04) (2018) 23823–23826.
URL http://103.53.42.157/index.php/ijecs/article/view/4007

[27] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, I. Abraham, Hotstuff: Bft consensus with linearity and responsiveness, in: Proceedings of the

2019 ACM Symposium on Principles of Distributed Computing, PODC '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 347–356. doi:10.1145/3293611.3331591.
URL https://doi.org/10.1145/3293611.3331591

[28] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, R. P. Liu, Survey: Sharding in blockchains, IEEE Access 8 (2020) 14155–14181.

[29] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, T. Watteyne, Understanding the limits of lorawan, IEEE Communications Magazine 55 (9) (2017) 34–40.

[30] A. Lavric, V. Popa, Internet of things and lora$^{\text{TM}}$ low-power wide-area networks: A survey, in: 2017 International Symposium on Signals, Circuits and Systems (ISSCS), 2017, pp. 1–5.

[31] M. Saelens, J. Hoebeke, A. Shahid, E. De Poorter, Impact of eu duty cycle and transmission power limitations for sub-ghz lpwan srds: an overview and future challenges, EURASIP Journal on Wireless Communications and Networking 2019 (1) (2019) 219.

[32] B. Baek, Iota: A cryptographic perspective, 2019.

[33] A. Augustin, J. Yi, T. Clausen, W. M. Townsley, A study of lora: Long range amp; low power networks for the internet of things, Sensors 16 (9). doi:10.3390/s16091466.
URL https://www.mdpi.com/1424-8220/16/9/1466

[34] Randao: Verifiable Random Number Generation (2017).
URL https://www.randao.org/whitepaper/Randao_v0.85_en.pdf

[35] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, J. Zhao, Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory, IEEE Transactions on Vehicular Technology 68 (3) (2019) 2906–2920. doi:10.1109/TVT.2019.2894944.

[36] M. Zeng, Y. Li, K. Zhang, M. Waqas, D. Jin, Incentive mechanism design for computation offloading in heterogeneous fog computing: A contract-based approach, in: 2018 IEEE International Conference on Communications (ICC), 2018, pp. 1–6. doi:10.1109/ICC.2018.8422684.

[37] Y. Lewenberg, Y. Sompolinsky, A. Zohar, Inclusive block chain protocols, in: R. Böhme, T. Okamoto (Eds.), Financial Cryptography and Data Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 528–547.

[38] G. Yu, X. Zha, X. Wang, W. Ni, K. Yu, J. A. Zhang, R. P. Liu, A unified analytical model for proof-of-x schemes, Computers  Security 96 (2020) 101934. doi:https://doi.org/10.1016/j.cose.2020.101934.
URL http://www.sciencedirect.com/science/article/pii/S0167404820302108

[39] Q. Zhao, S. Chen, Z. Liu, T. Baker, Y. Zhang, Blockchain-based privacy-preserving remote data integrity checking scheme for iot information systems, Information Processing  Management 57 (6) (2020) 102355. doi:https://doi.org/10.1016/j.ipm.2020.102355.
URL http://www.sciencedirect.com/science/article/pii/S0306457320308505

[40] J. Li, J. Wu, G. Jiang, T. Srikanthan, Blockchain-based public auditing for big data in cloud storage, Information Processing  Management 57 (6) (2020) 102382. doi:https://doi.org/10.1016/j.ipm.2020.102382.
URL http://www.sciencedirect.com/science/article/pii/S0306457320308773

[41] B. Putz, M. Dietz, P. Empl, G. Pernul, Ethertwin: Blockchain-based secure digital twin information management, Information Processing  Management 58 (1) (2021) 102425. doi:https://doi.org/10.1016/j.ipm.2020.102425.
URL http://www.sciencedirect.com/science/article/pii/S0306457320309195

[42] H. Baniata, A. Anaqreh, A. Kertesz, Pf-bts: A privacy-aware fog-enhanced blockchain-assisted task scheduling, Information Processing  Management 58 (1) (2021) 102393. doi:https://doi.org/10.1016/j.ipm.2020.102393. URL http://www.sciencedirect.com/science/article/pii/S0306457320308888

[43] X. Xu, G. Sun, L. Luo, H. Cao, H. Yu, A. V. Vasilakos, Latency performance modeling and analysis for hyperledger fabric blockchain network, Information Processing  Management 58 (1) (2021) 102436. doi:https://doi.org/10.1016/j.ipm.2020.102436.
URL http://www.sciencedirect.com/science/article/pii/S0306457320309298

[44] K. R. Ozyilmaz, A. Yurdakul, 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.

[45] S. M. Danish, M. Lestas, W. Asif, H. K. Qureshi, M. Rajarajan, A lightweight blockchain based two factor authentication mechanism for lorawan join procedure, in: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), 2019, pp. 1–6.

[46] S. R. Niya, S. S. Jha, T. Bocek, B. Stiller, Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and lorawan, in: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1–4.

