A Novel Autonomous Adaptive Frame Size for Time-Slotted LoRa MAC Protocol

Hanan Alahmadi, Fatma Bouabdallah, Ahmed Al-Dubai and Baraq Ghaleb

***Abstract*—LoRa networks represent a promising technology for IoT applications due to their long range, low cost, and energy efficiency. However, their ALOHA-based access method and duty cycle restrictions can limit their scalability and reliability in high-density networks. To address such a challenge, this paper proposes the Autonomous Time-Slotted LoRa (ATS-LoRa), protocol that allows LoRaWAN nodes to autonomously determine their optimal transmission parameters without extensive downlink transmissions from the gateway. ATS-LoRa utilizes the geographical coordinates of the nodes and their gateway in a novel way to allow them to determine their appropriate transmission parameters such as the spreading factor (SF), the channel frequency (CF), and the slot ID (SID). ATS-LoRa performance is evaluated through extensive simulations under different operating conditions showing an average throughput of around 47 times better than the Adaptive Data Rate (ADR) algorithm of LoRaWAN protocol.**

***Index Terms*—LPWAN, LoRaWAN, IoT, TDMA, MAC protocols**

I. Introduction

ACHIEVING a long transmission range while maintaining a low energy consumption profile is crucial for a wide range of Internet of Things (IoT) applications such as smart cities[1], smart monitoring [2], and smart agriculture [3]. Short-range wireless technologies like Bluetooth and Zigbee are limited in their range, while cellular technologies like 4G and 5G are power-hungry. In response, Low Power Wide Area Networks (LPWAN) [4] have emerged as a promising alternative, albeit with the tradeoff of reduced transmission rates and increased latency. Among LPWAN technologies, LoRa[5], short for Long Range, stands out due to its cost-effective deployment and, thus, has recently garnered significant attention from the research community. Additionally, LoRa's proprietary physical layer provides high resistance to interference and enables massive connectivity. It offers an estimated coverage range of up to 5 km in urban areas and up to 15 km in rural areas, with data transmission rates ranging from 0.3 to 37.5 Kbps and an estimated battery lifetime of up to 10 years [6].

LoRa is a unique radio communication technology underpinned by a proprietary modulation technique called Chirp Spread Spectrum (CSS). LoRaWAN, built upon LoRa modulation, serves as its Medium Access Control (MAC) protocol and provides networking capabilities. Leveraging a star topology analogous to cellular networks, LoRaWAN enables end nodes to transmit packets to gateways with a maximum payload of 250 bytes [7]. Gateways then relay these packets to the LoRaWAN network server, which assumes responsibility for the overall network management. LoRa's physical layer offers various transmission parameters that significantly impact the overall network's performance. These parameters include different supported bandwidths (BW), carrier frequencies (CF), spreading factors (SF), transmission powers (TX), and code rates (CR). The typical BW values are 125 kHz, 250 kHz, and 500 kHz, with 125 kHz being the most commonly used. The number of available channels depends primarily on the node deployment region. For instance, the European band, assumed in this study, has eight uplink channels [7]. Spreading factors (SF) are configurable radio parameters ranging from 7 to 12, determining the number of payload bits encoded in a given chirp signal. Lower SF values result in higher data rates, leading to lower energy consumption and transmission time, however, over short distances and vice versa. Additionally, SFs are orthogonal [5], meaning simultaneous transmissions on the same channel frequency (CF) but encoded with different SFs can be successfully received at the gateway, provided the difference in their received power does not exceed a specific threshold. The physical layer employs coding rates (CR) of the forward error correction (FEC) mechanism, supporting values of $4/5$, $4/6$, $4/7$, and $4/8$.

