Robust Gateway Placement for Scalable LoRaWAN

Frank Loh, Dominique Bau, Johannes Zink, Alexander Wolff, Tobias Hoßfeld
Institut für Informatik, Universität Würzburg, Würzburg, Germany
{firstname.lastname}@uni-wuerzburg.de

***Abstract*—Many exciting applications are currently emerging due to the use of technology from the Internet of Things (IoT) in Smart Cities. In recent years, Long Range Wide Area Networks (LoRaWANs) have become one of the main technologies in this field due to long distance coverage, low power consumption sensors, and economic reasons. So far, LoRaWANs have been planned and deployed mainly with the aim of minimizing the number of gateways, while still providing coverage in a geographic area. Packet collisions and future traffic have not been considered. In contrast, we present a gateway placement strategy that is robust against an increase in the number of sensors and in the network load. To this end, we modify an existing local search algorithm for a geometric set cover problem such that it takes the capacities of gateways into account. If we set these capacities below the maximum, the network can accommodate additional sensors in the future. Moreover, our algorithm can extend the initial placement once the capacities are exhausted.**

I. Introduction

Weather, climate, traffic, and pollution monitoring as well as smart water metering [1] are only a small subset of application areas that currently emerge given the Internet of Things (IoT) and Smart Cities. Although some deployments have high bandwidth or very low delay requirements, a large number of applications require cheap sensors and energy-efficient communication. There, Long Range Wide Area Networks (LoRaWANs) have proven to be viable [2].

Nowadays, a LoRaWAN provider sets up the IoT gateway infrastructure with coverage and reasonable Quality of Service (QoS) for all sensor nodes. As usual with new infrastructure, novel use cases will emerge and will be used, which in turn creates more traffic. As a consequence, there is an expected, but unknown increase of IoT traffic in such networks. This is a key problem which is, to the best of our knowledge, not taken into account by existing gateway placement algorithms, which mainly aim at minimizing the number of required gateways for given network traffic patterns and loads [3].

In this work, we address the IoT gateway placement problem as a geometric set cover problem where we are given a set of gateway candidates (disks) and a set of sensors (points). Each gateway has a *capacity* – the maximum number of sensors that can be assigned to it. The task is to select a minimum-size subset of the gateway candidates such that each sensor can be assigned to a gateway that covers it and no gateway capacities are violated. We solve this problem by modifying an existing local search heuristic for the uncapacitated version (which is already NP-hard). For small problem instances, we compare the solutions of this algorithm to exact solutions computed using an integer linear program (ILP). By limiting gateway ranges and capacities, we reserve gateway capacity towards a future network load or sensor increase. Based on the resulting placements, we study different LoRaWAN network configurations and sensor assignments to compute the packet collision probability for different placements. Finally, we emulate future network scenarios with increasing sensor numbers, study their influence on our placement, and provide ways to extend our placement.

Our contribution is threefold. First, we present a novel approach for IoT gateway placement by solving a capacitated geometric set cover problem in reasonable time for realistic Smart City scenarios. Second, we present a methodology to study the QoS for a LoRaWAN based on a collision probability investigation. Third, by varying input parameters for the placement algorithm, we show how to compute gateway placements that are robust against load increase and how to extend an existing placement when the capacities are exhausted.

The remainder of this work is structured as follows. We first review background and related work concerning gateway placement (Section II). Then, we introduce our methodology (Section III). Next, we present our scenarios (Section IV) and evaluate them (Section V). Finally, we conclude (Section VI).

II. Background and Related Work

We summarize fundamental background required to understand the gateway placement approach. Afterwards, we outline related work on LoRaWAN and gateway placements therein.

A. Gateway Placement

Gateway placement is a geometric variant of the well-known combinatorial optimization problem Set Cover. In Set Cover, one is given a ground set U (our sensors) and a family $\mathcal{S}$ of subsets of U (for each gateway, the set of sensors that can communicate with the gateway), and the task is to find a smallest *cover*, that is, a subset $\mathcal{C}$ of $\mathcal{S}$ such that $\bigcup_{S \in \mathcal{C}} S = U$. Clearly, a cover can only exist if $\mathcal{S}$ itself covers U. It is well-known that Set Cover is NP-hard and that a simple greedy approach (that repeatedly selects the set that covers the largest number of uncovered elements) yields an $(\ln n)$-approximation [4], which is essentially best possible.

We assume that sensors and gateways are points in the plane and that a gateway can communicate with all sensors within a certain distance r. In this variant of Geometric Set Cover, a subset of the (gateway) disks must cover all (sensor) points. This is equivalent to the Hitting Set problem where the sensors correspond to disks of radius r, and the task is to find the smallest *hitting set*, that is, the smallest set of points

ISBN 978-3-903176-42-3© 2021 IFIP

in the plane such that every disk contains at least one of the points. Many geometric variants of set cover and hitting set have been studied. Since geometry restricts the ways to cover the ground set, usually better approximations exist for these variants. E.g., using so-called epsilon-nets yields constant-factor approximations [5]–[7]. Mustafa and Ray [3] presented a simple local search algorithm with an additional parameter k. For every $k > 1$, their algorithm computes a $(1 + b/\sqrt{k})$-approximation in $n^{O(k)}$ time, where b is a constant. Their iterative local search algorithm, which we denote by GeometricLocalSearch, works as follows. Initially, set $\mathcal{C} = \mathcal{S}$, i.e., put all (gateway) disks into the cover. While there is a set of k (gateway) disks from $\mathcal{C}$ that can be replaced by $k - 1$ (gateway) disks from $\mathcal{S}$ such that all (sensor) points are still covered by $\mathcal{C}$, reduce the size of $\mathcal{C}$ by at least one by applying this replacement. We study a capacitated version of GEOMETRIC SET COVER where each gateway may only communicate with the ℓ *closest* sensors in its range. We call this problem VORONOI COVER. We solve this new problem using the local search algorithm of Mustafa and Ray because it can easily be adapted to our setting (though we do not prove that the algorithm keeps its approximation guarantee). This adapted version, which we call VoronoiLocalSearch, is described below.

