Network Size Estimation for LoRa-Based Direct-to-Satellite IoT

Diego Maldonado, Juan A Fraire, Pablo Ilabaca, Hervé Rivano, Sandra Céspedes

Network Size Estimation for LoRa-Based Direct-to-Satellite IoT

Diego Maldonado*, Juan A. Fraire*†‡, Pablo Ilabaca§, Hervé Rivano*, Sandra Céspedes§
*Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France
†CONICET - Universidad Nacional de Córdoba, Argentina
‡Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
§CSSE Department, Concordia University, Montreal, QC, Canada.

***Abstract*—The emerging paradigm of Direct-to-Satellite Internet of Things (DtS-IoT) involves Earth surface nodes communicating directly with Low Earth Orbit (LEO) satellites, utilizing standard Low-Power Wide Area Networks (LPWAN) protocols. One of the core challenges faced in this paradigm is scaling the Medium Access Control (MAC) from a limited number of nodes to potentially thousands within the satellite's coverage area. To address this issue, medium access control schemes can utilize a priori information on the number of nodes the satellite will cover along its orbit. However, developing technically viable solutions for network size estimation that are both precise and accurate remains an open research challenge. This work presents the implementation, parameter selection, and evaluation of the first LoRa/LoRaWAN-compatible network size estimation protocol that leverages the onboard Optimistic Collision Information (OCI) estimator. Our solution, LoRa-OCI (L-OCI), was integrated into FLoRaSat, a C++ discrete-event DtS-IoT simulator that integrates realistic orbital and LoRa/LoRaWAN communication models. Through an extensive simulation campaign, we can determine appropriate LoRa configurations to achieve low root mean square error (RMSE) and low power consumption. Additionally, our results indicate that the approach is relatively insensitive to LoRa parameters when assessing the aggregated throughput of a Slotted ALOHA Game (SAG) protocol throttled by L-OCI.**

***Index Terms*—Direct-to-Satellite, Internet of Things, LoRa/LoRaWAN, Network Size Estimation**

I. Introduction

Direct-to-Satellite Internet of Things (DtS-IoT) is a novel approach to integrated terrestrial and spatial wireless communications where small low-energy nodes on Earth directly communicate with a Low Earth Orbit (LEO) satellite [1]. Small satellites, including affordable CubeSats, enable the establishment of dependable and cost-effective networks by relaying packets to ground stations or other satellites in cases where a constellation is deployed [2]. This can lead to an expansion of the network's coverage and support. Conventional Low-Power Wide Area (LPWAN) technologies [3], such as LoRa, which are typically employed in urban and rural applications, can be modified to support DtS-IoT networks [4]. Furthermore, adopting open standards like LoRaWAN facilitates the unrestricted deployment of LoRa networks, as they comply with regional regulations. When coupled with CubeSats, this approach offers an economical alternative for building satellite networks that is competitive with available low-power, low-data rate satellite solutions (e.g., Argos [5], APRS [6], S-AIS [7], and ADS-B [8]).

One of the primary obstacles encountered in DtS-IoT networks involves the scalability of the network, particularly in situations where thousands of nodes (e.g., asset tracking, environmental monitoring, smart metering, agriculture, utilities, etc.) must be served by a single satellite. Using Intelligent Medium Access Protocols (MAC) is critical in addressing the unavoidable collisions between packets and enhancing the network's overall performance. Recent research has shown that incorporating information on the network size, i.e., the number of nodes under satellite coverage, into a purpose-built MAC protocol [9] can significantly enhance the network's performance in terms of throughput. However, obtaining this information is often challenging, as the nodes are frequently deployed in harsh, remote environments and may be isolated from one another [10].

This study aims to implement and evaluate a network size estimation mechanism for DtS-IoT using the LoRa/LoRaWAN protocol stack. We employ the Optimistic Collision Information (OCI) estimator [10] to achieve this goal. The selection of OCI is based on its superior performance in terms of low Root Mean Square Error (RMSE) estimation and power efficiency compared to other modern network size estimators. Nevertheless, the simulations conducted in [10] relied on simplistic assumptions without considering any underlying DtS-IoT protocol or communication model. Our proposed mechanism, named LoRa/LoRaWAN-based OCI (L-OCI), is designed to operate on a Frame Slotted ALOHA approach, employing LoRa/LoRaWAN framing and Chirp Spread Spectrum modulation with the corresponding Spreading Factors (SFs) specified in the LoRa/LoRaWAN specification [11]. Consequently, our approach can be implemented on standard commercial IoT nodes. To assess the feasibility of OCI in a practical scenario, we offer a multi-objective parameter selection algorithm for L-OCI. Also, we implemented L-

This work has received support from Project STARS STICAMSUD 21-STIC-12 Code STIC2020003, Project ANID FONDECYT Regular 1201893, ANID Basal Project FB0008, the French National Research Agency (ANR) ANR-22-CE25-0014-01 STEREO project, and the MISSION project from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 101008233.

