Chapter 4

An incentive mechanism-based Stackelberg game for scheduling of LoRa Spreading factors

4.1 Introduction

A LoRa network suffers from the interference problem when large number of LNs try to transmit the sensory data to the LG using SF. Interference problem can be reduced if each LN is allocated a time duration for its transmission and leaves the SF free for the remaining time during which it can be used by other LNs. Due to the interference problem, a LN needs to wait for accessing the allocated SF to transmit the data to the LG. In this chapter, we address the problem: *how long can a LN be scheduled on a given SF so that the utility of the LN can be maximized and minimizes the waiting time of the network?* We propose a game theory-based approach that allocates suitable SFs to the LNs and computes the optimal transmission time duration of sensory data on those SFs such that load among SFs is balanced. Load balancing on SFs minimizes the waiting time of the network. This time duration depends on the various factors such as the size of data transmission, processing cost, dissatisfaction cost (due to SF interference), and the price received from the applicants connected to the NS. LG gains the price from those applicants via NS for providing data. Further the LG pays some partial amount from the price gain to the LNs for forwarding the data. This is because, LNs consume their own resources for processing and forwarding data and they need to pay incentives to the EUs for collecting data. Such factors are modelled as monetary transfers in the proposed game. We minimize the waiting time of the LNs by scheduling them on the given SFs.

4.1.1 Motivation of this work

This section illustrates the limitations of the existing work which motivates for the proposed work.

• **Interactions among LNs** Authors in [15] designed a full MAC protocol enabling collision resolution. Authors in [16] proposed interference model for the LoRa modulation for different CRs and SFs. Different SFs were allocated based on random, fixed, and packet error rate schemes. Authors in [14] analyzed inter-SF and co-SF interference in the LoRa network. Based on the simulation results they also state that, for higher SFs, the effects of inter-SF interference is more noticeable. A theoretical analysis of the achievable LoRa throughput in the presence of both co-SF and inter-SF interference is presented in [18]. The main limitation in the existing work is that they do not consider interactions among LNs. Multiple access in wireless network using game theoretic approach is surveyed in papers [50, 51]. These survey shown that the game models are useful for designing distributed channel access mechanisms in wireless networks to achieve stable and efficient solutions.

• **Load balancing** Though there exist several work on the performance analysis of LoRa network [18, 52–55], most of them do not consider load balancing while allocating the SFs for improving the network performance. Authors in [18] proposed and compared various distance based SF allocation schemes that can improve network performance in the presence of co-SF interference. In paper [52], authors proposed resource allocation scheme for both power and SF with imperfect orthogonality in LoRa network. Authors in [53] formulated an optimization problem for maximizing the average packet transmission to propose a sub-optimal SF allocation. During SF allocation, determining time duration for transmission on shared channels is necessary to minimize drop rate. Optimal transmission policies are derived mathematically in [54]. Similarly, in [55] authors proposed a channel access control protocol that can improve network performance in terms of reliability, throughput, and energy consumption. Authors in paper [56], surveyed on a game theoretic approach for medium access protocols to access the shared channel. The authors in papers [57,58], proposed resource allocation scheme in wireless network using game theoretic approach. These work shown that game theory is an useful tool to improve the medium access protocol where, due to selfish behaviour, devices want to access resources based on their preferences which leads to imbalance the load on resources. The main drawback of such schemes is that they do not considered load imbalance on the SFs.

• **Scheduling of LNs** LoRa network faces the problem of scalability when they connect large number of LNs. Authors in [59] proposed a MAC layer protocol in which LG schedules LNs by dynamically specifying the allowed transmission powers and SFs. In paper [42] authors used matching game approach for allocating channels to the LNs such that they can maximize their utility. A nonconvex optimization problem for maximizing the network energy efficiency by considering user scheduling, SF assignment, and power allocation is presented in [60]. Authors in [39] proposed a synchronization and scheduling mechanism for LoRa networks consisting of class A devices whereas a new communication-planning mechanism in which the IoT gateways are used to plan the periodically repeated communication into a transmission schedule developed in paper [61]. Authors in [62] proposed a collision-free time-slotted scheduling approach where each device autonomously decides when to transmit a packet based on its unique identifier.

The main limitation in existing work is that they do not consider the scheduling of allocated time duration on SFs to minimize the waiting time of LNs.

4.1.2 Major contributions and overview of the solution

In this chapter, we propose a single-leader and multiple-followers Stackelberg game based approach to estimate the transmission time duration of LNs on feasible SFs as shown in Figure. 4.1. This computation is performed at LG because the LNs have limited processing and storage capacity. Once the LG received the request from LN, they computes the transmission time duration and schedule the LNs on the SFs. After computation, LG sends the control messages to the LNs for SF allocation along with the time slot of the LNs for data transmission. This chapter has following major contributions:

- We first propose a Hasse diagram based procedure for finding the subsets of the SFs for each LN, called as feasible subsets of SFs, on which they can successfully transfer the data to the LG.
- We next propose a follower game that models the interactions among LNs connected to the LG by using feasible subsets of SFs. We formulate a Stackelberg game among the LG (leader) and LNs (followers) to model the interactions between them. We propose an algorithm that chooses the pricing strategy of feasible subset of SFs to maximize the utility of LG. The LNs update their time duration based on the selected pricing strategy of LG to balance the load on different SFs. We finally select a feasible subset of SFs, called as optimal subset of SFs, which gives the maximum utility of the network and minimizes the network interference.
- To reduce waiting time of the network, we finally present an algorithm for scheduling the transmission of LNs on the SF. Such LNs are allocated the same SF.
- We used Network Simulator-3 [16] to demonstrate the impact of deployed LNs, game variables, and scheduling on the performance of the network. Network simulator is used

as it allows simulation of a large number of LoRa scenarios.

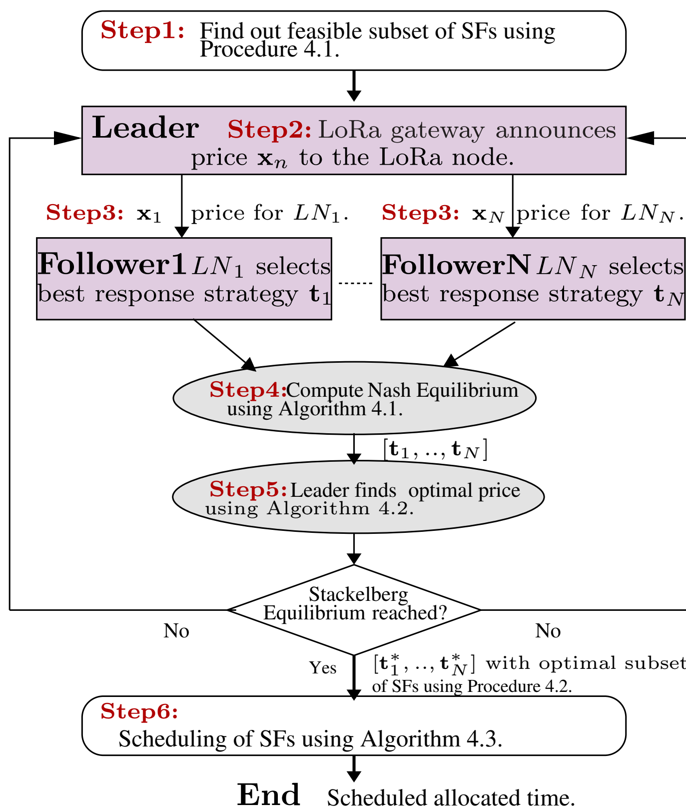


Figure 4.1: Illustration of the block diagram of single leader and multiple followers Stackelberg game.

The remaining of the chapter is organized as follows: next section presents a network scenario of smart home in urban area using a LoRa network. We use this scenario in this work. Section 4.3 presents a procedure for finding the feasible subsets of SFs. Section 4.4 presents the game analysis of follower and leader and finds the optimal subsets of SFs. Section 4.5 describes the scheduling of LNs for minimizing the waiting time of the network. The simulation results are presented in Section 4.6 and the chapter is concluded in Section 4.7.

4.2 Overview of the network scenario

A scenario of smart home in urban area using a LoRa network with different IoT sensors to collect sensory data is shown in Figure 4.2. It provides home automation to the users for convenience and increase resident satisfaction. Along with this, it also ensures efficient resource utilization and detect issues before they become catastrophic. For example, sensory data of people's thermostats help to detect the cause of power spikes, automatically switch off the lights whenever a user is not at home to save the electricity *etc.*. This scenario consists of multiple apartments in the building. Multiple LNs are installed in each apartment for collecting data from the sensors. These LNs broadcast data to the deployed LGs within the given range. However, multiple LGs in the network increase the possible combination of SFs for each LN and therefore increase the computational time complexities of the network. Such high complexities also increase the energy consumption of the network. To reduce such complexities, our proposed work considered one LG is bound to the LNs that fall within its range. If the number of LNs within the range of the LG increases, there is a need to increase the LGs also to mitigate the problem of overload. In that case, each LG is bound to a different group of LNs and receive packets from those bounded LNs only. The other LGs are separately scheduled the other unbound LNs. The LG forwards the collected data from the bounded LNs to the NS and finally to the applications.

Applicants, such as an electricity grid wants thermostats sensory data to know the power spikes and to forecast the electricity demands, use applications running on the collected data. The collection of personal data, in the home, exposes residents to the privacy and security risks. For example, smart light and smart door lock devices can be used as a pathway to enter into the home network to steal the data and find out more about the home user such as when the home is vacant and safe to rob. Due to such of these reasons, apartment owner does not want to share the data collected from the EUs willingly. In order to motivate EUs and to compensate their cost or risk, EUs usually be provided a reward as a monetary incentive.

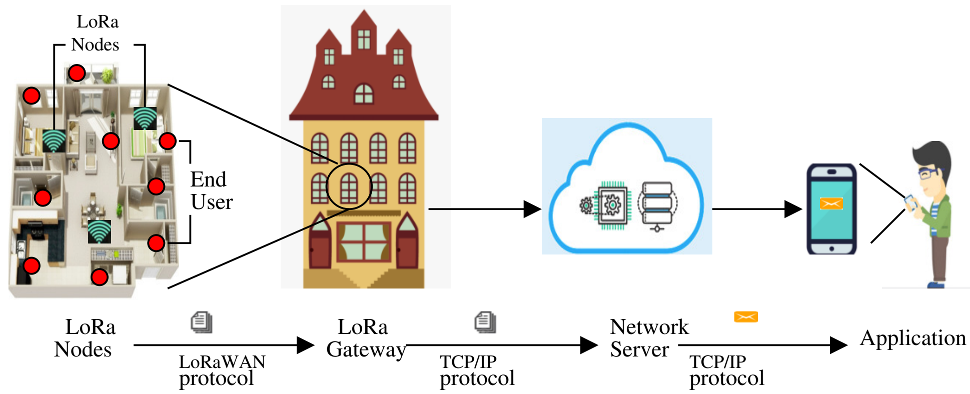


Figure 4.2: A smart home scenario using LoRa network.