[47] J. Lin, Z. Shen, C. Miao, Using blockchain technology to build trust in sharing lorawan iot, in: Proceedings of the 2nd International Conference on Crowd Science and Engineering, ICCSE'17, Association for Computing Machinery, New York, NY, USA, 2017, p. 38–43. doi:10.1145/3126973.3126980.
URL https://doi.org/10.1145/3126973.3126980

[48] Y. Li, J. Zhang, X. Gan, L. Fu, H. Yu, X. Wang, A contract-based incentive mechanism for delayed traffic offloading in cellular networks, IEEE Transactions on Wireless Communications 15 (8) (2016) 5314–5327.

[49] Y. Chen, S. He, F. Hou, Z. Shi, J. Chen, Promoting device-to-device communication in cellular networks by contract-based incentive mechanisms, IEEE Network 31 (3) (2017) 14–20.

## Appendix A. Proof the utilities satisfy the Spence-Mirrlees property

In this part, we will prove the utilities satisfy the Spence-Mirrlees property. Based on (6), we have $U_f = \theta_f \nu(T_f) - \phi B_f, \forall f \in \{1, \ldots, F\}$ with $\frac{\partial \nu}{\partial T_f} > 0$, $\frac{\partial^2 \nu}{\partial T_f^2} < 0$, $\frac{\partial \nu}{\partial \theta_f} > 0$, and $\nu(0) = 0$.

$$\frac{\partial U}{\partial T} = \frac{\partial U}{\partial \nu} \times \frac{\partial \nu}{\partial T} > 0, \tag{A.1}$$

as $\frac{\partial U}{\partial \nu} = \theta_f > 0$ and $\frac{\partial U}{\partial T} > 0$ as $\frac{\partial \nu}{\partial T_f} > 0$. On the other hand, as $\frac{\partial U}{\partial B} = -\phi$, we can attain

$$-\frac{\partial U/\partial T}{\partial U/\partial B} = -\frac{(\theta \frac{\partial \nu}{\partial T})}{-\phi} = \frac{\theta}{\phi} \times \frac{\partial \nu}{\partial T} > 0 \tag{A.2}$$

Thus, we have

$$\frac{\partial}{\partial \theta}[-\frac{\partial U/\partial T}{\partial U/\partial B}] = \frac{\partial}{\partial \theta}[\frac{\theta \partial \nu}{\phi \partial T}] = \frac{\partial \nu}{\phi \partial T} > 0. \tag{A.3}$$

The utilities satisfy the Spence-Mirrlees property.

## Appendix B. Proof of Lemma 1

In this part, we will prove the *Lemma 1*. According to the IC constraints (*Definition 2*) of type-$\theta_f$ and type-$\theta_{f'}$ gateways, we have

$$\begin{cases} \theta_f \nu(T_f) - \phi B_f \geq \theta_f \nu(T_{f'}) - \phi B_{f'}, \\ \theta_{f'} \nu(T_{f'}) - \phi B_{f'} \geq \theta_{f'} \nu(T_f) - \phi B_f. \end{cases} \tag{B.1}$$

Also, we can have

$$\theta_f[\nu(T_f) - \nu(T_{f'})] \geq \phi(B_f - B_{f'}) \geq \theta]\nu(T_f) - \nu(T_{f'})]. \tag{B.2}$$

Given that $B_f \geq B_{f'}$ and $\phi \geq 0$, then $\phi(B_f - B_{f'}) \geq 0$ as $T_f \not< T_{f'}$ with the monotonic increasing function $\nu$. Thus, *Lemma 1* is proved.

## Appendix C. Proof of Lemma 2

In this part, we will prove the *Lemma 2*. According to IC constraint of (7), $\forall f \in \{2, \ldots, F\}$, and $\theta_1 < \theta_2 < \cdots < \theta_F$, we have

$$\theta_f \nu(T_f) - \phi B_f \geq \theta_f \nu(T_1) - \phi B_1 \geq \theta_1 \nu(T_1) - \phi B_1 \geq 0. \tag{C.1}$$

Eq. (C.1) implies that all the other $(F-1)$ IR constraint can be satisfied and relaxed to IR constraint of type-$\theta_1$ if the IR constraint of the gateway with type-$\theta_1$ is satisfied. Thus, *Lemma 2* is proved.

## Appendix D. Proof of Corollary 2

In this part, we will prove the *Corollary 2*. According to *Lemma 1*, we have

$$\theta_{f+1}\nu(T_f) - \phi B_f \geq \theta_{f+1}\nu(T_{f-1}) - \phi B_{f-1}, T_f \geq T_{f-1} \tag{D.1}$$

which in turn implies that

$$\theta_{f+1}\nu(T_{f+1}) - \phi B_{f+1} \geq \theta_{f+1}\nu(T_{f-1}) - \phi B_{f-1}, \tag{D.2}$$

where $\forall f \in \{2, \ldots, F-1\}$. This indicates that the DIC for type-$\theta_{f+1}$ and contract $(T_{f-1}, B_{f-1})$ is satisfied. In other words, gateways with type-$\theta_{f+1}$ are incentivilized to accept $(T_{f+1}, B_{f+1})$ the most. Thus, we can extend this result down to type-$\theta_1$ to prove all DICs hold, i.e.,

$$\begin{aligned} \theta_{f+1}\nu(T_{f+1}) - \phi B_{f+1} &\geq \theta_{f+1}\nu(T_{f-1}) - \phi B_{f-1} \geq \cdots \\ &\geq \theta_1\nu(T_1) - \phi B_1, \forall f \in \{2, \ldots, F-1\}. \end{aligned} \tag{D.3}$$

Thus, *Corollary 2* is proved.