979-8-3503-3567-5/23/$31.00

OCI and conducted a comprehensive simulation using the FLoRaSat simulator [12]. FLoRaSat was specifically developed to simulate DtS-IoT networks, incorporating LoRa-based networks and realistic orbital propagation and channel models.

The paper is structured as follows. Section II provides background information on LoRa, LoRaWAN, and the current state of network size estimation. Section III provides the details of L-OCI. Section IV presents the simulation scenario and discusses the results. Finally, conclusions are presented in Section VI.

II. Background

A. LoRa and LoRaWAN Communication

LoRa is an IoT radio communication technique developed by Semtech and is widely used in various scenarios, making it a suitable option for the DtS-IoT paradigm. The LoRa radio has five configuration parameters that include Transmission Power (TP), Carrier Frequency (CF), Spreading Factor (SF), Bandwidth (BW), and Coding Rate (CR). The selection of these parameters can significantly impact the network performance. LoRa operates on license-free Industrial, Scientific, and Medical (ISM) frequency bands regulated by telecommunication organizations for different world regions [13].

LoRaWAN is a data link layer built on top of the physical layer provided by LoRa [11]. The LoRaWAN specification defines three operation modes, Class A, Class B, and Class C, to enhance the adaptability of the network to different use scenarios. LoRaWAN Class A nodes are typically low-power end nodes that can send data to a gateway but only receive data in response after sending data. Class B nodes, on the other hand, can receive data at predetermined time slots to enable time-critical applications. Lastly, Class C nodes continuously listen for incoming data after sending it, making them suitable for applications requiring immediate responsiveness.

LoRa/LoRaWAN has been demonstrated to be an effective communication technology for DtS-IoT in several recent studies. In-orbit deployments, such as ThingSat, FossaSat, and LacunaSat satellite series have successfully demonstrated the feasibility of using LoRa/LoRaWAN in ISM bands for direct node-to-satellite communications. Additionally, numerous research papers, including [1], [4], [14], have supported the approach. The present study centers on utilizing LoRa's regional parameter set for the European region in the context of DtS-IoT. In addition, a variant of the LoRaWAN Class B mode is assumed due to its beaconing-based approach, which is deemed appropriate for detecting the presence of satellites in the DtS-IoT framework.

B. OCI Network Size Estimator

OCI is based on the observation that in a network with a large number of nodes, collisions frequently occur, while in a network with a small number of nodes, collisions occur less frequently. OCI uses the probability of successful transmission to calculate the number of nodes in the network. To this end, nodes transmit (small) dummy frames, and OCI estimates the number of participating nodes on the receiver side [10]. The OCI estimator has been proven to be highly accurate with a low Root Mean Square Error (RMSE) compared to other state-of-the-art estimators. Moreover, the OCI mechanism is computationally efficient, making it feasible to implement onboard resource-constrained nano-satellites. The OCI estimator operates as follows.

a) Data Collection Phase: Participating nodes uplink dummy frames following a Frame Slotted ALOHA (FSA) delimited by a beaconing system. After the transmission of a beacon, the frame starts. During the frame, the satellite determines the status of each uplink slot, which can be categorized as *idle*, *successful*, or *collided* depending on whether there is no received signal (below power sensitivity), a successful signal decode occurs, or two or more nodes attempt to transmit on the same slot, respectively. The resulting tuple i, c, s represents the number of *idle*, *collided*, and *successful* slots, respectively. It should be noted, however, that a slot may be perceived as *idle* by the satellite, even when multiple nodes attempt to transmit on the same slot. Still, low reception power makes successful decoding infeasible.