4.3 Finding the feasible subsets of SFs

We consider a LoRa network consisting of multiple LNs, one LG, one NS, and an application, as shown in Figure 4.2. We use Calligraphic letters such as $\mathcal{N}$, Italic capital letters such as N, and Italic small letters such as n to denote sets, numbers, and index, respectively. **Bold** notations are used for vector quantity throughout the chapter. Denote the set of LNs by $\mathcal{N} = \{1, 2, \cdots, N\}$ and set of SFs by $\mathcal{F} = \{7, 8, \cdots, 12\}$. LNs transmit data from End users to the LG and receive price for the data. Let $\mathcal{F}_n = \{1, 2, \cdots, F_n\}$ be the combination of feasible SFs used by a LN n where 1 indicates the first SF in the combination $\mathcal{F}_n$, 2 indicates the second SF in the combination $\mathcal{F}_n$, and so on. These feasible subsets of SFs is calculated based on the suitability of the SFs defined in Definition 4.1 and maximum number of SFs that can be used by LNs. Followings are the assumptions considered in the chapter.

Assumption 1: *We consider deployed LNs are both static and mobile in nature and therefore are randomly distributed within the network.*

Assumption 2: *We assume that LNs have a constant stream of information to transmit at the NS via LG.*

Definition 4.1 *(Suitability of SF) A SF $f \in \mathcal{F}$ is suitable for a LN $n \in \mathcal{N}$ for transmitting its data if n lies in the range of f. Range of f is $[0, d_f]$ where d_f is the maximum distance at which the LN n can transmit on SF f. For example in Figure 4.3 suitable SFs for a LN 1 are SF10, SF11, and SF12.*

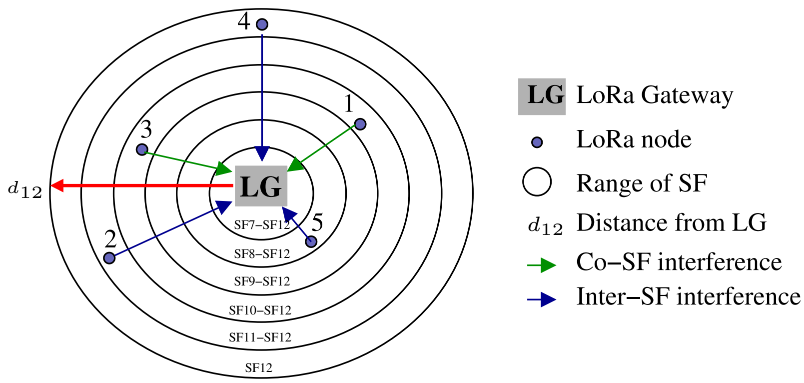


Figure 4.3: Network deployment of a LoRa illustrating the arrangement of LNs.

Definition 4.2 *(Partially ordered set) Partially ordered set (Poset) is an ordered pair of binary relation defined over a set $\mathcal{Z}$, (i.e., $\leq$, $\mathcal{Z}$) that satisfies following properties $\forall u, v, w \in \mathcal{Z}$:*

- *Reflexivity: i.e., $u \leq u$.*
- *Antisymmetry: i.e., if $u \leq v$ and $v \leq u \Rightarrow u = v$.*
- *Transitivity: i.e., if $u \leq v$ and $v \leq w \Rightarrow u \leq w$.*

A Poset of all possible subsets can be visualized using Hasse diagram where each node corresponds to a subset of SFs for a given LN. In a Hasse diagram, we evaluate each node for its feasibility and assign a color accordingly, *i.e.*, 'red' if node is feasible and 'white' otherwise. A node is said to be *feasible* for a LN $n \in \mathcal{N}$ if it consists only suitable SFs and number of SFs is not more than F_n, where suitability of a SF for a given LN is defined in Definition 4.1.

Procedure 4.1 illustrates the steps for finding the feasible subsets of SFs for $\mathcal{N}$. The inputs of the procedure are the set of LNs, LG, and maximum number of SFs that can be used by each LN, *i.e.*, F_n where $n \in \mathcal{N}$. Steps (2)-(4) draw the Hasse diagram for n where nodes consist the subsets of SFs. Steps (6)-(8) check the feasibility of each node n, *i.e.*, all SFs lie in the node are suitable and total number of SFs are less than F_n. The algorithm changes the color of the node from white to red if it is feasible. Let z_n^i and f_{nj}^i denote i^{th} feasible subset of LN n and j^{th} suitable SF of z_n^i subset, respectively. The outputs of Procedure 4.1 are the feasible subsets of SFs, denoted by $\mathbf{Z}$, where $\mathbf{Z} = [\mathbf{Z_1}, .., \mathbf{Z_n}, .., \mathbf{Z_N}]$, $\mathbf{Z_n} = [z_n^1, .., z_n^i, .., z_n^z]$, z_n^i consists $\{f_{n1}^i, \cdots, f_{nk}^i\}$ SFs, and $k \leq F_n$.

Procedure 4.1: Finding the feasible subsets of SFs

Input: Set N, LG, and F_n for $n \in \mathcal{N}$.
Output: Feasible subsets $\mathbf{Z}$ of SFs for $\mathcal{N}$.

1 **for** *each LN* $n \in \mathcal{N}$ **do**
2   Create Hasse diagram of Poset for n;
3   Assign 'white' color to all nodes of Hasse diagram;
4   Compute the possible SFs for n;
5   **for** *each node of Hasse diagram* **do**
6   **if** *node is 'white'* **then**
7   **if** *Consists possible SFs* ***and*** *No. of SFs* $\leq F_n$ **then** `// A feasible subset of` n
8   Assign 'red' color to *node*;

Example 1 *Let consider a scenario where a LN $n \in \mathcal{N}$ lies in third ring i.e., LN 1 as shown in Figure 4.3. The suitable SFs for transferring the data to the LG are SF_{10}, SF_{11}, and SF_{12} because it lies outside the transmission range of SF_7, SF_8, and SF_9. In this scenario, if n wants to use maximum two out of three possible SFs ($F_n = 2$) then it can use any of the following feasible combination of SFs: SF_{10}, SF_{11}, SF_{12}, (SF_{10}, SF_{11}), (SF_{10}, SF_{12}), or (SF_{11}, SF_{12}). Such combinations are the power order set (Poset) of the possible three SFs. The Hasse diagram for all possible subsets of SFs is shown in part (a) of Figure 4.4. Initially, all nodes are of 'white' color. Part (b) of Figure 4.4 shows the red color nodes which consist feasible subsets of SFs for n.*

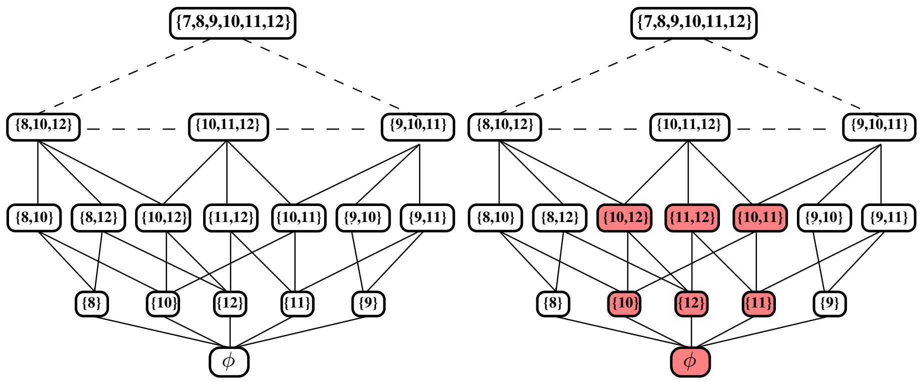


Figure 4.4: Illustration of Hasse diagram of feasible subsets of SFs for n. Part (a) shows possible subsets of SFs. Part (b) shows the all feasible subsets of SFs.

4.4 Identify optimal subset of SFs for each LN

In this section, we propose a follower and leader model and analyze the Nash Equilibrium (NE) among LNs and Stackelberg Equilibrium (SE) between LG and LNs, respectively. We then, propose a procedure to identify the optimal subset of SFs for each LN out of the estimated feasible subsets of SFs by using the follower and leader model.

4.4.1 Analysis of follower (LoRa node) game

This section derives the expression of the follower model for estimating the utility. The section also formulates and analyzes the optimization problem of followers by using the follower model.

4.4.1.1 Follower model

At the beginning of the game, LG allocates suitable SFs and price per data for each LN. Such prices are given based on the SFs and significance of the data. Based on the LG strategy, each LN reacts by selecting an optimal strategy which maximizes its utility. In the proposed game model, we consider positive magnitude of the cost if a LN gets some revenue from others. Negative magnitude indicates if a LN needs to pay to the others or consumes its own resources. The utility function of a LN $n \in \mathcal{N}$ consists the following terms:

• **Cost paid to the end users:** The LoRa network consists the end users which provide data to the LNs as shown in Figure 4.2. Due to the privacy concern, apartment owner does not want to share the data collected from the end users willingly. Therefore, LN gives the incentive to motivate the owner for providing their sensory data collected from EUs. Let t_n^f denotes the time duration of data forwarding from the connected EUs to the LG via LN $n \in \mathcal{N}$ on SF $f \in \mathcal{F}_n$. The LN n needs to pay cost to the EUs for that data. The cost function due to the data transmission for $\mathbf{t}_n$ time duration on all available SFs in the combination $\mathcal{F}_n$ from end users to LN n is denoted by $L_e(\mathbf{t}_n)$. The cost function should be monotonically increasing function of time duration *i.e.,* the cost increases with the increase in the time duration and strictly convex function.

Table 4.1: Illustration of properties of different functions. The Negativity, Convexity, and Strict convexity are denoted as Ne, Co, and St, respectively.

Cost functions	**Ne**	**Co**	**St**	**Remarks**
Linear	$\surd$	$\surd$	—	$0 \nless \mathscr{F}''(x)$
Quadratic	—	$\surd$	$\surd$	$\mathscr{F}(x) \nless 0$, $x < 0$
Sign Quadratic	$\surd$	—	—	$0 \nleq \mathscr{F}''(x)$
Exponential	—	$\surd$	$\surd$	$\mathscr{F}(x) \nless 0$, when $x < 0$.
Logarithmic	$\surd$	$\surd$	$\surd$	$0 \leq \mathscr{F}''(x)$, $\mathscr{F}(x) < 0$, when $x < 0$.