B. Related Work on LoRaWAN and Gateway Placement

Adelantado et al. [8] thoroughly investigate the limitations of LoRaWANs and consider the network behavior for e.g., different spreading factors and payload sizes. Bankov et al. [9] discuss the limits of LoRaWAN channel access. Ferré [10] presents a theoretical study for collisions and packet loss at a LoRaWAN gateway and validates the study with a simulation. Others do large-scale measurements to investigate LoRaWANs. For example, Liu et al. [11] describe a real-world LoRaWAN deployment in Shanghai with 66,000 sensors over an area of $140\,\mathrm{km}^2$. Their study includes the examination of packet loss characteristics and sources. A large scale real-world experimental evaluation is also done for Bangkok in [12]. Several gateway placement approaches focus on coverage and cost reduction. Recent works among those are based on k-means, c-means [13] or other clustering approaches [14]. Mnguni [15] surveys the area. Ousat et al. [16] present an approach based on mixed-integer non-linear optimization, which can be used only for small networks. In contrast to related work, our approach focuses on robustness against increasing load. Our main target is not primarily to reduce the number of placed gateways but reducing the collision probability by an in-depth analysis of the received placement and the placement parameters. With different parameters and different sensor settings and scenarios, we focus on future increase in the number of sensors in the network and thus, the load. How to scale a LoRaWAN has, to the best of our knowledge, not been studied so far.

III. METHODOLOGY

In this section, we present our approach for a robust gateway placement for LoRaWANs and the methodology to receive packet collision probabilities of the complete network.

A. Overview

Our methodology consists of three steps summarized in Table I. The gateway placement is the first step, where we place our gateways based on an input of sensors, possible gateway locations, a transmission range and a sensor limit for gateways. With this placement and additional LoRa transmission parameters, we set up a network configuration and map each sensor to the closest gateway in step two. In step three, we calculate the packet collision probability for each sensor and the complete network. In the following we start with details about our gateway placement approach.

B. Gateway Placement

1) Problem: Our goal is to model the problem of selecting gateway positions for a LoRaWAN as a GEOMETRIC SET COVER problem where the sensors are points that need to be covered and the gateways are unit disks whose radius r equals the sending range of a gateway. However, note that this model is not suitable ad-hoc because in GEOMETRIC SET COVER, a disk can cover an arbitrary number of points, whereas in a LoRaWAN each gateway has a maximum number of sensors it can serve reliably.

Consequently, we generalize GEOMETRIC SET COVER for unit disks to what we call VORONOI COVER, where we are given, besides (sensor) points P and (gateway) unit disks D in the plane, an additional parameter ℓ specifying the limit of (sensor) points being *assigned* to a (gateway) disk in the cover $C \subseteq D$. We say a (sensor) point p is *assigned* to a (gateway) disk $c \in C$ if for all $c' \in C \setminus \{c\}\colon d(p, c) < d(p, c')$, where d is the Euclidean distance function, in which (gateway) disks are represented by their center points. We assume that, for any given point in such an instance, all distances to the disk centers are pairwise different. This is in line from what we would expect from real world applications. In other words, p is assigned to c if p lies in the Voronoi cell of c in the Voronoi diagram constructed by the centers of C. Each Voronoi cell is allowed to host at most ℓ points. For an example, see Figure 1. Observe that GEOMETRIC SET COVER for unit disks is the special case of the VORONOI COVER when $\ell = \infty$. VORONOI COVER is similar to CAPACITATED DISK COVER [17], [18], which is also GEOMETRIC SET COVER with a limit of (sensor) points per (gateway) disk, but there, (sensor) points do not need to be assigned to the closest (gateway) disk and the algorithm is supposed to determine such an assignment beside selecting (gateway) disks.

2) Algorithms: Next, we describe our new method VoronoiLocalSearch built upon GeometricLocalSearch. We proceed in the same way, but when we try to replace k (gateway) disks by $k-1$ (gateway) disks, we accept only if for all (gateway) disks, the number of (sensor) points assigned to each of them does not exceed the capacity ℓ. Note that, during this replacement, not only the $k - 1$ new disks need to be checked, but also the disks in the cover around, which may receive points that were previously assigned to one of the k disks taken out from the cover. Moreover, we apply several speed-up techniques to make this originally theoretical algorithm considerably faster and usable for practical applications. Most

TABLE I: Overview over the methodology of our 3-step approach

#	Input	Methodology step	Output	Sect.
1	sensor and possible gateway locations, range r for gateways, sensor limit ℓ per gateway	gateway placement	sensor and gateway positions	III-B
2	sensor and gateway positions, LoRa transmission parameters	network configuration and sensor mapping	for each sensor: used SF, best gateway, number sensors in collision radius with SF	III-C
3	for each sensor: sensors in collision radius with SF, LoRa packet parameters	packet collision calculation	collision probability for each sensor and within complete network	III-D

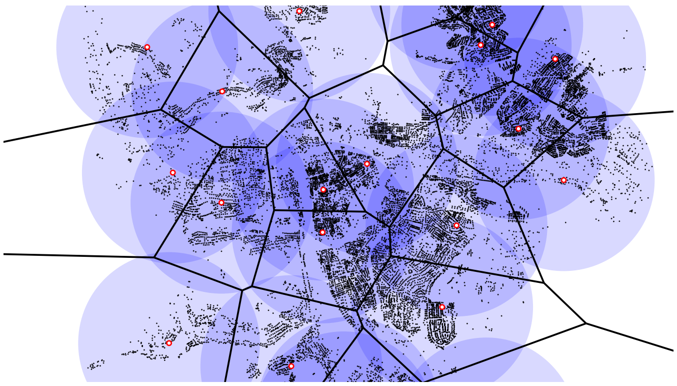

Fig. 1: VORONOI COVER (clipped) for Würzburg, Germany, with 28,000 sensors (1 per building) using 25 gateways. Each gateway defines a Voronoi cell; all sensors in this cell are assigned to that gateway. The blue disks indicate the gateway range (1.5 km). We used a capacity of $\ell = 2{,}000$ per gateway; some of the central cells come close to this limit.