b) Naive Estimation Phase: Using the collected data, a simple estimate of the number of nodes n_k within the satellite's footprint (or cluster size) during frame k is obtained, assuming that no more than two nodes are involved in every collision. This estimate ϕ of n_k is computed as $\phi = s + 2c$. However, it should be noted that the naive estimation may not be accurate in realistic scenarios where high-order collisions are not neglected, and it may underestimate the number of nodes when $n_k > w$. Fig. 1 illustrates the naive estimation as a function of the cluster size n_k for various frame lengths in terms of slots w, demonstrating that it always converges to $2w$.

c) Polynomial Fitting Phase: To address the underestimation issue without knowledge of the order of collisions, OCI creates a function that maps the naive estimations curve to the real number of nodes curve. For this method to work, the naive estimations and the real number of nodes must be bijective, and the frame length w must be long enough so that the naive estimation has no repeated values. To create the function, two polynomial fits are utilized. The first fit is applied to smooth the naive estimation curve, while the second

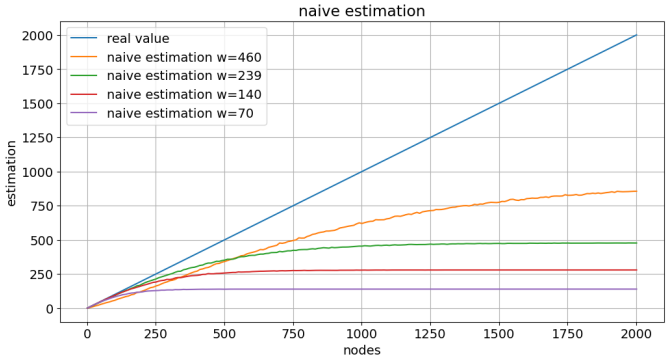

Fig. 1. Naive estimations for frames with different number of slots w

fit maps the resulting curve of the first fit to the real number of nodes curve.

d) Operation Phase: After obtaining the polynomial coefficients of the second fit, the OCI mechanism enters the operational phase, where the function is applied to the naive estimation to obtain the OCI estimation of the real value. The value obtained from OCI can be utilized to optimize the performance of MAC schemes by regulating their operation. This value can be stored onboard or delivered to ground operations for future use.

The efficiency of OCI is remarkable and discussed in [10]. However, the OCI estimator was conceived based on a series of assumptions that limit its applicability. The assumption that the beacon is always decoded and all nodes remain synchronized with the satellite neglects the potential for errors in realistic DtS-IoT networks. The model assumes that each frame contains a separate stable cluster of nodes that remain under satellite coverage for the entire time frame. However, in a realistic scenario, the nodes within the satellite's coverage area can experience rapid changes. Some nodes may receive the beacon signal but subsequently move out of coverage, while others may come into coverage but miss the beacon. Moreover, the model only accounts for collisions as the cause of data loss, neglecting the possibility of the capture effect signal strength below the receiver sensitivity level.

In the subsequent section, we present a modification of the OCI estimator tailored to the practical LoRa/LoRaWAN protocols implemented in DtS-IoT, thereby alleviating the aforementioned idealized premises.

III. L-OCI

One cannot directly use the OCI estimator in LoRa-based networks without extending the MAC layer specified by LoRaWAN. Specifically, a protocol based on a TDMA technique with time-slotted frames and a time synchronization system must be added to the MAC layer.

For L-OCI, we leverage LoRaSync [15], which we find is an appropriate Class B variant technique that satisfies the synchronization requirements of OCI and DtS-IoT. LoRaSync extends LoRaWAN Class B by adopting a beaconing mechanism that enables time synchronization and scheduling uplink slots during the beacon window. Fig. 2 depicts the structure of the beacon period, which is divided into three phases.

a) Phase 1: Beacon Reserved: During this phase, the gateway broadcasts the beacon, and ground nodes decode it. To optimize energy usage, avoiding persistent reception in resource-constrained nodes is important. Therefore, it is necessary to use appropriate techniques such as [16] to estimate the satellite visibility, which can determine the optimal time to open the reception window. Once the beacon is received, the node will be synchronized on a frame level.

b) Phase 2: Beacon Window: Then, uplink slots (UL) are slotted in time during this phase. A UL includes a maximum clock offset threshold time δ_{max} at the beginning and end of the slot and a maximum time on air ToA_{max} in between, which depends on the LoRa parameters and the payload size. The parameter δ_{max} provides the required time synchronization contention guards for the beacon and clock drifts in DtS-IoT. Note that the downlink mechanism defined by LoRaWAN Class B still exists but is not depicted here since user data downlinks are unnecessary for L-OCI estimation.