Table 4.1 illustrates that why linear, quadratic, and exponential value functions of one variable are not suitable as cost function in LoRa network. A logarithmic barrier

function is a strictly convex function because it is convex and has greater curvature than a linear function. The cost function is therefore given as

$$L_e(\mathbf{t}_n) = \sum_{f=1}^{F_n} \left[-e_n^f \log \left(1 - \frac{t_n^f}{\delta} \right) \right], \tag{4.1}$$

where e_n^f is the pricing coefficient determined by LN n on SF f and δ is a parameter that is introduced to give cost values very close to the values obtained by a quadratic one. The relation between $L_e(\mathbf{t}_n)$ function and the quadratic one can be understood from its Taylor expansion. The Taylor expansion of $log\,(1-y)$ is given by

$$\begin{aligned} \log(1-y) &= -\int_0^y \left(\frac{1}{1-y} \right) dy = -\int_0^y \left(1 + y + y^2 + ... \right) dy, \\ &= -\,y - \frac{y^2}{2} - \frac{y^3}{3} - \frac{y^4}{4} - .... \end{aligned} \tag{4.2}$$

Due to the decimal value of y, it tends to be negligible as increase in the power of the fraction. Neglect the higher degree of y in Equation 4.2 and substituting $y = \frac{t_n^f}{\delta}$, the cost function $L_e(\mathbf{t}_n)$ from Equation 4.1 is given by

$$L_e(\mathbf{t}_n) \approx \sum_{f=1}^{F_n} \left[-e_n^f \left(-\frac{t_n^f}{\delta} - \frac{{t_n^f}^2}{2\delta^2} \right) \right] = \sum_{f=1}^{F_n} \left[\frac{e_n^f t_n^f}{\delta} + \frac{e_n^f {t_n^f}^2}{2\delta^2} \right]. \tag{4.3}$$

• **Processing cost of a LN:** Let $L_p(\mathbf{t}_n)$ be the processing cost for computing the received data from the EUs to LN n during $\mathbf{t}_n$ time duration on all available SFs in the combination $\mathcal{F}_n$. The processing cost depends on the hardware configuration of the LN n where the magnitude of the processing cost is negative. Let a_n and b_n are the hardware constants of LN n. Similar to the cost function modeled in paper [63], the processing cost function is defined as,

$$L_p(\mathbf{t}_n) = \sum_{f=1}^{F_n} \left[a_n \times (t_n^f)^2 + b_n \times t_n^f \right]. \tag{4.4}$$

• **Dissatisfaction cost of a LN:** Dissatisfaction cost occurs due to interference problem during data transmission from LNs to LG. In the proposed work, we assume that N number of LNs, out of all deployed LNs in the network, are transmitting data at any given time. Let h_n^f and W are the channel gain and bandwidth of a link between LN $n \in \mathcal{N}$ on SF $f \in \mathcal{F}_n$ and LG, respectively. The transmission power of LN n on SF f is denoted by p_n^f and σ^2 is the power of white Gaussian noise. Using Shannon channel capacity [64], the transmission rate (bits/second) of LN n on SF f is given by

$$R_n^f = W \log \left(1 + \frac{p_n^f h_n^f}{\sigma^2} \right). \tag{4.5}$$

The LN n in the system uses available SF for transmitting the received data to the LG. Let d_n^f be the number of LNs in the system which are using the same or lower SF from the SF f of LN n. Such d_n^f LNs create the interference problem to LN n because they also uses the equal or higher transmission power during transmission of data to the LG. Such interference problem creates dissatisfaction to LN n during data transmission. The channel gain and transmission power of other d_n^f LNs are denoted by h_j and p_j, where $0 \leq p_n^f \leq p_j$ and $1 \leq j \leq d_n^f$, respectively. The transmission rate of LN n in the presence of other d_n^f LNs is given by

$$r_n^f = W \log \left(1 + \frac{p_n^f h_n^f}{\sum_{j=1}^{d_n^f} p_j h_j + \sigma^2} \right). \tag{4.6}$$

Dissatisfaction cost of LN n during t_n^f time duration is the difference of the transferred data with and without interference problem. This cost increases based on the deviance from the transmission rate without interference. Therefore, using Taguchi loss function, the dissatisfaction cost during $\mathbf{t}_n$ time interval can be defined as

$$L_s(\mathbf{t}_n) = \sum_{f=1}^{F_n} \left[\theta_n^f \left(t_n^f R_n^f - t_n^f r_n^f\right)^2\right], \tag{4.7}$$

where θ_n^f is the loss function coefficient of LN n on SF f.

• **Price gained from LG:** LNs transmit data to the LG as shown in Figure 4.2. LG pays price to the LNs for providing such data. In SG, price function of LN n also depends on the strategy of other LNs. The Stackelberg Equilibrium (SE) estimates the total data from all the LNs which is known as forecasted data denoted by D. Due to greedy nature, each LN wants to maximize its utility therefore they try to deviate from the stable state. Hence, to prevent LNs from deviation, price function reduces price when any LN transmits data for more time duration from the time duration estimated by the SE. Therefore, price function can be defined as

$$p(\mathbf{t}) = \gamma \left(1 + \frac{D - \sum_{i=1}^{N} t_i^f r_i^f}{\alpha}\right) + x_n^f, \tag{4.8}$$

where $\mathbf{t} = [t_1^f, t_2^f, ..., t_N^f]^T, \forall f \in \mathcal{F}_n$, γ is the pricing parameter to give the elasticity in pricing, α is normalization parameter, and x_n^f is the base price assigned by LG to LN n on available SF f. The base price depends on the importance of transmitted data from LN n and the load on SF f. The price paid by LG to LN n for the data that is generated in t_n^f time duration is the product of the data transmitted by LN n and $p(\mathbf{t}^f)$. If a LN decreases its transmission time from the time estimated at the equilibrium state, price gain from LG decreases which also prevents LNs from the deviation.

$$L_g(\mathbf{t}_n) = \sum_{f=1}^{F_n} \left[p(\mathbf{t}) t_n^f r_n^f\right], n \in \mathcal{N}. \tag{4.9}$$

4.4.1.2 Follower game

Based on the price announced by the leader, the followers compete with each other to maximize their utilities. The follower game can be formulated with the following information:

- **Players:** Each LN is one player and there are N players.
- **Strategies:** A LN n obtains price x_n^f from the LG on SF f, where $n \in \mathcal{N}$. The strategy of follower n is the time duration t_n^f for forwarding the data from end users to the LG on SF f.
- **Utilities:** The utility of player n is the net utility, which is the price gained from LG minus the cost paid by the LN, (*i.e.*, end users cost, processing cost, and dissatisfaction cost). Thus, utility of player n from Equation 4.10 is given as

$$\begin{aligned} U_n(\mathbf{t}_n, \mathbf{t}_{-n}) =& L_g(\mathbf{t}_n) - (L_e(\mathbf{t}_n) + L_p(\mathbf{t}_n) + L_s(\mathbf{t}_n)), \\ =& \sum_{f=1}^{F_n} \left[t_n^f \left(\left(\gamma' - \frac{\gamma}{\alpha} \sum_{i=1}^{N} t_i^f r_i^f + x_n^f \right) r_n^f - {b'_n}^f \right) - {t_n^f}^2 \left(\theta_n^f \left(R_n^f - r_n^f \right)^2 + {a'_n}^f \right) \right], \end{aligned} \tag{4.10}$$

where $\gamma' = \gamma + \frac{\gamma}{\alpha} D$, ${b'_n}^f = \frac{e_n^f}{\delta} + b_n$, and ${a'_n}^f = \frac{e_n^f}{2\delta^2} + a_n$.

In a competitive environment, each player aims at maximizing its own net utility, the follower level game is expressed as follows:

Problem 1

$$\begin{aligned} \max_{\mathbf{t}_n} \quad & U_n(\mathbf{t}_n, \mathbf{t}_{-n}) \\ s.t. \quad & \sum_{f=1}^{F_n} t_n^c \leq {t_n}^{max}, \\ & t_n^f \geq 0, \quad n \in \mathcal{N}, f \in \mathcal{F}_n. \end{aligned} \tag{4.11}$$

Problem 1 expresses that the LN n optimizes the strategy to maximize its utility and constraint imposes that the duration of forwarding the data from end users to the LG must not be greater than its duty cycle, where $n \in \mathcal{N}$.

4.4.1.3 Nash Equilibrium (NE) analysis

NE is a stable outcome of the follower game where all LNs interact through self-optimization and reach a point by calculating best response strategy where no LN has any incentive to deviate. To define the best response strategies of LNs, we use optimization Problem 1 (Equation 4.11) which maximize the utility of LNs with the constraints. All LNs are non-cooperative. Theorem 4.1 proves the existence of the best response strategy of each LN.

Theorem 4.1 *Let t_n^f be the strategy of a LN $n \in \mathcal{N}$ for data forwarding time duration on SF $f \in \mathcal{F}$ to the LG. The best response $t_n^{f\star}$ of the LN n is given as*

$$t_n^{f\star} = Q_n^f - \frac{1}{A_{nn} \sum_{f=1}^{F_n} (\frac{1}{A_{nn}})} \left(\sum_{f=1}^{F_n} Q_n^f - t_n^{max} \right),$$

where

$$Q_n^f = \frac{\gamma' r_n^f - {b'_n}^f + x_n^f r_n^f - \frac{\gamma}{\alpha} r_n^f \sum_{i=1, i \neq n}^{N} t_i^f r_i^f}{A_{nn}} \text{ and } A_{nn} = 2{a'_n}^f + 2\frac{\gamma}{\alpha} {r_n^f}^2 + 2\theta_n^f \left(R_n^f - r_n^f \right)^2. \tag{4.12}$$

Proof: Using Lagrangian multipliers $\lambda_{n,1}$ and $\lambda_{n,2}$ for constraints defined in Equation 4.11

$$\begin{aligned}\mathcal{L}_n(\mathbf{t}_n, \mathbf{t}_{-n}) = & \sum_{f=1}^{F_n} \left[t_n^f \left(\left(\gamma' - \frac{\gamma}{\alpha} \sum_{i=1}^{N} t_i^f r_i^f + x_n^f \right) r_n^f - {b'_n}^f \right) \right. \\ & \left. - {t_n^f}^2 \left(\theta_n^f \left(R_n^f - r_n^f \right)^2 + {a'_n}^f \right) \right] + \lambda_{n,1} t_n^f - \lambda_{n,2} \left(\sum_{f=1}^{F_n} t_n^f - t_n^{max} \right) = 0,\end{aligned} \tag{4.13}$$

$$\text{s.t.} \qquad \lambda_{n,1} t_n^f, \lambda_{n,2} \left(\sum_{f=1}^{F_n} t_n^f - t_n^{max} \right) = 0, \lambda_{n,1}, t_n^f \geq 0, \lambda_{n,2} > 0. \tag{4.14}$$

The first and second order derivatives of the utility function of a follower defined in Equation 4.14 with respect to t_n^f is given as

$$\begin{aligned} \frac{d\mathcal{L}_n(\mathbf{t}_n, \mathbf{t}_{-n})}{dt_n^f} &= (\gamma' - \frac{\gamma}{\alpha} \sum_{i=1}^{N} t_i^f r_i^f - \frac{\gamma}{\alpha} t_n^f r_n^f + x_n^f) r_n^f - 2 t_n^f {a'_n}^f \\ &- {b'_n}^f - 2 t_n^f \theta_n^f (R_n^f - r_n^f)^2 + \lambda_{n,1} - \lambda_{n,2}. \end{aligned} \tag{4.15}$$

$$\frac{d^2\mathcal{L}_n(\mathbf{t}_n, \mathbf{t}_{-n})}{d(t_n^f)^2} = -2{a'_n}^f - 2\frac{\gamma}{\alpha} {r_n^f}^2 - 2\theta_n^f \left(R_n^f - r_n^f \right)^2. \tag{4.16}$$

We can see from Equation 4.16 that $\frac{d^2\mathcal{L}_n(\mathbf{t}_n, \mathbf{t}_{-n})}{d(t_n^f)^2} < 0$.