## Appendix E. Proof of Lemma 3

In this part, we will prove the *Lemma 3*. If $\theta_f\nu(T_f) - \phi B_f > \theta_f\nu(T_{f-1}) - \phi B_{f-1}$ for some gateways with type-$\theta_f$, the gateways will try to maximize the utility by raising $B_{f'}$ for all $f' > f$ so that $\theta_f\nu(T_f) - \phi B_f = \theta_f\nu(T_{f-1}) - \phi B_{f-1}$ holds. Thus, the optimum implies that all LDICs are binding and hence *Lemma 3* is proved.

## Appendix F. Proof of Equation (17)

In this part, we will prove the transformation of (17).

$$\sum_{f=1}^{F} p_f[\frac{\phi B_1}{\theta_1} + (\frac{\phi B_k}{\theta_k} - \frac{\phi B_{k-1}}{\theta_k})] \tag{F.1}$$

$$
\begin{aligned}
&= p_1(\frac{\phi B_1}{\theta_1})+ \\
&\quad p_2(\frac{\phi B_1}{\theta_1} + \frac{\phi B_2}{\theta_2} - \frac{\phi B_1}{\theta_2})+ \\
&\quad p_3(\frac{\phi B_1}{\theta_1} + \frac{\phi B_2}{\theta_2} - \frac{\phi B_1}{\theta_2} + \frac{\phi B_3}{\theta_3} - \frac{\phi B_2}{\theta_2})+ \\
&\quad \ldots \\
&\quad + p_F(\frac{\phi B_1}{\theta_1} + \sum_{i=2}^{F}(\frac{\phi B_i}{\theta_i} - \frac{\phi B_{i-1}}{\theta_i})) \\
&= (\sum_{f=1}^{F} p_f)(\frac{\phi B_1}{\theta_1}) + (\sum_{f=2}^{F} p_f)(\frac{\phi B_2}{\theta_2} - \frac{\phi B_1}{\theta_2})+ \\
&\quad (\sum_{f=3}^{F} p_f)(\frac{\phi B_3}{\theta_3} - \frac{\phi B_2}{\theta_3}) + \cdots + p_F(\frac{\phi B_F}{\theta_F} - \frac{\phi B_{F-1}}{\theta_F}) \\
&= p_1\frac{\phi B_1}{\theta_1} \\
&+ p_2\frac{\phi B_1}{\theta_1} + p_2\frac{\phi B_2}{\theta_2} - p_2\frac{\phi B_1}{\theta_2} \\
&+ p_3\frac{\phi B_1}{\theta_1} + p_3\frac{\phi B_2}{\theta_2} - p_3\frac{\phi B_1}{\theta_2} + p_3\frac{\phi B_3}{\theta_3} - p_3\frac{\phi B_2}{\theta_3} \\
&\quad \ldots \\
&+ p_F\frac{\phi B_1}{\theta_1} + p_F\frac{\phi B_2}{\theta_2} - p_F\frac{\phi B_1}{\theta_2} + \cdots + p_F\frac{\phi B_F}{\theta_F} - p_F\frac{\phi B_{F-1}}{\theta_F} \\
&= \sum_{f=1}^{F} p_f\frac{\phi B_1}{\theta_1} + \sum_{f=2}^{F} p_f\frac{\phi B_2}{\theta_2} + \sum_{f=3}^{F} p_f\frac{\phi B_3}{\theta_3} + \cdots + p_F\frac{\phi B_F}{\theta_F} \\
&- \sum_{f=2}^{F} p_f\frac{\phi B_1}{\theta_2} - \sum_{f=3}^{F} p_f\frac{\phi B_2}{\theta_3} - \cdots - p_F\frac{\phi B_{F-1}}{\theta_F}
\end{aligned}
$$

$$
\begin{aligned}
&= (p_2 + p_3 + \cdots + p_F)(\frac{\phi B_1}{\theta_1} - \frac{\phi B_1}{\theta_2}) + p_1 \frac{\phi B_1}{\theta_1} + \cdots + \\
&\quad p_F(\frac{\phi B_{F-1}}{\theta_{F-1}} - \frac{\phi B_{F-1}}{\theta_F}) + p_{F-1} \frac{\phi B_{F-1}}{\theta_{F-1}} + p_F \frac{\phi B_F}{\theta_F} \\
&= \sum_{f=1}^{F} \pi_f B_f
\end{aligned}
$$

where

$$
\pi_f = \begin{cases} (\sum_{i=f+1}^{F} p_i)(\frac{\phi}{\theta_f} - \frac{\phi}{\theta_{f+1}}) + \frac{\phi p_f}{\theta_f}, 0 < f < F \\ \frac{\phi p_f}{\theta_f} & f = F. \end{cases}
$$

View publication stats