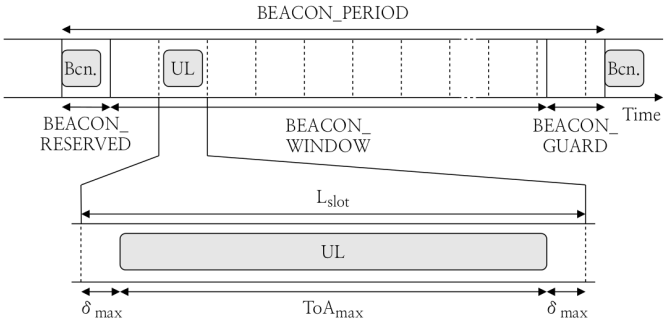

Fig. 2. LoRaSync beacon period and slot structure (adapted from [15])

c) Phase 3: Beacon Guard: Finally, the beacon period ends with the guard phase, allowing the gateway to decode the last transmissions.

The number of slots w in the frame is a crucial parameter for L-OCI, as illustrated in Fig. 1. It is calculated as $w = \lfloor T_{win}/T_{slot} \rfloor$, where T_{win} is the beacon window time, and T_{slot} is slot duration. Therefore, w depends on the frame length, the maximum time-on-air (ToA_{max}), and the underlying LoRa parameters. The choice of SF and BW parameters of LoRa, and the frame length influence this non-trivial trade-off that determines the optimal operation of L-OCI. A higher SF (e.g., SF12) increases communication range and ToA_{max} but reduces the number of slots in the frame, which can limit the number of transmission opportunities. Conversely, a lower SF like (e.g., SF9) reduces both communication range and ToA_{max}, allowing more slots in the frame but reducing communication range. An optimal L-OCI parameter selection mechanism is provided below.

A. Parameter Selection Algorithm

The parameters involved in LoRa and LoRaSync are numerous. This study focuses on three specific parameters: Spreading Factor (SF), Bandwidth (BW), and frame length (BCN). A heuristic approach is proposed to determine the optimal combination of parameters $x = (SF, BW, BCN)$ for a given cluster size n_k. To establish an optimal solution, a Multi-objective Optimization Problem (MOP) is defined in this study based on two objectives related to the OCI estimator. The first objective is minimizing the energy spent to obtain the estimator's polynomial coefficients, while the second is minimizing the estimation error. The problem is formalized as

$$\text{minimize} \;\; F(x) = (f_1(x), f_2(x))^T, \;\; \text{s.t.} \;\; x \in \Omega, \qquad (1)$$

where Ω is the decision space. The first objective function, $f_1(x)$, is the amount of energy consumed in transmissions that failed to be decoded due to collisions or insufficient received signal power during the OCI estimation phase. The second

objective, $f_2(x)$, is the estimation error attained during the OCI test phase. Here, x is only constrained to live in Ω.

Various methods address the MOP presented in equation (1). The approach utilized in this study is MOEA/D [17], a genetic programming-based algorithm that facilitates obtaining the Pareto curve of optimal solutions for a given multi-objective optimization problem. It is chosen due to its excellent search performance and high computation efficiency. To adapt the problem to MOEA/D, the parameter vector x was converted into the decision vector $\bar{x} = \{w(SF, BW, BCN), ToA_{max}(SF, BW)\}$. The remaining parameters of LoRa and LoRaSync are treated as fixed values. The Pareto curve of the optimal solution to the MOP is obtained by the MOEA/D algorithm, which decomposes the objectives into two scalar optimizations. During each generation, the population of candidates' objective function $F(x)$ is calculated by simulating a smaller version of L-OCI. The model considers the spatial distribution of the network using LoRaSync, making it more accurate than OCI but less than L-OCI as implemented in the FLoRaSat simulator.