Since the value of second order derivative of $\mathcal{L}_n(\mathbf{t}_n, \mathbf{t}_{-n})$ is negative, the utility function $U_n(\mathbf{t}_n, \mathbf{t}_{-n})$ in Equation 4.10 is concave in $\mathbf{t}_n$ and **Problem 1** is convex optimization problem. Therefore the follower level game has at least one NE [63].

The value of t_n^f can be obtained from the coefficient matrix by setting the first order derivative of utility function as calculated in Equation 4.15 equals to zero. Let A and B are the coefficient matrices. Equation 4.15 can be rewritten as

$$\overbrace{\begin{bmatrix} A_{11} & \frac{\gamma}{\alpha} r_1^f r_2^f & \dots & \frac{\gamma}{\alpha} r_1^f r_N^f \\ \frac{\gamma}{\alpha} r_2^f r_1^f & A_{22} & \dots & \frac{\gamma}{\alpha} r_2^f r_N^f \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\gamma}{\alpha} r_N^f r_1^f & \frac{\gamma}{\alpha} r_N^f r_2^f & \dots & A_{NN} \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} t_1^f \\ t_2^f \\ \vdots \\ t_N^f \end{bmatrix}}^{\mathbf{t}} = \overbrace{\begin{bmatrix} {\beta_1}^f + x_1^f r_1^f \\ {\beta_2}^f + x_2^f r_2^f \\ \vdots \\ {\beta_N}^f + x_N^f r_N^f \end{bmatrix}}^{B}, \tag{4.17}$$

where, $A_{nn} = 2{a'_n}^f + 2\frac{\gamma}{\alpha} {r_n^f}^2 + 2\theta_n^f \left(R_n^f - r_n^f \right)^2$, ${\beta_n}^f = \gamma' r_n^f - {b'_n}^f + \lambda_{n,1} - \lambda_{n,2}$, and

$1 \leq n \leq N$.

From the strictly diagonal dominant theorem [65], the coefficient matrix A is nonsingular if $\gamma \leq \frac{\alpha A_{nn}}{r_n^f \sum_{i=1,i\neq n}^{N} r_i^f}, \forall n \in \mathcal{N}$. Since matrix A is nonsingular, inverse of the matrix A is possible. Therefore Best response strategies of followers $\mathbf{t}^f = [t_1^f, \cdots, t_n^f, \cdots, t_N^f]$ as the time duration to transmit data to the LG on SF f can be calculated as $A^{-1}B$, which is defined as

$$t_n^{f\star} = \frac{1}{A_{nn}}(\beta_n{}^f + x_n^f r_n^f - \frac{\gamma}{\alpha} r_n^f \sum_{i=1,i\neq n}^{N} t_i^f r_i^f). \tag{4.18}$$

By putting value of t_n^f into the constraint of Equation 4.14, we get

$$\lambda_{n,2} = \frac{1}{\sum_{f=1}^{F_n} A_{nn}} \left(\sum_{f=1}^{F_n} \left(\frac{\gamma' r_n^f - b'_n{}^f + x_n^f r_n^f - \frac{\gamma}{\alpha} r_n^f \sum_{i=1,i\neq n}^{N} t_i^f r_i^f}{A_{nn}} - t_n^{max} \right) \right). \tag{4.19}$$

Substituting $\lambda_{n,2}$ from Equation 4.19 into Equation 4.18 and hence proved. □

Proposition 2 *The best response strategy of a LN $n \in \mathcal{N}$ is unique.*

Proof: The best response strategy of a follower n defined in Equation 4.12, is unique if it consists positivity, monotonicity, and scalability. From Equation 4.12, we confirm that $t_n^f > 0$ because $\sum_{f=1}^{F_n} Q_n^f - t_n^{max}$ will always return negative value and therefore confirms its positivity. Let t_{-n}^f and t'^f_{-n} are the data transmission time duration of LN other than n on SF f. Equation 4.12 shows that if $t_{-n}^f \geq t'^f_{-n}$ then $\mathscr{F}(t_{-n}^f) \geq \mathscr{F}(t'^f_{-n})$ and therefore t_n^f consists monotonicity. The best response strategy t_n^f consists scalability if $\mu\mathscr{F}(t_{-n}^f)$ is greater than $\mathscr{F}(\mu t_{-n}^f)$, *i.e.*, $\mu\mathscr{F}(t_{-n}^f) - \mathscr{F}(\mu t_{-n}^f) \geq 0$ and $\mu \geq 0$. Using Equation 4.12, we get

$$\begin{aligned}\mu\mathscr{F}(t_{-n}^f) - \mathscr{F}(\mu t_{-n}^f) = \mu(Q_n^f - \frac{1}{A_{nn}\sum_{f=1}^{F_n}(\frac{1}{A_{nn}})}(\sum_{f=1}^{F_n} Q_n^f - t_n^{max}))\\ - \mu Q_n^f - \frac{1}{A_{nn}\sum_{f=1}^{F_n}(\frac{1}{A_{nn}})}(\sum_{f=1}^{F_n} \mu Q_n^f - t_n^{max}) \geq 0.\end{aligned}$$

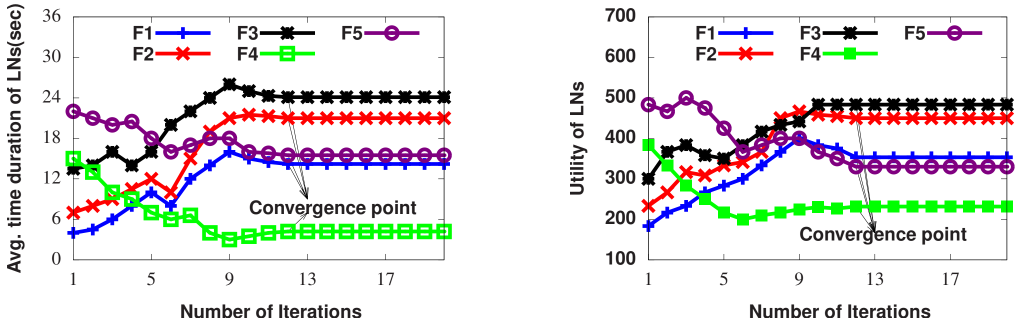
□

We propose a distributed algorithm for finding the NE by using the estimated best response strategy of LNs. Algorithm 4.1 allows each LN to play its best response strategy, announce only its data forwarding time duration on SF $f \in \mathcal{F}_n$ to the LG, and repeat until equilibrium is reached. Since the objective function is bounded from below, the iterations will eventually converge to a fixed point, *i.e.*, a NE point similarly as in [66]. Once LNs reach this point, they will have no incentive to change since the NE is unique and convergence is achieved.

Algorithm 4.1: Nash Equilibrium among the LNs

/* Run the following at each LN $n \in \mathcal{N}$. */
Input: Precision threshold η, $\tau \leftarrow 0$, $t_n^f[0] \leftarrow \eta$;
Output: Best response strategy t_n^f of n ;
do
  $\tau \leftarrow \tau + 1$;
  /* Using Equation 4.12 for estimating $t_n^f[\tau+1]$. */
  $t_n^f[\tau+1] = Q_n^f[\tau] - \sum_{f=1}^{F_n} A_{nn}(\sum_{f=1}^{F_n} Q_n^f[\tau] - t_n^{max})$;
while $\left(\left\|t_n^f[\tau+1] - t_n^f[\tau]\right\| > \eta\right)$;

Example 2 *Let we consider* 10 *LNs to obtain the convergence rate of the follower game which will increase as the number of followers increases. Initially, all followers set their time duration randomly but after each iteration, according to the strategies of other LNs, they re-calculate their time duration at which utility of all LNs maximized. Parts (a) and (b) of Figure 4.5 illustrate that the followers (denoted as F1$\cdots$F5 among 10 LNs) iteratively update their time duration and utilities, respectively and converges to a stable value at iteration* 13, *confirming the convergence and stability of Algorithm 1. Note that time duration of F1 is less than F5 in the stable state in part (a) of Figure 4.5 whereas utility is high of F1 in part (b) of Figure 4.5. This is because of the different pricing coefficients.*



(a) Avg. time duration (b) Avg. utility

Figure 4.5: Avg. time duration and utility of LNs at convergence of Algorithm 4.1.

4.4.2 Analysis of leader (LoRa gateway) game

This section derives the expression of the leader model for estimating the utility. The section also formulates and analyzes the optimization problem of leaders by using the leader model.

4.4.2.1 Leader model

The energy consumption of a LN is high when using the low SF for forwarding data from end users to the LG. Therefore, the LNs do not transmit data on low SF voluntarily, which causes the interference problem. The leader solves such problem by making the pricing strategy in such a way that the LNs also want to use the low SF for forwarding the data and maximizes their own net utility. The utility function of the leader consists the following terms:

• **Price paid to the LNs:** The LG pays prices to the LNs for the received data as given in Equation 4.9. The LoRa network consists N LNs where LN n forwards data for $\mathbf{t}_n$ time duration with $\mathbf{r}_n$ transmission rate. The price paid to the LNs by the LG is given as

$$G_p(\mathbf{t}) = \sum_{n=1}^{N}\sum_{f=1}^{F_n}\left(\gamma\left(1+\frac{D-\sum_{i=1}^{N} t_i^f r_i^f}{\alpha}\right)+x_n^f\right)t_n^f r_n^f. \tag{4.20}$$

• **Revenue gain from applications:** The LG gains price from the applications which are connected to the NS for providing the data. The revenue generated from the data received by the LNs is given by

$$G_r(\mathbf{t}) = \sum_{n=1}^{N} \sum_{f=1}^{F_n} \left(g_{1n} t_n^f r_n^f - g_{2n} \left(t_n^f r_n^f \right)^2 \right), \tag{4.21}$$

where g_{1n} and g_{2n} are the coefficients such that $g_{1n} >> g_{2n}$ which are used to measure the gain received from data on LG. The quadratic form of the utility function allows for tractable analysis and also serves as a good second-order approximation for the broader class of concave utility.