In order to regulate the access to the unlicensed shared medium, LoRa is imposed to respect harsh duty cycles. The duty cycle can be defined as the percentage of time a node or a gateway can transmit during a given time period. For example, in the Europe band, EU863-870 ISM Band [7], the duty cycle for uplink and downlink transmissions from nodes to the gateway and vice versa should not exceed 1% on all the eight uplink/downlink channels. In other words, to maintain the duty cycle, a node or the gateway should wait for at least $ToA_{SF} \times 99$ before transmitting again after transmitting a packet with a duration of ToA_{SF} on these channels. ToA_{SF} is the packet's Time on Air using spreading factor SF [8]. For downlink transmission. there is one extra dedicated downlink channel with 10% duty cycle. However, even with this channel the problem of constrained downlink traffic is not resolved. As a result, the gateway, with its limited duty cycle, cannot easily regulate the node's uplink transmissions and their parameters through downlink packets. In other words, adopting a centralized approach where gateways control node's transmissions through downlink packets that are transmitted to individual nodes, which is the case of the

H. Alahmadi and F. Bouabdallah are with the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia (email: hjalahmadi1@kau.edu.sa).

A. Al-Dubai and B. Ghaleb are with the School of Computing, Edinburgh Napier University, Edinburgh, UK.

Adaptive Data Rate (ADR) algorithm in LoRaWAN, might not be optimal. Thus, duty cycle regulation is considered as one of the main challenges in LoRa networks [9].

LoRaWAN's MAC layer utilizes the ALOHA access approach, where nodes transmit packets immediately as long as their duty cycle allows. However, this approach is prone to collisions with a large number of connected nodes, hindering network scalability. TDMA, an alternative MAC protocol, offers improved performance where nodes access the channel in their predetermined time slots that are distributed centrally by the gateway node/server [10]. However, the centralized scheduling mechanism can overload LoRa networks as it requires the server to synchronize nodes with the gateway. Hence, it is evident that decentralized TDMA approaches are more suitable for LoRa networks due to their ability to rapidly converge to the optimal network configuration without being constrained by duty cycle limitations.

Hence, this paper introduces the Autonomous Time-Slotted MAC protocol (ATS-LoRa) that allows LoRa nodes to self-determine their transmission parameters without relying on extensive downlink transmissions from the gateway. Our innovative approach empowers nodes to autonomously calculate their optimal transmission parameters based on the location information of nodes and gateways which significantly enhances network performance in terms of collision rate, throughput, end-to-end delay, and energy efficiency, as detailed in Section IV. **Some localization techniques can be used to localize nodes as well as their associated gateway [11] [12]. In fact, it is undeniable that using localization techniques may not give very accurate results but, in our study, very precise locations are not required for the well-functioning of the algorithms. Indeed, according to our algorithm, determining the best transmission parameters for a node needs a range of distances from the gateway rather than the precise locations. The main novelty of ATS-LoRa is the dynamic, relatively short frame sizes, and slot durations. Simulation results show that ATS-LoRa protocol achieves a frame size of only 150 slots with a very large number of connected nodes (4000 nodes). Indeed, in the previous proposed protocol, named Sector-Based Time-Slotted (SBTS-LoRa) MAC protocol [13], although it considerably reduces the collision rate compared to LoRaWAN ADR algorithm, but at the expense of large frame size especially in dense networks. In other words, with a larger number of connected nodes (5000 nodes) the frame size in SBTS-LoRa was quite large (up to 2400 slots). In this work, we aim at considerably reducing the frame sizes. In fact, one of the main limitations found in the literature is the fixed or large frame sizes of the proposed TDMA approaches.**

The main contributions of this paper can be summarized as follows:

- We propose a novel three-mode decentralized TDMA-based MAC protocol that enables nodes to independently determine their transmission parameters without relying on downlink traffic from the gateway. Moreover, the protocol incorporates dynamic frame sizes and slot durations adapting to the dynamic nature of LoRa networks.
- Through extensive simulation studies, we evaluated the performance of the three modes of our proposed protocol and compared it to the LoRaWAN and SBTS-LoRa protocols in terms of collision probabilities, end-to-end delays, throughput, and energy efficiency under large-scale deployments.

The remainder of this paper is structured as follows. Section II surveys the recent related work. Section III provides a comprehensive explanation of the proposed protocol and its three distinct modes. Results and discussion are given in Section IV. Finally, Section V concludes the paper by summarizing the key findings and contributions.