B. Performance Metrics

a) RMSE: To assess the efficacy of the OCI estimations, the evaluation criterion of choice is the Root-Mean-Square Error (RMSE), also used in [10]. The RMSE is derived through the calculation of the formula

$$\epsilon_{rsme} = \sqrt{\sum_{k=1}^{K} (\phi_k - n_k)^2 / K}, \tag{2}$$

where K corresponds to the number of clusters being examined, n_k signifies the actual number of nodes in the kth cluster, and ϕ_k denotes the estimated value of n_k. This approach provides an intuitive measure of the accuracy of the estimator's predictions by quantifying the differences between the predicted and actual values of network size, thereby allowing for comparing the estimator's performance under different network configurations.

b) Throughput: The Slotted Aloha Game (SAG) [9] is a MAC protocol proposed for satellite networks using FSA communication, thus suitable for OCI and L-OCI. SAG gives a transmission probability to each node based on the number of slots and the currently active nodes in the network (estimated with OCI or L-OCI). If there are fewer nodes than slots, the probability is set to 1, meaning all nodes will attempt to transmit during the frame. If there are more nodes than slots, the probability is adjusted so that the number of nodes attempting to transmit is close to the number of slots. SAG aims to achieve the maximum theoretical throughput with arbitrarily large node populations. It is calculated as the number of successful transmissions in a frame divided by the time length of the frame in slots.

C. Implementation in FLoRaSat

FLoRaSat (Framework for LoRa-based Satellite Networks) is a simulation tool built on top of OMNeT++ discrete-event simulator to model end-to-end satellite IoT scenarios using LoRa/LoRaWAN adaptations for space applications [12].

The simulator is a combination of several existing frameworks. The LoRa portion of the simulation software is founded on the FLoRa (Framework for LoRa), initially designed to investigate IoT applications in urban and suburban areas. FLoRaSat is designed to replicate LoRa's capture effect by decoding the initial transmission, with any subsequent overlapping signals considered collisions. As all nodes are assumed to have the same configuration settings, it can be expected that the nearest node to the satellite will transmit the first and strongest signal during a given period. In addition, the LoRaWAN layer was modified to include the LoRaSync and L-OCI mechanisms. To incorporate satellite mobility, the *OS3* [18] and *leosatellites* [19] frameworks were updated and adapted, resulting in a suitable implementation of the SGP4 orbital propagator model.

FLoRaSat generates the dataset for naive estimations during the L-OCI training phase. The L-OCI is then trained offline with this dataset using our Python implementation. The resulting estimators are then provided to the nodes' MAC layer in the extended FLoRaSat to simulate the SAG traffic simulations[1].

IV. Results

A. Scenario and Parameters

This study considers a specific scenario where a Low Earth Orbit (LEO) satellite serves as the DtS-IoT gateway for relaying packets from ground nodes to the network server via a ground station. The satellite is assumed to be in a circular orbit with an altitude of 600 kilometers and an inclination of 98 degrees. Ground nodes are randomly and uniformly deployed over a circular region. The satellite is positioned directly above the deployment center to ensure all nodes receive L-OCI's beacon signal. Consequently, all nodes attempt to transmit during the frame on a randomly selected slot according to a uniform distribution. Fig. 3 depicts this scenario using the FLoRaSat GUI, where the red area represents the communication range based on the selected parameters.

The control variables of our study are the spreading factor (SF), the bandwidth (BW) of LoRa in kHz, and the beacon period (BCN) in seconds. At the same time, the remaining parameters comply with the LoRa Alliance specification for the European region [13]. The payload size is set to 20 bytes. We determine the time on air (*airtime*) and the maximum range of communication (*range*) for a LoRa transmission based on the values of (SF, BW). The channel model assumed is a free space path loss model, and we consider the received sensitivity values from a typical LoRa transceiver, Semtech's SX1272 [20]. We run 30 simulation repetitions with different seeds for each input set (SF, BW, BCN).

B. Experiments and Results Analysis

1) Sensitivity to LoRa Parameters: The present study examines the influence of several L-OCI estimators trained using

[1]FLoRaSat repository with L-OCI: https://gitlab.inria.fr/jfraire/florasat

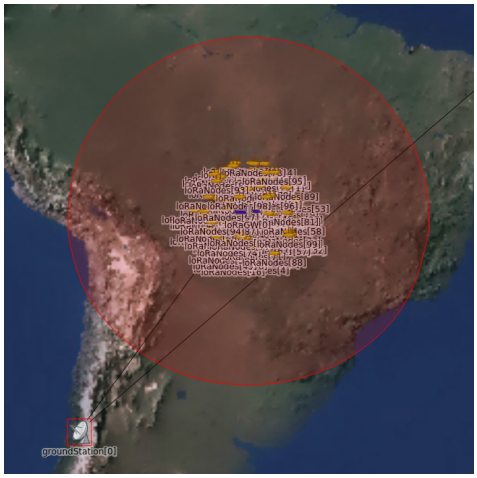

Fig. 3. DtS-IoT simulation scenario (FLoRaSat UI)