4.4.2.2 Leader game

Based on the strategies selected by followers, the leader optimizes its strategy to maximize its utility. The leader game can be formulated with the following information:

- **Player:** LoRa gateway.
- **Strategy:** The price x_n^f paid to the LN n for forwarding the data using SF f.
- **Utility:** The utility for a leader is the net utility which is the generated revenue minus price paid to LNs.

$$\begin{aligned} U(\mathbf{x}_n, \mathbf{x}_{-n}) =& G_r(\mathbf{t}) - G_p(\mathbf{t}), \\ =& \sum_{n=1}^{N} \sum_{f=1}^{F_n} \left[g_{1n} t_n^f r_n^f - g_{2n} (t_n^f r_n^f)^2 - \left(\gamma' - \frac{\gamma}{\alpha} \sum_{i=1}^{N} t_i^f r_i^f + x_n^f \right) t_n^f r_n^f \right]. \end{aligned} \tag{4.22}$$

In SG, the leader aims to maximize its revenue. The leader level game is expressed as follows:

Problem 2

$$\begin{aligned} \max_{\mathbf{x}_n} \quad & U(\mathbf{x}_n, \mathbf{x}_{-n}) \\ s.t. \quad & 0 \leq x_n^f \leq X^{max}, n \in \mathcal{N}, f \in \mathcal{F}_n, \end{aligned} \tag{4.23}$$

where X^{max} is the maximum price limit given by LG to the LN. LG optimizes strategy to maximize its utility and constraint imposes that price paid to the LN to be greater or equal to zero and not exceed its limit. Using backward induction method from Equation 4.12, Problem 2 can be rewritten as

Problem 3

$$\max_{\mathbf{x}_n} U(x_n^f, t_n^f) = \sum_{n=1}^{N} \sum_{f=1}^{F_n} \left(t_n^{f\star} \left(\, g_{1n} - \gamma' + \frac{\gamma}{\alpha} \sum_{i=1}^{N} t_i^{f\star} r_i^f - x_n^f \, \right) r_n^f - g_{2n} (t_n^{f\star} r_n^f)^2 \right).$$

$$s.t. \quad 0 \leq x_n^f \leq X^{max}, n \in \mathcal{N}$$

4.4.2.3 Stackelberg Equilibrium analysis

The optimal strategy of the leader and the best response strategies of the followers, constitutes SE of the game. To find out the optimal strategy of LG, Proposition 3 proves the optimality of the best response strategies of LNs.

Proposition 3 *The best response strategy of a LN $n \in \mathcal{N}$ is an optimal solution.*

Proof: Problem 1 is convex optimization problem because objective function $U_n(\mathbf{t}_n, \mathbf{t}_{-n})$, defined in Equation 4.11, is the quadratic function given x_n^f and constraint is affine. From Equation 4.11, $t_n^{f\star} = \min\{t_n^f, t_n^{max}\}$. Let assume $t_n^f \geq t_n^{max}$ and hence $t_n^{f\star} = t_n^{max}$. The objective function of LG $U(\mathbf{t}_n, \mathbf{t}_{-n})$, defined in Equation 4.23, after substituting $t_n^{f\star}$ is to maximize Problem 2. However, we can observe that the objective function is a decreasing function with respect to x_n^f. For maximizing this function, we must have

$$x_n^f \leq \frac{\left(\tilde{Q}_n^f - \frac{1}{\sum_{f=1}^{F_n} \frac{1}{A_{nn}}}(\sum_{f=1}^{F_n} \frac{\tilde{Q}_n^f}{A_{nn}} - t_n^{max}) - A_{nn} t_n^{max}\right)}{r_n^f \left(\frac{1}{A_{nn} \sum_{f=1}^{F_n} \frac{1}{A_{nn}}} \sum_{j=1, j\neq f}^{F_n} \frac{x_n^j}{A_{nn}} - 1\right)}, \tag{4.24}$$

where $\tilde{Q}_n^f = \gamma' r_n^f - {b'}_n^f - \frac{\gamma}{\alpha} r_n^f \sum_{i=1, i\neq n}^{N} t_i^f r_i^f$. Thus $t_n^{f\star} = t_n^f$, which completes the proof.

□

We propose the following theorem to prove that the obtained unique and optimal best response strategies of LNs is admitted by LG.

Theorem 4.2 *The LG finds its unique optimal best response strategy given the optimal strategies of LNs i.e., estimated transmission time duration at NE state.*

Proof: To prove the uniqueness of the best response strategy of LG, we use Hessian matrix of Problem 3 by taking the second order partial derivative of $U(x_n^f, t_n^f)$ with respect to x_n^f and x_l^f, is given as

$$\frac{d^2 U(x_n^f, t_n^f)}{dx_n^f dx_l^f} = \begin{cases} 2\Delta t_n^f r_n^f (\frac{\gamma}{\alpha} \Delta t_n^f r_n^f - 1) - 2 g_{2n} \Delta {t_n^f}^2 {r_n^f}^2 & if n = l \\ 0 & else \end{cases} \tag{4.25}$$

where $\Delta t_n^f = \frac{r_n^f}{A_{nn}}(1 - \frac{1}{A_{nn}\sum_{f=1}^{F_n} \frac{1}{A_{nn}}})$. Since value of α is almost equal to forcasted demand of data therefore is much larger than γ and $(\frac{\gamma}{\alpha} \Delta t_n^f r_n^f - 1)$ returns negative value. Due to the positivity of Δt_n^f negativity of $(\frac{\gamma}{\alpha} \Delta t_n^f r_n^f - 1)$, diagonal elements of Hessian matrix will give negative value. The Hessian matrix of $U(x_n^f, t_n^f)$ is strictly negative definite which implies that Problem 3 is a standard convex maximization problem and hence proved.

□

The optimal solution of Problem 3 can be obtain by Karush-Kuhn-Tucker conditions [67]. Therefore the Lagrangian of Problem 3 is given as

$$\mathcal{L}(x_n^f, t_n^f) = \sum_{f=1}^{F_n} t_n^{f\star} \left(g_{1n} - \gamma' + \frac{\gamma}{\alpha} \sum_{i=1}^{N} t_i^{f\star} r_i^f - x_n^f \right) r_n^f - g_{2n} (t_n^{f\star} r_n^f)^2 + \Lambda_1 x_n^f - \Lambda_2 (x_n^f - X^{max}), \tag{4.26}$$

where Λ_1 and Λ_2 are the Lagrangian multipliers of x_n^f. Equate the derivative of $\mathcal{L}$ with respect to x_n^f is equals to zero

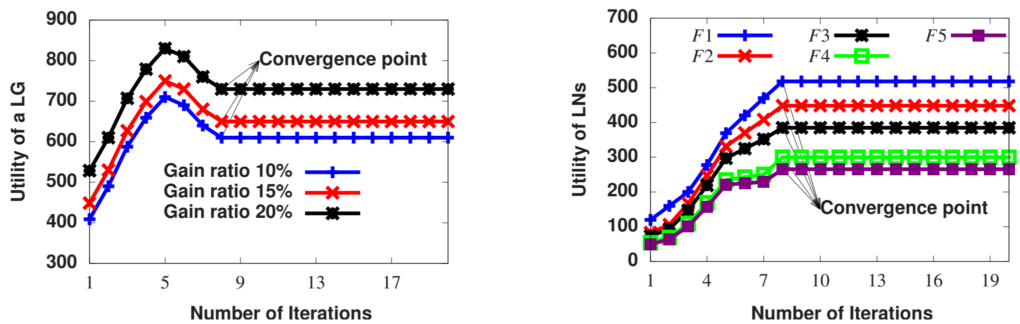
$$x_n^{f\star} = \frac{t_n^{f\star}}{\Delta t_n^f}(\frac{\gamma}{\alpha}\Delta t_n^f r_n^f - 1) - 2g_{2n}t_n^f r_n^f + (g_{1n} - \gamma' + \frac{\gamma}{\alpha}\sum_{i=1}^{N} t_i^{f\star} r_i^f). \tag{4.27}$$

The LoRa network consists a SE when a LG (leader) estimates the optimal price for maximizing their utility while the LNs select their time duration for data forwarding to maximize their benefit with minimum energy consumption. We propose a distributed algorithm as explained in Algorithm 4.2, for finding the SE by using the estimated best response strategy of LNs and optimal strategy of LG.

Algorithm 4.2: Stackelberg Equilibrium in LoRa

/* Run at LoRa gateway for all connected $n \in \mathcal{N}$. */
Input: Precision threshold ω, ϵ, $\tau \leftarrow 0$, $x_n^f[0]$;
Output: Optimal strategy $t_n^{f\star}$ of n and $x_n^{f\star}$ of LoRa gateway;
do
  $\tau \leftarrow \tau + 1$;
  /* Follower game: Each n maximizes its net utility */
  Estimate t_n^f of each $n \in \mathcal{N}$ using Algorithm 4.1;
  /* Leader game: Use Equation 4.27 for estimating $x_n^f[\tau+1]$ */
  $x_n^f[\tau+1] = x_n^f[\tau] + \epsilon \bigtriangledown_U (x_n^f[\tau])$;
while $\left(\left\|x_n^f[\tau+1] - x_n^f[\tau]\right\| < \omega x_n^f[\tau]\right)$;

Example 3 *The convergence rate of Algorithm 4.2 is shown in Figure 4.6. Similar to Example 1, we consider* 10 *LNs to obtain the convergence rate of Algorithm 4.2. Gain ratio defined as the difference between g_{1n} and g_{2n} which increases as we decrease g_{2n}. Parts (a) and (b) of Figure 4.6 illustrate that the leader and each follower iteratively update their utility based on the price of the leader, respectively and converges to a stable value at iteration* 8*, confirming the convergence and stability of the Algorithm 4.2. It can be seen from part (a) of Figure 4.6 that utility of the LG increases with the increase in gain ratio and this increment increase exponentially due to the quadratic nature of second term in Equation 4.22.*



(a) Utility of LG (b) Utility of LNs

Figure 4.6: Utility of LG and LNs at the convergence of Algorithm 4.2.

4.4.3 Finding optimal subset of SFs

A subset of the SFs for a LN $n \in \mathcal{N}$ is said to be *optimal* if the LN n satisfies its service requirement and the network maximizes its revenue. In this section, we propose a procedure to identify the optimal subset of SFs for each LN out of the estimated feasible subsets of SFs.