notably, we start by removing unnecessary (gateway) disks ($k = 1$); then we increment k. In our experience, going up to $k = 2$ already yields good results and is still relatively fast, whereas $k = 3$ yields only a rather small improvement at the cost of a much higher running time. In practice, incrementing k beyond 3 does not make sense. This is confirmed by a small pre-study (see below). Moreover, we use a grid depending on the radius r of the (gateway) disks since a set of disks can only be replaced by disks in the neighborhood. With this easy exploitation of the locality, we can save a significant amount of time. Beside this, we randomize the order of disks we check and we mark areas that are not affected by changes to save re-computation time. We have implemented our algorithm in Java which will be published after acceptance.

For comparison reasons, we have also implemented the integer linear program (ILP) for VoronoiLocalSearch. For each (gateway) disk c and for each pair of a (sensor) point p and a (gateway) disk c, we have a variable corresponding to:

$$x_c = 1 \Leftrightarrow c \text{ is in the cover}$$
$$x_{p,c} = 1 \Leftrightarrow p \text{ is assigned to } c$$

We will also use the ordered set D_p, which contains all (gateway) disks that a (sensor) point p lies in and is sorted increasingly by distance from p (the ordering is denoted by $\prec_p$), and the set P_c, which contains all (sensor) points that lie within a (gateway) disk c. The objective is to minimize $\sum_{c \in D} x_c$ subject to the following constraints.

– Each sensor is assigned to at least one gateway:
$$\sum_{c \in D_p} x_{p,c} \geq 1 \qquad \text{for } p \in P$$
– At most ℓ sensor are assigned to each gateway:
$$\sum_{p \in P_c} x_{p,c} \leq \ell \qquad \text{for } c \in D$$
– Sensors can only be assigned to gateways in the cover:
$$x_{p,c} \leq x_c \qquad \text{for } p \in P, c \in D_p$$
– Each sensor p is assigned to the closest gateway in the cover:
$$\sum_{c' \prec_p c} x_{c'} \leq (1 - x_{p,c}) \cdot |D_p| \qquad \text{for } p \in P, c \in D_p$$
– Gateway c is either in the cover ($x_c = 1$) or it isn't.
$$x_c \in \{0, 1\} \qquad \text{for } c \in D$$
– Sensor p is assigned to gateway c ($x_{p,c} = 1$) or it isn't.
$$x_{p,c} \in \{0, 1\} \qquad \text{for } p \in P, c \in D_p$$

To solve the ILP, we have used IBM CPLEX, a commercial state-of-the-art ILP solver. An ILP solution for VORONOI COVER minimizes the number of chosen disks.

C. LoRaWAN Configuration and Sensor to Gateway Mapping

In this section, we introduce LoRa related parameters required for LoRaWAN configuration and determine the possible transmission distance of a sensor in a LoRaWAN.

1) LoRa: LoRa is a low-power wide-area network modulation technique based on the chirp spread spectrum. It uses license-free radio frequencies as from 867 MHz to 869 MHz in Europe and 125 kHz bandwidth (BW) for uplink. In the following, we introduce several other variable parameters that influence the transmission behavior in LoRaWAN.

Spreading Factor: The Spreading Factor (SF) in LoRa defines the number of raw bits a symbol carries. Packets transmitted with higher spreading factors can be transmitted over longer distances and are more robust against interference at the cost of increasing duration to transmit a single symbol. This so called symbol duration is $T_s = 2^{SF}/BW$ and influences the Time on Air (ToA) of a LoRa packet.

Time on Air: The ToA of a LoRa packet is

$$T_{\text{packet}} = (n_{\text{preamble}} + 4.25) \cdot T_s + (n_{\text{ps}} \cdot T_s), \qquad (1)$$

where the first term is the preamble duration. The payload duration is $(n_{\text{ps}} \cdot T_s)$ with n_{ps} as number of payload symbols

$$n_{\text{ps}} = 8 + \max(\lceil n_{\text{payload}} \rceil \cdot (CR + 4), 0)$$

and

$$n_{\text{payload}} = \frac{8\,PL - 4\,SF + 28 + 16\,CRC - 20\,IH}{4\,SF - 8\,DE}.$$

The parameters used above are detailed in Table II.

TABLE II: Parameter overview

Parameters	Variable	Value
gateway height	h_{T}	5 m
sensor height	h_{R}	4.5 m
bandwidth	BW	125 kHz
coding rate	CR	4
payload	PL	variable; based on scenario
cyclic redundancy check	CRC	1
enabled or disabled header	IH	1
low datarate optimize	DE	0
number of preamble symbols	n_{preamble}	8

Hata Path Loss Model: The Hata model is especially developed to determine the transmission path loss within radio access networks in urban environments. For that reason, the urban version according to [19] is used here. The path loss P_{loss} is calculated according to

$$\begin{aligned} P_{\mathrm{loss}} = 69.55 + 26.16\log_{10}(f) - 13.82\log_{10}(h_{\mathrm{R}}) \\ -a(h_{\mathrm{T}}) + [44.9 - 6.55\log_{10}(h_{\mathrm{R}})]\log_{10}(d) \end{aligned} \quad (2)$$

with

$$a(h_{\mathrm{T}}) = 3.2\left[\log_{10}(11.75 \cdot h_{\mathrm{T}})\right]^2 - 4.97$$

and f as the radio frequency of $867\,\mathrm{MHz}$ for LoRa, h_{R} as receiver height in meters, h_{T} as transmitter height in meters, and d as distance between both in meters.