II. Related Work

To address the primary drawbacks of LoRaWAN, a number of TDMA-based solutions have been proposed in the literature [10] that can be broadly classified into synchronized-based, beacon-based, and autonomous-based.

In beacon-based approaches, the gateway sends beacons to the nodes that contain control information at the beginning of each frame. These beacons are used by nodes to set up their transmission parameters and timeslots. Different beacon-based TDMA protocols have been proposed in the literature. Examples of these protocols include RT-LoRa [14], DG-LoRa [15], FCA-LoRa [16], and TS-VP-LoRa [17]. All these protocols divide the time into successive frames and a specific duration is assigned for receiving beacons at the beginning of each frame. During the beacon period, nodes will be in a listening mode and will not begin transmitting until they receive a beacon. Hence, the duration of the beacon period has a great impact on the energy consumption of nodes. In the RT-LoRa [14] protocol, for instance, multiple beacons are transmitted by the gateway during its beacon slot using different spreading factors (SFs) to enable nodes to determine their optimal SF. While this approach helps nodes reduce their packet error rate (PER), it necessitates a longer beacon period, leading to increased energy consumption. FCA-LoRa [16] on the other hand, employs fixed frame and timeslot durations, which is not suitable for the dynamic nature of LoRa networks due to the varying ToA caused by different payload sizes and SFs. In contrast, TS-VP-LoRa [17] proposes variable slot durations based on the payload size and SF, but its fixed frame duration results in a fixed number of timeslots per SF.

In synchronized-based approaches, which are more compatible with LoRaWAN Class A devices, nodes send synchronization requests during their synchronization phases to initiate transmissions. Examples of this approach include [18], [19], [20], and [21] in which extensive downlink traffic is required from the server, to handle the synchronization, which might be limited by the gateway duty cycle. Xu et al. in [20] proposed a synchronization-based TDMA protocol named S-MAC algorithm based on the assumption of SF orthogonality and traffic periodicity of nodes. The main idea is to use a maximization algorithm to assign the nodes with the same SF different channels or time offsets to mitigate collisions among them. Garrido-Hidalgo et al. in [21] proposed another synchronization-

based TDMA protocol based on an online multi-agent approach to assign transmission parameters to nodes.

In autonomous-based approaches, nodes independently determine their transmission parameters, including SF, channel, and timeslot, without the need for server-side scheduling. This is achieved by providing nodes with basic information about their surroundings, allowing them to make informed decisions. For instance, nodes derive their time slots from their MAC addresses and DeVAddr in[22] and TS-LoRa [8]. However, the two approaches require frame length information broadcasted by the server, limiting their autonomy. Moreover, using MAC addresses may not guarantee unique timeslots and both approaches require large frame sizes, leading to longer waiting times and lower throughput. **Lasri et al. in [23] proposed that connected nodes autonomously control their traffic to mitigate collisions in high-density networks. To accomplish this, they operate under the assumption that the server possesses advance knowledge of the number of required transmissions per node within a predetermined time-frame and each node initiates transmissions with a specified probability.** In SBTS-LoRa [13], nodes independently determine their transmission parameters, including timeslots, based on their proximity to the gateway. **Specifically, SBTS-LoRa is a MAC protocol that allows nodes to determine their transmission parameters autonomously based on their location relative to the gateway. Specifically, based on nodes' distance to the gateway, they can individually determine their transmission parameters and timeslots. SBTS-LoRa leverages some tools from the geometry of circles to determine the timeslots of nodes.** This has been proven to enhance the scalability, however, it still suffers from large frame sizes, especially with high SFs.

To overcome the limitations of the existing solutions, this paper introduces a TDMA-based approach that enables nodes to self-determine their transmission parameters based on their proximity to the gateway. By providing each node with its location and the location of its assigned gateway, nodes can choose the minimum eligible SF that ensures successful packet delivery. Additionally, the proposed algorithm guarantees a minimum frame size that adheres to nodes' duty cycles. Further details are provided in the subsequent section.