Procedure 4.2 uses the output of Procedure 4.1, *i.e.*, feasible subsets of SFs $\mathbf{Z_n}$, where $n \in \mathcal{N}$. It illustrates the following steps to identify the optimal subset of SFs for each n from $\mathbf{Z_n}$. Steps (1) and (2) of Procedure 4.2 repeat for each LN n and each feasible subset for n, respectively, to estimate the best response strategy of n and optimal strategy of LG. To do this, Step (3) estimates t_n^i and x_n^i for z_n^i using Algorithm 4.2, where $z_n^i \in \mathbf{Z_n}$ subset of SFs of n consists $\{f_{n1}^i, \cdots, f_{nk}^i\}$ SFs. The procedure repeats this step for all feasible subsets of n and selects an optimal subsets of SFs for n which maximizes the utilities of n and LG. The outcomes of Procedure 4.2 can be arranged in a two-dimensional matrix $a[u][n]$ where a LN n allocates $a[u][n]$ time duration for using SF $u \in \mathcal{F}$ and LN $n \in \mathcal{N}$.

Example 4 *Let continue Example 1 where LNs n, m, p, and q want to use maximum two possible SFs for satisfying its service requirement. We assume that all four LNs are lie in third ring and therefore Part (b) of Figure 4.4 shows the feasible subsets of*

Procedure 4.2: Identify optimal subsets of SFs for LNs

Input: Feasible subsets $\mathbf{Z} = [\mathbf{Z_1}, .., \mathbf{Z_n}, .., \mathbf{Z_N}]$.

Output: Optimal subset of SFs for each $n \in \mathcal{N}$.

```
for each LN n ∈ N do
    for each 'red' color node z_n^i ∈ Z_n do
        Estimate t_n^i and x_n^i for z_n^i using Algorithm 4.2;
    // An optimal subset of SFs for n
    Select t_n^i, x_n^i, and {f_n1^i, ..., f_nk^i} which maximize the utilities of n and LG;
```

SFs for the LN n, m, p, and q. We use Procedure 4.2 and compute the optimal subset of LNs n, m, p, and q as shown in blue color node of parts (a)-(d), respectively, of Figure 4.7. Table 4.2 illustrates the time duration of LNs on the optimal subset of SFs, where 0 and $\mathbf{x}$ denote non-selected and infeasible SFs, respectively. For instance, first column shows that 3 and 4 time durations are allocated to LN n for using SF_{10} and SF_{11}, respectively. Such time duration satisfies its service requirement and the network maximizes its revenue.

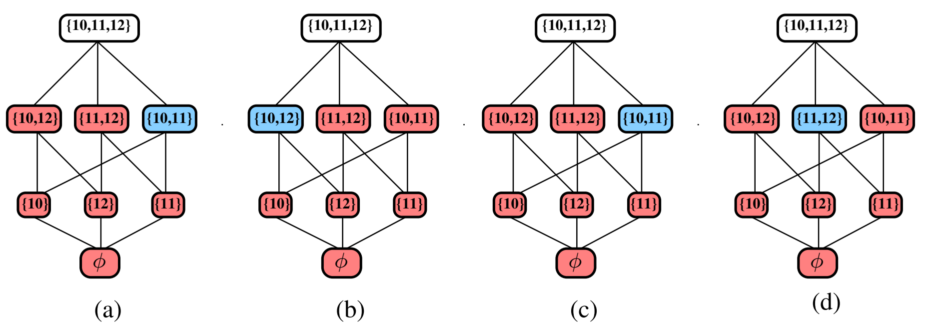


Figure 4.7: Illustration of Hasse diagram of optimal subsets of SFs for n, m, p, and q shown in parts (a)-(d), respectively.

4.5 Scheduling of LNs using SFs to minimize the waiting time

In a LoRa network, SF can be used by only one LN for a given time duration and other LNs need to wait for the availability of the SF. Such waiting time can be reduced if the LNs are properly scheduled on the SFs. Figure 4.8 shows an example scenario to

Table 4.2: Illustration of optimal subsets of SFs for LNs.

SF$\backslash\mathcal{N}$	n	m	p	q
SF_7	x	x	x	x
SF_8	x	x	x	x
SF_9	x	x	x	x
SF_{10}	3	2	7	0
SF_{11}	4	0	2	4
SF_{12}	0	2	0	1

illustrate the advantages of the scheduling of LNs for allocated time durations. Part (a) of Figure 4.8 illustrates four LNs (n, m, p, q) with their time durations for using the allocated SFs. For example, LN n uses SF_{10} and SF_{11} for 3 and 4 time durations, respectively. Parts (b) and (c) show that the proper scheduling of the allocated time duration reduces the waiting time of the LNs which helps to increase the network performance. The waiting time of the LNs is defined in Definition 4.3.

Definition 4.3 *(Waiting time of a LN) Let a set of SFs $\mathcal{F}_n$ is allocated to a LN $n \in \mathcal{N}$. Waiting time is the time n has to wait its turn to get allocated SFs free for transmission. It is the difference between the time n is ready for data transmission and the time it starts transmitting the data to the LG by using the allocated SFs.*

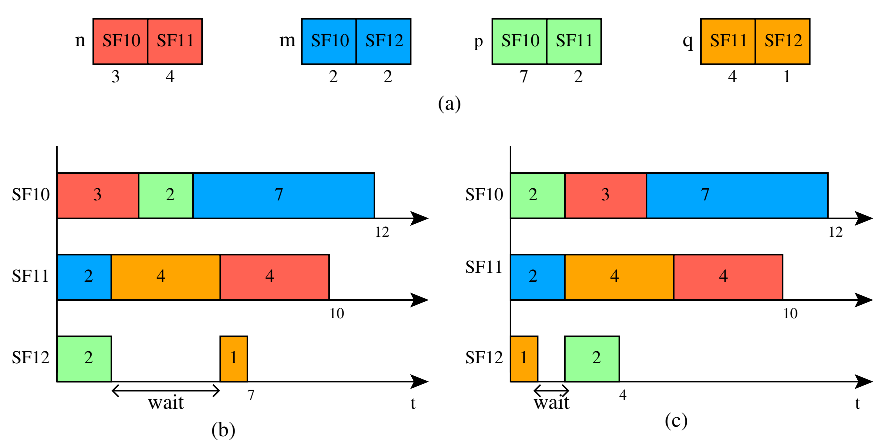


Figure 4.8: Illustration of scheduling of SFs. Part (a) shows the time duration for accessing the SFs corresponding to the LNs. Parts (b) and (c) show the without and with scheduled SFs, respectively.

We present an algorithm for scheduling of LNs on SFs to minimize the total waiting time of the LoRa network. Lemma 4.1 proves that the scheduling of the allocated time duration for using the SFs is NP-Hard problem, therefore the algorithm uses a greedy approach. This greedy approach can be solved in polynomial time. Next, Lemma 4.2 proves that Algorithm 4.3 can find out an optimal scheduling order of data transmission if the LNs are selected greedily based on ascending allocated time duration. Finally, Lemma 4.3 proves that the greedy approach in Algorithm 4.3 has polynomial time complexity.

Lemma 4.1 *Scheduling of the allocated time duration to the LNs for using the SFs is NP-Hard problem.*

Proof: Scheduling of the allocated time duration for using the SFs is NP-Hard problem which can be proven by reduction to the minimization Knapsack problem. Let w_n^f is the waiting time of LN n using SF f, then scheduling problem can be written as

$$\begin{aligned} \min \quad & \sum_{n=1}^{N}\sum_{f=1}^{F} w_n^f y_n^f \\ s.t. \quad & \sum_{f=1}^{F} t_n^f \leq {t_n}^{max}, \\ & t_n^f \geq 0, \quad n \in \mathcal{N}, f \in \mathcal{F}, \end{aligned} \tag{4.28}$$

where $y_n^f = 1$ if LN n has allocated SF f, else $y_n^f = 0$. Constraints impose that time duration taken by a LN n on all feasible SFs must not exceed its duty cycle. This is the minimization knapsack problem, which is NP-hard, and thus scheduling problem is also NP-hard and hence proved. □

• **Scheduling of LNs for minimizing the waiting time:** Finally, we present a greedy algorithm using Procedure 4.1 and Procedure 4.2, where the aim is to schedule the allocated time duration for using the SFs in a way that minimizes the total waiting time of the network. In other words, we want to minimize $\sum_{i=7}^{12}$ (waiting time on SF_i). Algorithm 4.3 illustrates the steps of the scheduler. The algorithm uses the outcomes of Procedure 4.2, *i.e.*, matrix $a[u][v]$ where $v \in \mathcal{N}$ and $u \in \mathcal{F}$. Steps (1) and (2) of Algorithm 4.3 run for all SFs and LNs in the network. Step (3) uses for searching the other LNs of other SFs which can be schedule at the same time. Step (4) calls $Extract\_Min$ function for a SF $i \in \mathcal{F}$ that returns the LN which consists minimum allocated time. Step (4) also uses $Difference\_Min$ function for finding a LN $k \in \mathcal{N}$ which consists minimum difference between the allocated time of i and k. The output of Algorithm 4.3 can be arranged in a matrix $b[u][v][w]$ where a LN v allocates $b[u][v][w]$ time schedule for using u SF at the w^{th} position.

Algorithm 4.3: Scheduling of LNs on SFs.

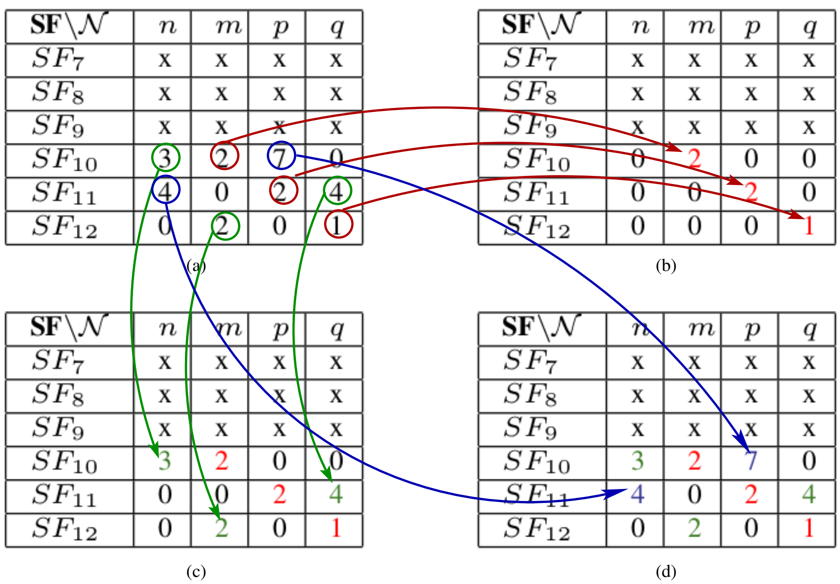
Input: t, allocated time duration matrix $a[][]$, and ζ.
Output: Scheduled allocated time duration matrix $b[][]$.

```
   // Starting from SF_7 to SF_12
   // Set SF ← 12
1  for int i ← 7 to SF − 1 do
       // Allocated LN n ∈ N to SF i ∈ F
2      for int n ← 1 to N do
           // Other SFs k ∈ (F − i)
3          for int k ← i + 1 to SF do
               // Find the minimum difference allocated time LN k
4                  Difference_Min(Extract_Min(i), k);

   Function Extract_Min(i)
   // Return minimum allocated time LN n for SF i
       u = argmin_j {a[i][j]} , 1 ≤ j ≤ N,
       b[i][l_i] = u, a[i][u] = ζ, l_i = l_i + 1;
       return
   Function Difference_Min(u, k)
   // Return minimum time difference k of i
       l = argmin_l {|u − a[k][l]|} , 1 ≤ l ≤ N,
       b[k][l][l_k] = l, a[k][l] = ζ, l_k = l_k + 1
       return
```