2) Sensor to Gateway Mapping: Next, we describe the sensor to gateway mapping process like it is done in LoRaWAN. This is required to receive information about interferences between transmitting sensors.

Transmission Distance: First, we calculate the maximal possible transmission distance for each LoRa sensor. The goal for each sensor in a LoRaWAN is to transmit to the closest gateway to transmit with the lowest possible SF and save ToA. For that reason, we first calculate the distance to the closest gateway for each sensor based on the gateway and sensor locations and the sensor assignment received from our algorithm VoronoiLocalSearch. Afterwards, we calculate the possible transmission distance for each sensor which is influenced by several factors, among others, the used path loss model with the geography the network is located in, the sensor- and gateway height, the transmission power and the used SF for transmission. All required parameters we use for a LoRa transmission in this work are summarized with the respective value in Table II. For this study, we use a bandwidth of $125\,\mathrm{kHz}$. The sensor transmission power Tx is set to $12\,\mathrm{dBm}$ stated as realistic parameter in [20]. The RSSI tolerance for different SFs is received by adding the transmission power to the radio frequency sensitivity from literature [21]. By activating Equation 2 towards the distance, the maximal transmission distance for different spreading factors and RSSI values is received. The result is shown in Table III. This means, a sensor transmitting with one specific SF from the table can connect to gateways or interfere other sensors in that specific range. With the distance to the closest gateway calculated before, we receive a minimal SF for each sensor that is used for transmitting packets. Additionally, for a sensor x using SF_x for transmission, all sensors k within its transmission radius r_x are interfered by x. Thus, we log for all sensors k that they are interfered by x with SF_x. The same is done for the complete transmission path for sensor x to its closest gateway to make sure interference takes place between sensor and gateway. In this way, we iterate over all sensors x in the complete network to receive all possible interferences for each sensor with the respective SF. In the following, we use this information for collision probability calculation.

TABLE III: Hata model results

Spreading factor	RSSI tolerance [dBm]	Distance [m]	Spreading factor	RSSI tolerance [dBm]	Distance [m]
7	-135	1175	10	-144	1964
8	-138	1394	11	-145	2079
9	-141	1655	12	-148	2468

D. Collision Probability Calculation

In LoRaWAN, an ALOHA-like random channel access mechanisms is used which result into packet collisions. Since this is an essential QoS parameter the last step in our methodology is the calculation of packet collision probabilities for all sensors and the complete network. For simplicity reasons, we only study a single LoRaWAN channel while the collision probability determination for multi-channel investigations is left as future work. Two packets p_1 and p_2 collide in a LoRaWAN channel when the transmission interval I_1 of p_1 and I_2 of p_2 defined by transmission start t_0 and transmission end t_1 overlap. For each sensor x in the network we receive a number of sensors k_x with the respective SFs potentially interfering with x from step 2. We set the transmission rate to one packet per hour. This is valid since changing the packet transmission rate per sensor has the same influence on the packet transmission rate in the network and thus the collision probability as changing the number of sensors in the network and vice versa. With this input, we calculate the transmission start times for x and all k_x as uniform random numbers between $0\,\mathrm{s}$ and $3600\,\mathrm{s}$. This is valid for a sufficiently large number of sensors according to Metzger [22]. We calculate the ToA T_{packet} for each packet with the received SF from step 2 according to Equation 1. An overview of the used parameter values is presented in Table II. The transmission end time for each packet is received by $t_1 = t_0 + T_{\mathrm{packet}}$. If the transmission interval I_x of sensor x overlaps with any transmission interval I_k of any k_x in the transmission range, x collides. Because of the randomness of transmission start times t_0, we repeat this 10,000 times to study the collision probability for packets for each individual sensor and 10 times to study the overall packet collision probability within the network.

IV. Scenarios

To simulate realistic scenarios, we derive the positions of gateways and sensors from real-world data. More precisely, we use the centroids of about 28,000 buildings of the city of Würzburg, Germany, from OpenStreetMap as a base set of geographic coordinates. For the potential gateway positions, we use the grid points of a grid of cell side length $2r/\sqrt{2}$ and $20\,\%$ of the sensor positions uniformly at random. The former guarantee that we can reach all sensors even in sparse areas, while the latter provide enough candidate positions for dense areas, where the capacity ℓ of a single gateway would quickly be exceeded.

A. Scenario Definition

To study the applicability, the performance, and the robustness regarding future proof deployment, we study two different scenario sets. First, we define a *pre-study scenario* set to investigate the choice of the parameter k for our algorithm VoronoiLocalSearch, the randomness of the presented algorithm and the randomness of results for different sensor input. An overview of all pre-study scenarios is presented in Table IV. For all scenarios, we take a random subset of size 2,800 as sensor positions. This number is a good trade-off between a sufficiently large network and enough potential for scaling up and out. Note that our algorithm VoronoiLocalSearch also works for larger deployments in larger cities. But then, due to exceeding runtimes of the ILP, no comparison to an optimal solution is possible. The results of the pre-study lead us to answer the question how many runs are required for each scenario by evaluating each run of each algorithm by its runtime, the number of selected gateways, and the actual arrangement of the selected gateways for usage in the *robustness-study*. Then, our robustness-study contains three steps: (1) the initial placement study with input parameters received in the pre-study, (2) the load increase to study robustness towards future load and sensor increase, and (3) the extension study where we add additional gateways with our algorithm to avoid packet collision probability peaks.