TABLE I
STUDY PARAMETERS AND ERROR
(TRAINING PARAMETERS: w=167, $SF = 10, BW = 125, BCN = 91$)

SF	BW	BCN	RSME
9	125	50	29.693
10	125	91	5.556
11	250	80	5.586
11	125	152	5.747
12	125	295	17.909

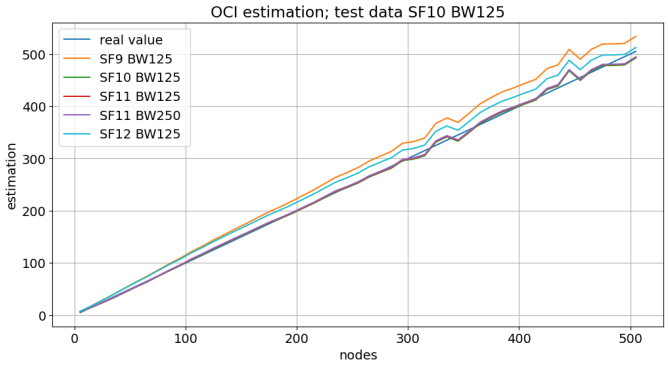

Fig. 4. Comparison of different L-OCI estimators

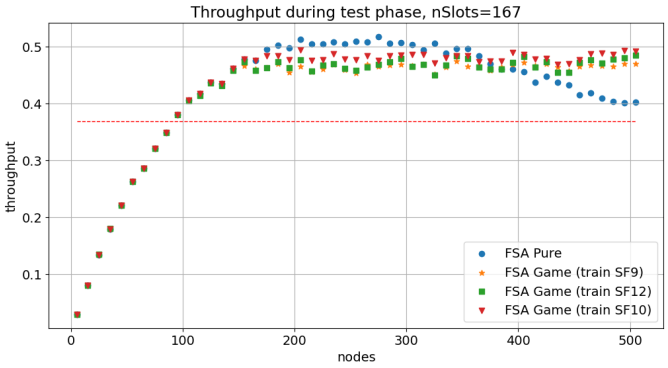

Fig. 5. Throughput comparison using estimations from Fig. 4

different sets of parameters listed in Table I on a simulation with identical settings ($SF = 10, BW = 125, BCN = 91$). The study employs $w = 167$ slots and varies the number of nodes from 1 to 501 in increments of 10 nodes. The RSME of the estimators is presented in the last column of Table I, and their estimations are illustrated in Fig. 4. At the same time, the nodes' positions are adjusted with respect to the training phase. Although the smallest RSME is achieved when the simulation parameters match the trained L-OCI (($SF = 10, BW = 125, BCN = 91$)), $SF11$ yields comparable performance, even when operating in a different BW, unlike $SF9$ and $SF12$. Further analysis revealed a considerable number of lost transmissions due to weak signal power in the reception during the L-OCI estimation phase, resulting in an overestimation of the number of nodes by L-OCI. The findings suggest that L-OCI is insensitive to LoRa's $SF10$ and $SF11$ and that a single L-OCI estimator can be applied to both.

2) Impact on Throughput: In this experiment, we investigate the impact of using L-OCI estimations on the throughput of the SAG protocol, as presented in [9], [10], during the operational phase. We use the OCI estimations with $BW = 125$ obtained from the previous experiment: $SF9$, $SF10$, and $SF12$. The obtained throughput is compared with the FSA pure throughput, where no uplink throttling is applied, serving as a baseline. ALOHA's maximum theoretical throughput is provided in a red horizontal line. The results are presented in Fig. 5. We explain the results above ALOHA's theoretical throughput with the capture effect modeled in FLoRaSat. Indeed, a collision can still result in successful frame decoding. The pure FSA protocol achieves maximum throughput between 200 and 300 nodes, slightly higher than the scenarios employing SAG, but FSA's throughput decreases steadily after that, validating the findings in [10]. Instead, with the SAG access control, the throughput peaks when the number of nodes equals the number of slots, oscillating around these values after that. The highest average throughput is observed when using the OCI estimations obtained with SF10 (48.1%), which also has the lowest RSME, as it was the parameter for which L-OCI was trained. However, the difference in throughput concerning SF9 (46.5%) and SF12 (46.8%) scenarios is minimal. To wrap up, the throughput results suggest using SF10 or SF11 to provide the best performance of the trained L-OCI model.