Example 5 *We continue Example 4 where each LN finds the optimal subsets of SFs with their time durations as shown in Table 4.2. Algorithm 4.3 runs for* $n, m, p,$ *and* q

LNs. The algorithm schedules SF_{10} to m for 2 time duration because $Extract\_Min(i)$ returns $\min\{n \rightarrow 3, m \rightarrow 2, p \rightarrow 7\}$*. Part (b) of Figure 4.9 illustrates the scheduler outcomes where we enter SF_{10} to m for 2. Next, the algorithm searches other LN which uses SF_{11} and almost equal to 2. The algorithm schedules SF_{11} to p for 2 time duration because $Difference\_Min(u, k)$ returns* $\min\{n \rightarrow |2-4|, p \rightarrow |2-2|, q \rightarrow |2-4|\}$*. The algorithm repeats this step for all LNs and the SFs. Figure 4.9 illustrates the scheduling of SFs for LNs.*



(a)

SF\$\mathcal{N}$	n	m	p	q
SF_7	x	x	x	x
SF_8	x	x	x	x
SF_9	x	x	x	x
SF_{10}	3	2	7	0
SF_{11}	4	0	2	4
SF_{12}	0	2	0	1

(b)

SF\$\mathcal{N}$	n	m	p	q
SF_7	x	x	x	x
SF_8	x	x	x	x
SF_9	x	x	x	x
SF_{10}	0	2	0	0
SF_{11}	0	0	2	0
SF_{12}	0	0	0	1

(c)

SF\$\mathcal{N}$	n	m	p	q
SF_7	x	x	x	x
SF_8	x	x	x	x
SF_9	x	x	x	x
SF_{10}	3	2	0	0
SF_{11}	0	0	2	4
SF_{12}	0	2	0	1

(d)

SF\$\mathcal{N}$	n	m	p	q
SF_7	x	x	x	x
SF_8	x	x	x	x
SF_9	x	x	x	x
SF_{10}	3	2	7	0
SF_{11}	4	0	2	4
SF_{12}	0	2	0	1

Figure 4.9: Illustration of scheduling of SFs for LNs.

Lemma 4.2 *Algorithm 4.3 finds an optimal transmission order within the scope of greedy solution if the LNs are selected in order of ascending allocated time duration.*

Proof: For simplicity, let consider one SF onto which multiple LNs want to transmit data. $P = p_1 p_2 \cdots p_y$ is the sequence of LNs who want to send data to the LG in order of increasing time duration. Let t_i is the time duration of LN p_i in the sequence, then total time in the system taken by SF is

$$T(P) = t_1 + (t_1 + t_2) + (t_1 + t_2 + t_3) + \cdots = \sum_{k=1}^{y} (y - k + 1) t_k.$$

Let assume that 2 LNs with sequence number $\tilde{a}$ and $\tilde{b}$ where $\tilde{a} < \tilde{b}$ and $t_{\tilde{a}} < t_{\tilde{b}}$ in the sequence change their position which makes sequence out of order. We denote this out of order sequence as P'. Then total time in the system taken by SF when it uses P' sequence, is

$$T(P') = (y - \tilde{a} + 1)t_{\tilde{b}} + (y - \tilde{b} + 1)t_{\tilde{a}} + \sum_{k=1, k\neq\tilde{a},\tilde{b}}^{y} (y - k + 1)t_k \tag{4.29}$$

The old sequence is preferable to the new because

$$T(P) - T(P') = (\tilde{b} - \tilde{a})(t_{\tilde{a}} - t_{\tilde{b}}) < 0 \tag{4.30}$$

Hence proved. □

Lemma 4.3 *The time complexity of the proposed scheduling of LoRa nodes is* $O(NM(logM)^2)$.

Proof: The time complexity of the proposed scheduling of LoRa nodes depends on the time taken for running Procedure 4.1, Algorithm 4.1, Algorithm 4.2, and Alogorithm 4.3. Procedure 4.1 requires $O(N)$ times. Let $\mathbf{p}$ and $\mathbf{q}$ be the number of iterations at which NE and SE in the network is calculated, respectively. Since Algorithm 4.1 is being called in Algorithm 4.2, time complexity of Algorithm 4.2 is $\mathbf{pq}$. As LoRa network uses 6 different SFs (SF_7 to SF_{12}), Step 1 of Alogorithm 4.3 requires constant time. Therefore, it does not contribute in the calculation of time complexity. As a result, the time complexity mainly depends on two 'for' loops where the loop at Step 2 depends on the number of LNs connected to LG (*i.e.* N), and loop at Step 3 contains two functions. Both functions compute *argmin* operation using min heap which takes $(NlogN)^2$ time. Let M is the total number of LNs present in the network whereas N is the total number of LNs connected to the LG. Since M is much higher than N, therefore time complexity of Step 3 can be approximated as $O(M(logM)^2)$, where $M \approx N^2$. The total time complexity of the proposed approach is $O(N + \mathbf{pq} + NM(logM)^2)$. As

SG formulated among N LNs, number of iterations required for convergence of Algorithm 4.1 and Algorithm 4.2 is in the order of constant because of the small value of N. Therefore the time complexity of the proposed scheduling of LoRa spreading factors is $O(NM(logM)^2)$ and hence proved. □

4.6 Results and discussions

In this section, we carry out the experimental evaluation to validate the performance of the proposed approach by giving answers to the following questions:

• How does the utilities of LNs and LG change along with the increase in the number of LNs?

• What is the impact of the game parameters on the utilities of LNs and LG?

• What are the waiting time and throughput difference between scheduling and non-scheduling approach?

• How efficiently the proposed approach allocates SFs to increase throughput of the network compare to the existing approach?

4.6.1 Simulation setup

We validate the proposed approach for the performance of the LoRa network using Network Simulator-3 [16]. We have modified the traffic control model (lorawan-tracing-example.cc). Each experiment repeats 100 times and present the average results. Most of the simulation parameters are considered from [18]. The simulation setup of LoRa network mainly consists of end users, multiple LNs, a LG, a network server, and applications. The network is set up by randomly deployed the LNs in a disc-shaped field of radius 1.2 Kilometer. All LNs and LG are configured to use the same bandwidth with 868 MHz channel frequency. To calculate additive noise present in the communication channel, we used noise expression with variance $\sigma^2 = -174 + 10\log_{10} W$ where first term is the receiver sensitivity level and W is the bandwidth [42].

For the performance comparison, we adopted three other existing schemes such as random, distance based, and equal interval based SF allocation. These are the baseline and widely used schemes for the allocation of SFs which motivates to consider them for comparison [14, 18, 52, 53]. In random scheme, allocation of SFs is done by uniform random choice and not depending on the location of the LNs. Because of the randomness, it does not require processing for allocation and easy to implement. The allocation of SFs to the LNs is carried out by using the strength of the received signal (given in RSSI value) at the LoRa Gateway (LG) in distance based scheme. In equal interval scheme, the network is divided into K disjoint annuli of width R/K each, where K is the number of SFs and R is the coverage of the LG. We consider that all LNs have a duty cycle of 1% and does not follow perfect orthogonality in SFs.

4.6.2 Experimental results

4.6.2.1 Utility gain perceived from the proposed model

We study the comparison of network performance based on the different SF allocation schemes. Parts (a) and (b) of Figure 4.10 show the average utility of LNs and utility of LG, respectively, when the total number of LNs in the network increases. Utility is compared for SF allocation based on our proposed game model, random, and distance based schemes. Part (a) of Figure 4.10 illustrates that average utility of the LNs decreases as the number of devices connected to the LG increases, irrespective of the used SF allocation scheme. It also shows that the average utility of LNs is higher when SF is allocated using our proposed game model as compared with other schemes. Difference in utility increases as the number of LNs increase because of the high network interference. In the case of random SF allocation, the performance is initially better than distance based allocation but decreases as the total number of LNs increases. Similarly, part (b) illustrates that the utility of LG initially increases rapidly along with the increase in the number of LNs but later it starts decreasing. This is because initially LG received more data with the increase of LNs but as the number of LNs become larger, all data can not be reached to the LG due to the interference in the network.

• **Observation:** An interesting observation from this result is that random allocation of SFs scheme performs worst when the number of LNs become large. The proposed approach gives the best utility of the network because of the estimated optimal time duration to transmit data from LN to LG.

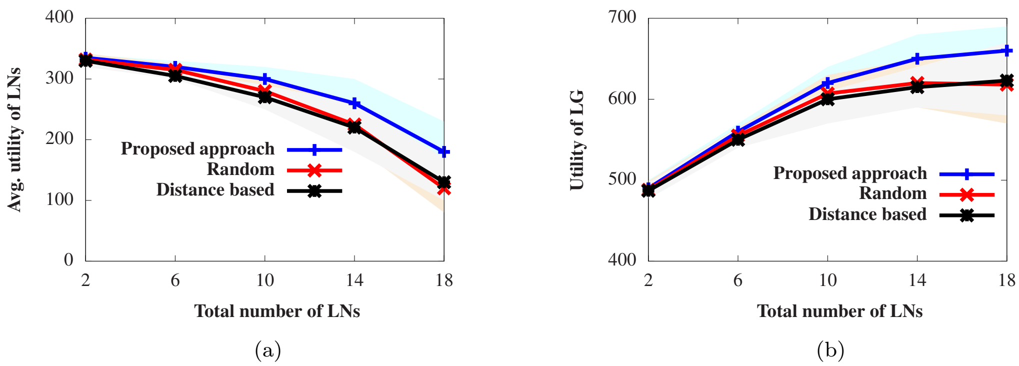


Figure 4.10: Average utility of LNs (part (a)) and LG (part (b)) using proposed, random, and distance based scheme.