B. Pre-Study for Algorithm Configuration

Parameter Choice for the Local Search Algorithm: One input parameter for the placement algorithm is an integer k, used in the VoronoiLocalSearch to specify up to which number of disks the algorithm tries to replace k by $k-1$ disks. To study this parameter, we fix all parameters of our algorithm ($r = 1.5\,\mathrm{km}$, $\ell = 500$) with a random input set of 2,800 sensor positions, and the set of gateway position candidates. For each of $k = 1, 2, 3, 4$, we perform 10 runs with an algorithm timeout limit of $30\,\mathrm{min}$. For $k = 3$ this timeout has been reached in 7 runs and for $k = 4$ in all 10 runs. The solution found until this point is taken. The average time to run the test for $k = 1$ is $0.32\,\mathrm{s}$ and the output solution consists of 20.5 (gateway) disks, for $k = 2$ it is $1.91\,\mathrm{s}$ with 17.8 (gateway) disks and for $k = 3$, it is $1690.21\,\mathrm{s}$ for 16.8 (gateway) disks. The additional time used for $k = 4$ in our tests did not yield any improvement compared to $k = 3$. Since we see the best trade-off between rumtime and quality for $k = 2$, we use this value in the following.

Randomness Algorithm: Since VoronoiLocalSearch use randomness internally, we cannot expect the results to be the same every time. For that reason, in the *randomness algorithm* scenario, we fix all parameters of our algorithms ($r = 1.5\,\mathrm{km}$, $\ell = 500$), the set of 2,800 sensor positions, and the set of

TABLE IV: Pre-study overview

Study	Study goal	Study result
Choice of k	receive best value of k for the algorithm: trade-off runtime and result	$k = 2$ as best value
Randomness of the algorithm	receive suitable number of reruns by quantification of algorithm randomness	10 reruns per scenario
Randomness of the input	study influence of random sensor position selection	randomness of algorithm dominates randomness of sensor selection
Algorithm input	get meaningful parameter for sensor limit ℓ and gateway range r as algorithm input	$r \in \{1.5, 2.0, 2.5, 3.0\}$ km $\ell \in \{300, 400, \ldots, 1000\}$ $\cup\{\infty\}$

gateway position candidates. We do 100 runs of the algorithm to analyze the variance of the results among different runs of the same scenario, with regards to number of gateways per placement, and number of sensors interfering each other according to step 2 of our methodology. Based on the result, we calculate the approximation error to receive a suitable number of reruns for our algorithm for later experiments. By using a maximal approximation error of $5\,\%$ and 1,000 reruns, we receive 5-6 runs with regard to the number of sensors and 5-8 runs with regard to the number of gateways. For that reason, we use 10 reruns for the following scenarios.

Randomness Input: In the next scenario, we fix all parameters of our algorithms ($r = 1.5\,\mathrm{km}$, $\ell = 500$), and use a different set of 2,800 sensor positions chosen uniformly at random from the larger base set of 28,000 building coordinates for each run. We do 10 runs with the VoronoiLocalSearch to receive different gateway placements. For each run, we use 10 different random sensor sets and compare the number of sensors interfering each other according to step 2 of our methodology. The purpose of this test is to analyze how much the actual choice of sensor positions affects the results of the algorithms or if the randomness of the gateway placement algorithm dominates the random sensor selection.

For a single run we receive a minimal average number of potentially interfering sensors of 155, the maximum is 182. When we calculate the average amount of potentially interfering sensors for each individual placement over the 10 random sensor selections we receive values between 160 to 177 of potentially interfering sensors and a standard deviation of 3.0. Comparing different placements, we receive a standard deviation of 6.2. Thus, the randomness of the algorithm dominates the randomness of the chosen sensor positions.

Algorithm Input: In the last pre-study step, we aim on finding meaningful values for the placement algorithm input parameters r as the range a gateway covers and ℓ as the sensor limit per gateway. In this test, we create in total 66 different scenarios by testing all possible parameter combinations:

- range r of gateways: 0.5, 1.0, …, $3.0\,\mathrm{km}$,
- sensor limit ℓ per gateway: 100, 200, …, 1,000, and ∞.

We have one run per scenario to receive an overall insight in the behavior for different parameter settings. We use a different random sets of 2,800 sensor positions for each run of each scenario. This study is designed to limit the parameter space for the following placement study. Thus, the influence of algorithm randomness does not influence the overall statement. In the tests, we see that r smaller than $1.5\,\mathrm{km}$ is too small for a good placement since according to Table III a transmission distance of less than $1.5\,\mathrm{km}$ only uses SF7 and SF8 which limits the potential for network scaling. Furthermore, the received gateway density is too high for a realistic deployment. The results show that on average up to 24 gateways are in the range for each sensor for $0.5\,\mathrm{km}$ distance and 7 to 12 gateways for each sensor with $1.0\,\mathrm{km}$ distance. Furthermore, we see that sensor limits per gateway of less than 300 show the same behavior. There, on average 10 gateways are available for each sensor for $\ell = 100$ and 5 gateways for $\ell = 200$. Based on these preliminary studies, we study gateway ranges $r = 1.5\,\mathrm{km}, 2.0\,\mathrm{km}, 2.5\,\mathrm{km}, 3.0\,\mathrm{km}$ and sensor limits $\ell = 300$ to 1000, step width 100 and $\ell = \infty$ in the following.

C. *Placement Study*

The placement study is our main investigation and consists of three parts: initial placement study, future robustness study, and additional placement study. Here, we sketch these steps and in the next section, we describe their results.

Initial Placement Study: Here, we discuss the influence of the different input parameters of our placement algorithm on the placement and thus, the collision probability. Therefore, we study all combinations of the gateway range r and the sensor limit ℓ we received in the pre-study. We always use a random set of 2,800 sensor positions and do 10 repetitions of each configuration. Based on the presented collision calculation III-D, we evaluate the placement algorithm input parameter towards an optimal solution. To solely focus on this goal, we use $1\,\mathrm{B}$ payload. This guarantees that our evaluations hold true for the smallest possible LoRa payload size while larger payloads are studied in the future robustness study afterwards.

Future Robustness Study: We define future robustness in this work by robustness of the gateway placement against increasing collision probability in increasing load situations. In a LoRaWAN, the load is defined by the number of sent packets, and thus number of sensors, and the packet size. In this study, we increase the number of sensors in the network and study the influence on the collision probability. Furthermore, we study the influence of different packet payload sizes of $1\,\mathrm{B}$, $4\,\mathrm{B}$, $8\,\mathrm{B}$, $16\,\mathrm{B}$, and $32\,\mathrm{B}$ on the network.