III. PROTOCOL DESCRIPTION

In this study, we consider a single-hop star network topology with a central gateway and randomly distributed nodes within a field of radius R. Communications between the nodes and the gateway are 1-hop. **As mentioned in Section I, LoRa regulates the access to the shared medium using duty cycle. In this study, all nodes on all used channels use a duty cycle of 1% as declared in Table I.** .

As mentioned in section I, higher spreading factors (SFs) have longer transmission ranges but also incur longer time on air (ToA) which may result in higher collision rates and longer end-to-end delays. Thus, using smaller SFs whenever possible is recommended as it may increase the network throughput. Considering that, this work proposes dividing the network field into six coronas, corresponding to the six supported SFs in LoRa as demonstrated in Fig.1. Each corona C_i is assigned a dedicated SF, and its zone represents the range over which that SF can be safely utilized. The eligible range of an SF is defined as the maximum distance from the gateway at which nodes can use that SF without their transmissions falling below the gateway's sensitivity level. This maximum distance is represented by the radius R_i of each corona. In essence, packet transmissions from nodes in a particular corona will not be lost due to the gateway's sensitivity limitations. Accordingly, determining the appropriate SF for each node necessitates accurately predicting SF ranges, which is **explained in Section III-A. Then, a detailed description of our proposed mechanism to outperform the SBTS-LoRa [13] protocol is conducted in Section III-B.**

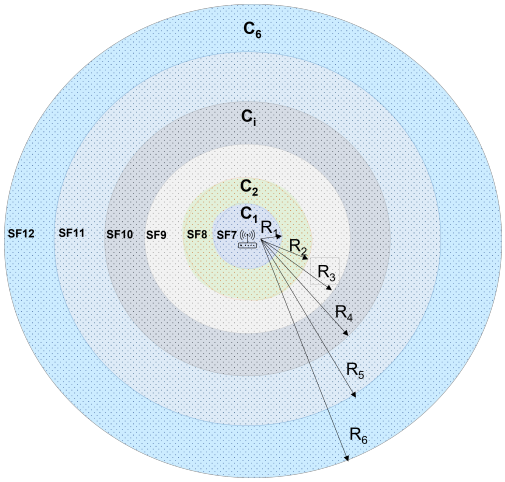

Fig. 1. Dividing the field into overlapping coronas with variable SFs.

A. Determining the SF transmission parameter

ATS-LoRa divides the network field into six coronas, which matches the number of supported SFs in LoRa. A specific SF is allocated to each corona C_i. The smallest SF, SF7, is given to the closest corona to the gateway. Coronas are assigned higher SFs as we move away from the gateway. The longest eligible distance for a node to use a particular SF such that its transmission is not received below the sensitivity threshold of the gateway is defined as the radius R of a given corona C_i. Hence, Packet Errors are mitigated. A node will compare its distance with the radius R_i of each corona C_i starting with the smallest corona to determine to which corona it belongs and thus its appropriate SF for its transmissions. A node n considers itself to belong to corona C_i if the following condition is met: $R_{i-1} < dist_n \leq R_i$, where $dist_n$ is the distance between node n and the gateway. particularly, a node n will select the SF of corona C_i with a R_i that is greater than its distance $dist_n$. By doing so, receiving packets below the sensitivity level of the gateway is completely avoided. We assume that each node knows its coordinates as well as the coordinates of the associated gateway in order to calculate its distance to the gateway. In addition, each node keeps a vector of R_i values, with a total length of six, for comparison with its distance. It is worth noting that the process of determining

the corresponding corona for nodes, and thus the appropriate SF, is done only once for static nodes during their network joining phase, which is the case in most of IoT applications. Furthermore, SF ranges and hence corona radiuses are not fixed values and vary greatly depending on the environment. Once a node has determined its SF, it will use one of the following three approaches to determine its channel frequency CF and time-slot SID.