4.6.2.2 Analysis of impact of game parameters

Next, we discuss the impact of the game parameters on network performance. Part (a)

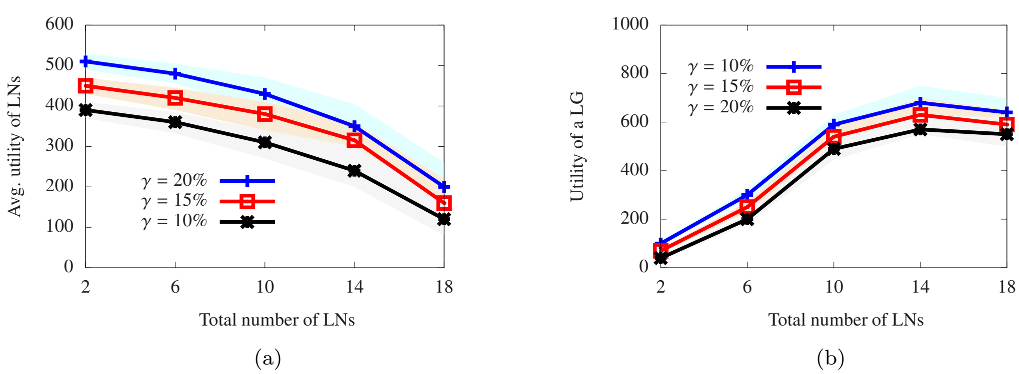


Figure 4.11: Impact of number of LNs on utility of LNs (part (a)) and LG (part (b)) at different of pricing parameter γ.

of Figure 4.11 illustrates that as number of LNs connected to the LG increases, the price set by the LG is distributed among more devices and each device gets less time for data transmission on the allocated SFs. Thus, the utilities of LNs decrease with more number of LNs and with the decrease in γ. The effect on the utility of LG as the number of LNs increase is shown in Part (b). Increase in the number of LNs connected to LG leads to the generation of more data and revenue, thus the utility of LG initially increases. The utility of LG starts decreasing after a certain increase in LNs because of the interference problem among LNs.

Finally, we analyze the impact of precision threshold on the convergence rate of Algorithm 4.1 and Algorithm 4.2 and utilities of LNs and LG, as shown in Table 4.3 and Figure 4.12, respectively. For high precision threshold, algorithms converge in less number of iterations which reduces the time complexity of the algorithms. However, it provides less accuracy of the result. Whereas for low precision threshold, result accuracy is high but it requires more time for convergence. The number of iterations required for convergence increase exponentially with the increase in the number of LNs. Table 4.3 shows the impact of η and ω on the rate of convergence. As value of η increases from 0.1 to 0.4, the required iterations decrease from 13 to 8, as shown in Table 4.3. Similarly, iterations required for convergence reduce from 9 to 7 when ω increases from 2 to 3. Parts (a) and (b) of Figure 4.12 illustrate the average utility of LNs decreases and utility of LG increases with the increase in the number of LNs.

• **Observation:** An interesting observation from this result is that the tuning of the pricing strategy can balance the load on SFs and increase the utility of the network. It can be also observed that more accurate optimal time duration can be find out at the expense of the time complexity which improves the utility of the network.

Table 4.3: Rate of convergence of Algorithm 4.1 and Algorithm 4.2.

		Algorithm 4.1			**Algorithm 4.2**		
		$\eta = 0.1$	$\eta = 0.4$	$\eta = 0.6$	$\omega = 2$	$\omega = 3$	$\omega = 4$
N	**2**	4	2	2	3	2	2
	6	8	5	3	5	4	3
	10	**13**	**8**	6	**9**	**7**	5
	14	24	17	12	14	12	9
	18	38	27	21	22	18	15

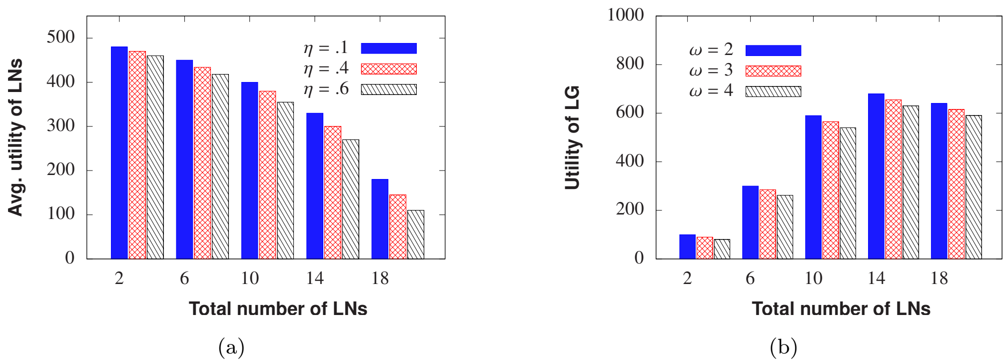


Figure 4.12: Impact of the convergence parameters on the utility of LNs.

4.6.2.3 Impact of scheduling on the network performance

We next present the impact of the proposed scheduling approach (Algorithm 4.3) on the waiting time and throughput of the network. The proper scheduling of the LNs reduces the concurrent accessing of a SF by multiple LNs and therefore it reduces the waiting time of the network. Specially, the propose scheduling helps when large number of LNs are exist in the network as shown in Part (a) of Figure 4.13. The main benefit of the propose scheduling is the increase the network throughput as shown in Part (c) of Figure 4.13. Part (b) of Figure 4.13 illustrates that the waiting time reduces in both cases of scheduling and non-scheduling approach as we increase the number of feasible SFs for a LN. This is because the load on the SFs reduces and more SFs are there for LNs to switch instead of waiting which reduces the waiting time. Part (d) of Figure 4.13 shows that throughput of the network increases with the increase of SFs in the set. In this result, we perform experiments on 10 LNs deployed in the network.

• **Observation:** An interesting observation from this result is that the waiting time of LNs increases as the number of LNs increases in the network but it can be reduced if more SFs present in the combination set of LN.

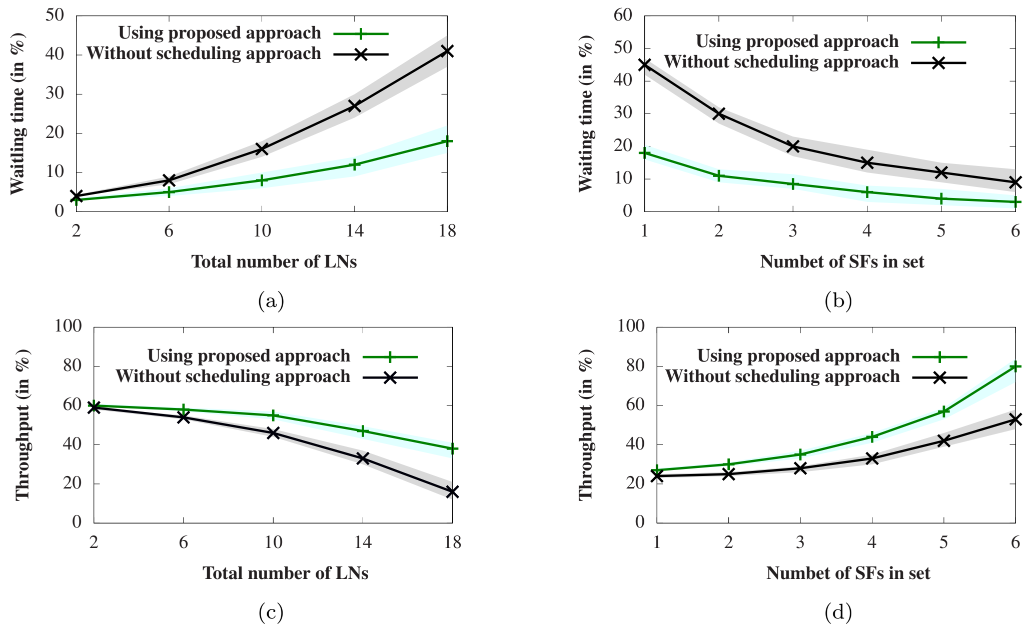


Figure 4.13: Impact of the number of LNs (part (a)) and optimal SFs (part (b)) on the waiting time of SFs. Impact of the number of LNs (part (c)) and optimal SFs (part (d)) on the throughput of the network.

4.6.3 Comparison with existing work

Finally, we compare the proposed approach with existing work which mainly focused on the SF allocation schemes for improving network performance. Authors in [18] compared the performance of random and distance based SF allocation schemes and their results indicate that distance based allocation outperforms random allocation. Equal-interval and area based SF allocation schemes are presented in [14] where equal-interval based allocation provides better performance. We compare our game-theoretic based SF allocation with random, distance, and equal-interval based schemes, as shown in Figure 4.14. Our SF allocation scheme outperforms all the other schemes for any number of LNs. This is because the proposed approach considers interactions among LNs and takes the allocation decision that can maximize the utility of all LNs using game theory. Our proposed approach also performs scheduling on the allocated SFs thus the waiting time of our approach is also minimum, as shown in part (a) of Figure 4.14. It can also be seen from part (b) of Figure 4.14 that throughput of all the allocation schemes has a notable decrease with the increase in the number of LNs because of inter-SF interference.

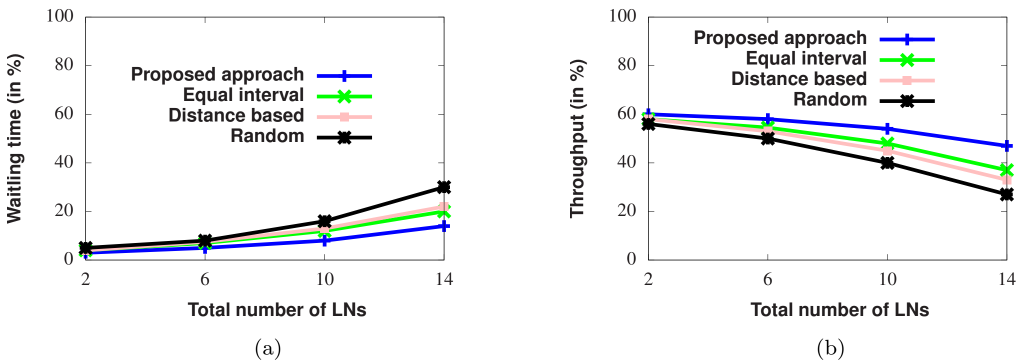


Figure 4.14: Comparison of waiting time (part (a)) and throughput (part (b)) of the proposed approach with [18] and [14].

4.7 Conclusion

In this chapter, we proposed an approach for optimal SF allocation and scheduling the LNs that are connected to the LG. We computed the required transmission time duration of each LN for using the SFs such that the requirement of LNs is satisfied and the network maximizes its utility. We used a game-theory approach for computing the time duration of LNs on suitable SFs. Nash equilibrium is obtained among the connected LNs at which all devices choose their optimal transmission time duration. The network reaches at a Stackelberg equilibrium when a LG (leader) estimates optimal price for maximizing its utility while the LNs (followers) select their best response time duration for forwarding data. The obtained optimal time duration of LNs are then scheduled to minimize the waiting time. The proposed approach is validated by simulating the LoRa network. We conclude from the results that the proper scheduling of the LNs improves the throughput of the LoRa network.