3) Error and Energy Trade-off: This experiment evaluates the effectiveness of the MOP heuristic proposed in Section III-A for optimizing energy consumption and estimation error in LoRa networks. The wasted power and RMSE are jointly analyzed in Fig. 6, where the former is calculated as the power spent by a node in non-successful transmissions. A parameter selection campaign was conducted for 100, 500, 1000, and 1500-node network sizes. Results show that wasted energy and estimation error increase with the number of nodes in the network. Moreover, the most energy-efficient solutions were obtained using $SF = 9$ and $BW = 125$ for all cases.

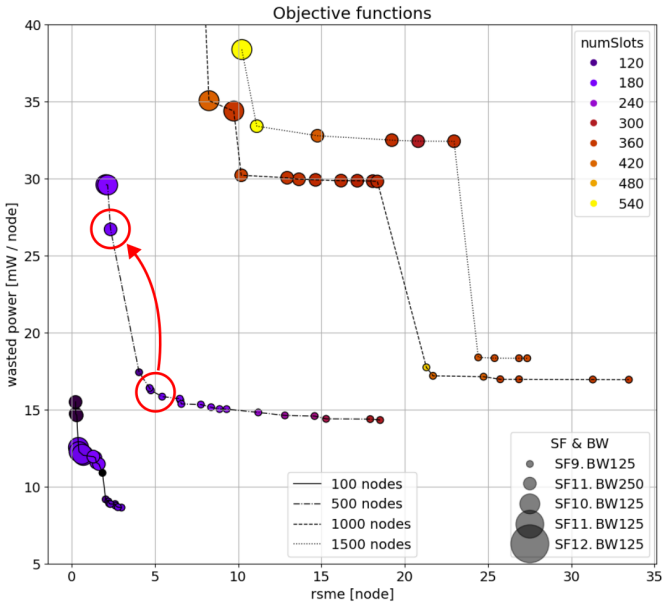

Fig. 6. Pareto curves for different network sizes obtained from MOP

For instance, the Pareto curve for a network of 500 nodes suggests that a parameter set with $SF = 9$ and $BW = 125$ can achieve an estimation error of $RSME = 5$ and a wasted energy per node of $16mW$. To further reduce the error, switching to $SF = 11$ and $BW = 250$ is required, resulting in a reduced error of $RSME = 2.5$ but increased wasted energy to $27mW$ per node (see red arrow in Fig. 6). The results demonstrate the effectiveness of the MOP heuristic for optimizing energy utilization and estimation error in LoRa networks. Furthermore, in light of the observations in the previous experiment, the marginal improvement in error might not be justified in terms of throughput due to the significant increase in energy consumption.

C. Takeaways

The study found that L-OCI is well-suited for LoRa-based DtS-IoT networks, but the modulation parameters and beacon period also impact communication. Multiple OCI estimators may be required for optimal performance, but when resources are limited, a single estimator can support diverse parameters with minimal error and negligible impact on throughput. SF9 provides optimal energy utilization at the expense of reduced error, but higher SF may be necessary for tighter link budgets.

V. Conclusion

In this work, we proposed L-OCI: a LoRa/LoRaWAN realization of the Optimistic Collision Information (OCI) for Direct-to-Satellite IoT (DtS-IoT). We evaluated L-OCI in realistic LoRa-based DtS-IoT networks providing the first evidence of the expected performance of the approach. An extensive simulation campaign showed that L-OCI maintains low error estimations and power efficiency, even in scenarios with a high packet loss rate. Also, L-OCI proved insensitive to $SF = 10$ and $SF = 11$ regarding RSME and throughput, suggesting a single estimator can fit both use cases. However, in resource-full satellites, multiple estimators could co-exist for optimal performance. Furthermore, our analysis showed that $SF = 9$ and $BW = 125kHz$ are appealing candidates for improving energy efficiency if the link budget can be closed under such parameters. Future work includes an extended analysis with different orbital parameters, more training parameters for L-OCI, and L-OCI extensions considering multiple SFs in the same frame.