Additional Placement Study: At the end, in the additional placement study, we show the performance of our placement algorithm to place additional gateways in overload situations within an existing network. We use the existing placement, define new algorithm input parameters and recalculate the overall packet loss in the network after the additional placement.

V. Evaluation

First, we present our evaluation of the placement study. Then, we discuss our placement strategy concerning robustness and compare it with an optimal placement with the goal of minimizing the number of gateways received by our ILP.

A. *Initial Placement Study*

Our algorithm VoronoiLocalSearch has two essential variable parameters. The gateway range r and the sensor limit ℓ. In the following, we study the influence of both parameters on the collision probability starting with the gateway range r.

Gateway Range: In this study, we test gateway ranges of $1.5\,\mathrm{km}$, $2.0\,\mathrm{km}$, $2.5\,\mathrm{km}$, and $3.0\,\mathrm{km}$. Figure 2 shows the results based on the range, independent on the sensor limit as Empirical Cumulative Distribution Function (ECDF) with the average collision probability for the whole network on the x-axis. The figure shows that the collision probability increases with larger gateway ranges since then, a higher SF is used for transmission according to Table III. In this case, each sensor interferes other sensors in a larger radius. Additionally, we see this influence in the maximal packet collision probability for single sensors. For $r = 1.5\,\mathrm{km}$, the highest number of sensors in the collision range of a single sensor is 509 while none of these sensors is transmitting its data with a SF larger than 9. Thus, we see a collision probability of 0.78 % for $1\,\mathrm{B}$ payload. Here, the average value over all sensors and runs is 0.30 %. For $r = 3.0\,\mathrm{km}$, a maximum of 853 other sensors are in the interference range of one sensor, and 107 transmit data with SF 10, 49 with SF 11, and 171 with SF 12. There, the collision probability for $1\,\mathrm{B}$ payload is 12.56 %. The overall average for $r = 3.0\,\mathrm{km}$ is 2.02 %. Regarding the total number of placed gateways, we see that less gateways must be placed for larger gateway ranges r with on average 17.22 gateways for the scenarios with $r = 1.5\,\mathrm{km}$, 11.83 gateways on average for $r = 2.0\,\mathrm{km}$, 9.33 gateways for $r = 2.5\,\mathrm{km}$, and only 8.12 gateways for $r = 3.0\,\mathrm{km}$. Thus, we show that it is not essential to minimize the number of gateways by maximizing the transmission distances in LoRaWAN to decrease collision probability but set up a placement to minimize the SF and thus the ToA for each transmission. While the pre-study shows that too small distances are not meaningful, leave no space for future scaling, or even overload the network with gateways, gateway distances of $1.5\,\mathrm{km}$ and $2.0\,\mathrm{km}$ are useful for our algorithm. Additionally, only for $r = 1.5\,\mathrm{km}$ and $r = 2.0\,\mathrm{km}$, all sensors are covered based on the Hata transmission model. Thus, we only consider $r = 1.5\,\mathrm{km}$ and $r = 2.0\,\mathrm{km}$ in detail.

Sensor Limit: The second input parameter for our gateway placement is the sensor limit ℓ. Since we see only little variance in the overall collision probability for $r = 1.5\,\mathrm{km}$, we study the collision probability based on the sensor limit ℓ on the x-axis for $r = 2.0\,\mathrm{km}$ in Figure 3. We see a slightly smaller collision probability for a sensor limit of 300 up to 500 compared to larger limits. The smallest median collision probability is received by $\ell = 500$ with 0.29 %. For higher sensor limits, no tendency towards increasing or decreasing probabilities is possible anymore. Furthermore, sensor limits of more than 500 were never reached or exceeded during the gateway placement. For $r = 2.0\,\mathrm{km}$, the largest number of sensors connected to a gateway is 401, thus much larger sensor limits are never relevant for initial placement calculation in our scenario. In addition, we see that the gateway range r is more important for the placement compared to the sensor limit.

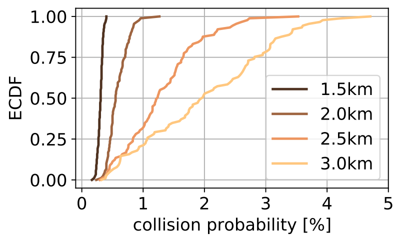

Fig. 2: collision probab. based on gateway range for all sensor limits

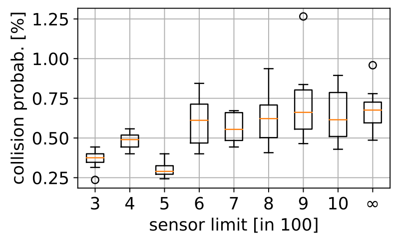

Fig. 3: collision probab. based on sensor limit for $r = 2.0\,\mathrm{km}$

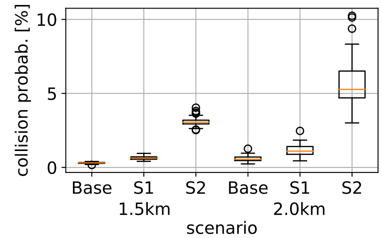

Fig. 4: collision probab. for different load increase scenarios for all sensor limits

B. Future Robustness Study

Based on the result of the placement study, our future robustness investigation for ranges of $1.5\,\mathrm{km}$ and $2.0\,\mathrm{km}$ and all sensor limits studies the influence of load increase on the collision probability.