References

[1] J. A. Fraire, S. Céspedes, and N. Accettura, "Direct-to-satellite IoT-a survey of the state of the art and future research perspectives: Backhauling the IoT through LEO satellites," in *ADHOC-NOW 2019, Proceedings 18*. Springer, 2019, pp. 241–258.

[2] J. A. Fraire, S. Henn, F. Dovis, R. Garello, and G. Taricco, "Sparse satellite constellation design for lora-based direct-to-satellite internet of things," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[3] F. Gu, J. Niu, L. Jiang, X. Liu, and M. Atiquzzaman, "Survey of the low power wide area network technologies," *Journal of Network and Computer Applications*, vol. 149, p. 102459, 2020.

[4] J. A. Fraire, O. Iova, and F. Valois, "Space-terrestrial integrated internet of things: Challenges and opportunities," *IEEE Comms. Magazine*, 2022.

[5] C. Vincent, B. J. Mcconnell, V. Ridoux, and M. A. Fedak, "Assessment of argos location accuracy from satellite tags deployed on captive gray seals," *Marine Mammal Science*, vol. 18, no. 1, pp. 156–166, 2002.

[6] R. Patmasari, I. Wijayanto, R. Deanto, Y. Gautama, and H. Vidyaningtyas, "Design and realization of automatic packet reporting system (APRS) for sending telemetry data in nano satellite communication system," *Journal of Measurements, Electronics, Communications, and Systems*, vol. 4, no. 1, pp. 1–7, 2018.

[7] J. Carson-Jackson, "Satellite AIS – developing technology or existing capability?" *Journal of Navigation*, vol. 65, no. 2, p. 303–321, 2012.

[8] K. Werner, J. Bredemeyer, and T. Delovski, "ADS-B over satellite: Global air traffic surveillance from space," in *2014 Tyrrhenian International Workshop on Digital Communications-Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV)*. IEEE, 2014, pp. 47–52.

[9] B. Zhao, G. Ren, and H. Zhang, "Slotted aloha game for medium access control in satellite networks," in *2019 IEEE/CIC Int. Conference on Communications in China (ICCC)*. IEEE, 2019, pp. 518–522.

[10] P. Ilabaca Parra, S. Montejo-Sánchez, J. A. Fraire, R. D. Souza, and S. Céspedes, "Network size estimation for direct-to-satellite iot," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6111–6125, 2022.

[11] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "Lorawan specification," *LoRa alliance*, vol. 1, 2015.

[12] J. A. Fraire, P. Madoery, M. A. Mesbah, O. Iova, and F. Valois, "Simulating lora-based direct-to-satellite iot networks with florasat," in *2022 IEEE 23rd WoWMoM*. IEEE, 2022, pp. 464–470.

[13] LoRaAlliance, "Rp002-1.0.3 lorawan® regional40 parameters," Available at https://hz137b.p3cdn1.secureserver.net/wp-content/uploads/2021/05/RP002-1.0.3-FINAL-1.pdf?time=1681357388 (2023/04/17).

[14] G. Colavolpe, T. Foggi, M. Ricciulli, Y. Zanettini, and J.-P. Mediano-Alameda, "Reception of lora signals from leo satellites," *IEEE TAES*, vol. 55, no. 6, pp. 3587–3602, 2019.

[15] L. Chasserat, N. Accettura, and P. Berthou, "Lorasync: energy efficient synchronization for scalable lorawan," 2022.

[16] C. Han, Y. Zhang, S. Bai, X. Sun, and X. Wang, "Novel method to calculate satellite visibility for an arbitrary sensor field," *Aerospace Science and Technology*, vol. 112, p. 106668, 2021.

[17] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[18] A. Valentine and G. Parisis, "Developing and experimenting with leo satellite constellations in omnet++," *arXiv:2109.12046*, 2021.

[19] B. Niehoefer, S. Šubik, and C. Wietfeld, "The cni open source satellite simulator based on omnet++," in *Proc. of the 6th International ICST Conference on Simulation Tools and Techniques*, 2013, pp. 314–321.

[20] SEMTECH, "Sx1272/73 datasheet," Available at https://www.semtech.com/products/wireless-rf/lora-connect/sx1272 (2023/04/17).