Increased Number Sensors: To study future robustness, we increase the load with more sensors in our network. Therefore, we define three increase steps: in increase step S1, we double the number of sensors in the network to 5,600 randomly selected sensors from our sensor pool. This is 20 % of all available sensors in our pool. In the second increase step S2, we use 14,000 sensors which is 50 % of all available sensors and in increase step S3, we use all 28,000 sensors. In Figure 4, we present the influence of the load increase on the collision probability for increase step S1 and S2 compared to no load increase (denoted as base) for gateway range limits $r = 1.5\,\mathrm{km}$ and $r = 2.0\,\mathrm{km}$ as boxplot. The figure shows that the load increase influences the collision probabilities for both gateway range limits. For $r = 1.5\,\mathrm{km}$, starting at a lower level with 0.30 % mean collision probability over all scenarios with the baseline, the collision probability for S1 is approximately doubled with 0.63 %. For S2, a much higher increase to an average collision probability of 3.01 % is visible. Furthermore, we see only a little variance in collision probability for the scenarios for each $r = 1.5\,\mathrm{km}$ result and only some outliers for S2. In contrast, for $r = 2.0\,\mathrm{km}$, an increase from 0.58 % for the baseline to 1.15 % for S1, and 5.61 % for S2 is received and a much higher increase of the variance, especially for S2. There, the collision probability is between 3.01 % and 10.25 %. Thus, we see that a range limit of $r = 1.5\,\mathrm{km}$ is better towards future scalability. This is, based on the Hata model for path loss, in the range of a transmission with SF 9.

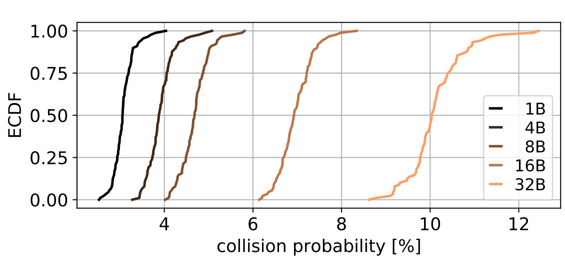

Fig. 5: collision probability for S2 based on payload size and $r = 1.5\,\mathrm{km}$ for all sensor limits

Increased Payload per LoRa Packet: We present the result of the payload increase in Figure 5 as ECDF for a gateway range limit $r = 1.5\,\mathrm{km}$ over all sensor limit scenarios for load increase step S2. The same evaluation is also possible with the baseline without increasing the number of sensors, but the collision probability differences are much smaller. We see, like expected, a collision probability increase with increasing payload size but doubling the payload sizes does not double or more than double the collision probability. For $1\,\mathrm{B}$ payload, we receive an average collision probability of 3.01 %, for $4\,\mathrm{B}$, it is 3.91 %, for $8\,\mathrm{B}$ it is 4.69 %, for $16\,\mathrm{B}$, it is 6.96 %, and for $32\,\mathrm{B}$, it is 10.12 %. This is a result of the aggregation benefit of several small payloads into one. As a last step, we use all possible 28,000 sensor positions in the increase step S3 to determine the network limits. The range limit for this study is again $r = 1.5\,\mathrm{km}$. There, we receive an average collision probability between 24 % and 28 % per placement when using $1\,\mathrm{B}$ payload per LoRa packet. With payloads of $16\,\mathrm{B}$ and $32\,\mathrm{B}$, in some situations more than 50 % of all packets are colliding. In the next step, we try to solve this overload situation by placing additional gateways with our algorithm.

C. Additional Placement Study

Here, we use the initial gateway placement positions of one placement scenario with a gateway range of $r = 1.5\,\mathrm{km}$ and a sensor limit of 500 sensors. We increase the sensor limit per gateway to 2,000 and place additional gateways with the goal of removing load from the gateways in the network center. We use 10 different runs and the initial placement has 18 gateways. After we extend the placement, we receive an average of 28.5 gateways, with a minimum of 27 and a maximum of 30. The average collision probability after the extension is 2.64 % for a payload of $1\,\mathrm{B}$. Thus, we see a massive decrease in the collision probability due to the additional placement. When we compare the result with a complete new placement without re-using the gateway position of the initial placement, we receive a number of 24 to 29 and an average of 25.8 gateways to cover the network. With the new placement, we receive a collision probability of 2.59 %. But note that the extension is not possible without any limits. If all sensors are transmitting with SF 7, additional mechanisms like transmission power control are required. Furthermore, a small impairment for each network extension step compared to a completely new placement is received. Thus, we prefer a robust initial placement compared to many additional placement steps.

D. Discussion

In related works, we see that maximizing the distances between sensors and gateway can minimize the number of required gateways for full network coverage. But usually no robustness towards future load, sensor numbers, and traffic increase is studied. Since minimizing the number of gateways is usually the goal in state of the art literature, we used our ILP to obtain a placement with the minimum number of gateways. There, we set the sensor limit $\ell = \infty$ and use $r = 2.468\,\mathrm{km}$ as the maximal gateway range with the Hata model according to Table III. For this scenario, again, we use a subset of 2,800 randomly selected sensors. We receive an optimal placement of 7 gateways. In contrast, with a gateway range of $r = 1.5\,\mathrm{km}$, we received 17.22 gateways on average. But with the optimal placement, many sensors transmit with large SFs and long ToAs. There, the packet collision probability is $2.62\,\%$ with $1\,\mathrm{B}$ payload, that is far from optimal. By increasing the number of sensors, we also study the future robustness of the optimal placement. For load increase step S1, we receive $5{,}3\,\%$ collision probability and $23.80\,\%$ for S2. Thus, an optimal placement in terms of minimizing the number of gateways is not robust against future load increase. Although it is possible to add gateways on demand, our evaluation shows that the overall number of required gateways is then higher compared to a near-optimal initial placement. For that reason, for a real deployment, we suggest a robust initial gateway placement capable of dealing with future load increase. This requires only few larger network extensions in the more distant future. To study the quality of the presented placement, we compare the result with an optimal placement computed by our ILP. There, we study the collision probability for $1\,\mathrm{B}$ payload for a placement with a subset of 2800 randomly selected sensors, $r = 1.5\,\mathrm{km}$, and 500 sensors as gateway limit. We receive an optimal placement of 14 or 15 gateways compared to 17 or 18 with our VoronoiLocalSearch, based on the random sensor input. The average collision probability in the network is $0.41\,\%$ for the initial placement scenario without load increase, $0.75\,\%$ for load increase step S1, and $3.54\,\%$ for S2. This is slightly higher than our approximation result. Furthermore, the runtime of the ILP lies between $129.52\,\mathrm{s}$ and almost $5\,\mathrm{h}$. The maximal runtime of our VoronoiLocalSearch is $4.77\,\mathrm{s}$. This gap increases for larger instances making the ILP, though comparable in terms of collision probabilities for the computed placement, only applicable for small networks. Our algorithm is very fast, is close to optimal with regard to number of placed gateways, and the increase in placed gateways additionally increases the robustness towards future load increase.

VI. Conclusion

It is essential to consider potential load increase when planning and deploying IoT networks. Our results show that merely minimizing the number of gateways does not yield robust gateway placements for LoRaWANs. In our case study we showed that increasing the number of gateways by a factor of 2.5 allowed us to increase the number of sensors by a factor of 5 until the packet collision probability surpassed that of the gateway-minimal placement. Our evaluation shows that for a LoRaWAN deployment in a small city such as Würzburg, a gateway range of $1.5\,\mathrm{km}$ is preferable for a future-robust placement. Certainly, this value depends on, among others, the sensor density, the geography, and the used path loss model. The overall goal is to minimize the required SFs when transmitting LoRa packages for all sensors without overloading the network with gateways. Additionally, we see that limiting the number of sensors per gateway in the placement has less influence. In future works, a more detailed study of gateway ranges and capacities is essential to further reduce the collision probability in various parts of the networks. We are considering to use a hybrid approach between the denser, inner network and the sparser periphery.

References

[1] R. O. Andrade and S. G. Yoo, "A comprehensive study of the use of LoRa in the development of smart cities," *Appl. Sci.*, 2019.

[2] P. J. Basford, F. M. Bulot, M. Apetroaie-Cristea, S. J. Cox, and S. J. Ossont, "LoRaWAN for smart city IoT deployments: A long term evaluation," *Sensors*, 2020.

[3] N. H. Mustafa and S. Ray, "Improved results on geometric hitting set problems," *Discrete Comput. Geom.*, 2010.

[4] V. Chvátal, "A greedy heuristic for the set-covering problem," *Math. Oper. Res.*, 1979.

[5] H. Brönnimann and M. T. Goodrich, "Almost optimal set covers in finite VC-dimension," *Discret. Comput. Geom.*, 1995.

[6] P. Agarwal and J. Pan, "Near-linear algorithms for geometric hitting sets and set covers," *Discrete Comput. Geom.*, 2020.

[7] N. Bus, N. H. Mustafa, and S. Ray, "Practical and efficient algorithms for the geometric hitting set problem," *Discrete Appl. Math.*, 2018.

[8] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Comm. Magazine*, 2017.

[9] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of LoRaWAN channel access," in *Int. Conf. Eng. and Telecomm. (EnT)*. IEEE, 2016.

[10] G. Ferré, "Collision and packet loss analysis in a LoRaWAN network," in *25th Europ. Signal Processing Conf.* IEEE, 2017.

[11] Q. Liu, Y. Mu, J. Zhao, J. Feng, and B. Wang, "Characterizing packet loss in city-scale LoRaWAN deployment: Analysis and implications," in *IFIP Networking Conf.* IEEE, 2020.

[12] N. Vatcharatiansakul, P. Tuwanut, and C. Pornavalai, "Experimental performance evaluation of LoRaWAN: A case study in Bangkok," in *14th Int. Joint Conf. Comput. Sci. and Softw. Eng.* IEEE, 2017.

[13] N. Matni, J. Moraes, H. Oliveira, D. Rosário, and E. Cerqueira, "LoRaWAN gateway placement model for dynamic Internet of Things scenarios," *Sensors*, 2020.

[14] Y. Jin, L. Ding, F. Yang, L. Qian, and C. Zhi, "LoRa network planning based on improved isodata algorithm," in *Int. Conf. Wireless Comm. Sign. Process. (WCSP)*. IEEE, 2020.

[15] S. Mnguni, A. M. Abu-Mahfouz, P. Mudali, and M. O. Adigun, "A review of gateway placement algorithms on Internet of Things," in *Int. Conf. Adv. Big Data, Computing and Data Comm. Syst.* IEEE, 2019.

[16] B. Ousat and M. Ghaderi, "LoRa network planning: Gateway placement and device configuration," in *Int. Congr. IoT*. IEEE, 2019.

[17] P. K. Mishra, S. K. Jena, G. K. Das, and S. V. Rao, "Capacitated discrete unit disk cover," *Discret. Appl. Math.*, 2020.

[18] S. Bandyapadhyay, S. Bhowmick, T. Inamdar, and K. R. Varadarajan, "Capacitated covering problems in geometric spaces," *Discret. Comput. Geom.*, 2020.

[19] M. Paredes, S. Bertoldo, L. Carosso, C. Lucianaz, E. Marchetta, M. Allegretti, and P. Savi, "Propagation measurements for a LoRa network in an urban environment," *J. Electromag. Waves Appl.*, 2019.

[20] S. V. Akram, R. Singh, M. A. AlZain, A. Gehlot, M. Rashid, O. S. Faragallah, W. El-Shafai, and D. Prashar, "Performance analysis of IoT and long-range radio-based sensor node and gateway architecture for solid waste management," *Sensors*, 2021.

[21] Semtech. (2021) Sx1276/77/78/79 datasheet. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rbr/6EfVZUorrpoKFfvaF_Fkpgp5kzjiNyiAbqcpqh9qSjE

[22] F. Metzger, T. Hoßfeld, A. Bauer, S. Kounev, and P. E. Heegaard, "Modeling of aggregated IoT traffic and its application to an IoT cloud," *Proc. IEEE*, 2019.