1) Equal Radius-Based Corona (ERBC) mode: In this mode, we further divide each corona C_i into K equal-width sub-coronas. Then, each sub-corona is labeled with an identifier F_j and assigned a unique channel CF_j, as shown in Fig. 2. According to ERBC mode, a node n that is located on corona C_i and sub-corona F_j will use the SF_i and the frequency CF_j that corresponds to the SF assigned to corona C_i and the channel assigned to sub-corona F_j, respectively. The goal of this partitioning is to mitigate collisions among nodes that are located on a given corona, by spatially distributing frequencies among them. To do so, we firstly find the width L_i of sub-coronas of a given corona C_i as follows:

$$L_i = \frac{R_i - R_{i-1}}{K} \tag{1}$$

where R_i and R_{i-1} are the radii of corona C_i and C_{i-1} respectively and K is the number of the available frequencies. In order for a node n to determine its F_j, and hence its appropriate frequency CF_j, it will use the following formula

$$F_j = \frac{dist_n - R_{i-1}}{L_i} \mod K \tag{2}$$

where $dist_n$ is the distance of node n from the gateway. We assume that each channel is labeled with an ID F_j and once a node determines the channel ID F_j, it can determine the corresponding channel frequency CF_j. For example, in Fig. 2, node n will use the frequency CF_j that is assigned to the sub-corona F_j because it is located in the range of that sub-corona.

Note that nodes located in the same corona and the same sub-corona are using the same SF and the same channel which may cause collisions among simultaneous transmissions. To avoid this, we aim to assign these nodes different time slots

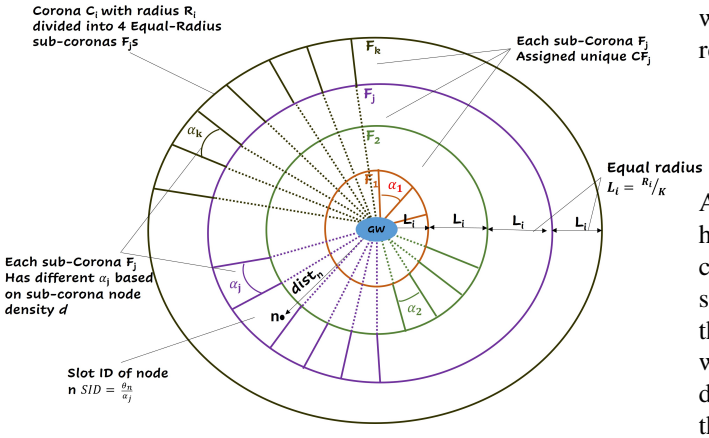

Fig. 2. Equal Radius-Based Corona (ERBC) mode example of Corona C_i with radius R_i

in order to enhance the network throughput. Thus, in order to determine the slot-ID SID for a node, we separate communications among nodes on the same sub-corona F_j using slots with a slot duration compatible with the used SF in corona C_i. In fact, partitioning nodes on the same sub-coronas into different time slots emulates dividing them into sectors with specific angle α_j. The number of needed slots m_j on every sub-corona F_j depends on the node density d of that sub-corona. Furthermore, the frame size, which is $m_j \times ToA_{SF}$, must respect the duty cycle of nodes, which is 1%. To do that, m_j must has a minimum value of 100 as follows

$$m_j = \begin{cases} 100, & \text{if } m_j < 100 \\ A_{F_j} \times d, & \text{if} m_j \geq 100 \end{cases} \tag{3}$$

where A_{F_j} is the area of sub-corona F_j and d is the network density. After a node has determined the appropriate m_j, it will simply find the sector's angle $\alpha_j = 2\pi/m_j$. Finally, a node determines its slot-ID SID as follows

$$SID = \frac{\theta_n}{\alpha_j} \tag{4}$$

where θ_n is the node angle relative to the associated gateway. For example, node n in Fig. 2 uses $SID = 3$ based on its θ_n to the gateway.

As shown in Fig. 2, each sub-corona F_j has different area and hence different node density. According to that, the number of needed slots m_j for each sub-corona F_j is different. In other words, inner sub-coronas have smaller area and hence smaller m. However, outer sub-coronas have larger areas and hence larger m.

2) Equal Area-Based Corona (EABC) mode: In this mode, each corona C_i is divided into K sub-coronas such that all sub-coronas of a given corona C_i have the same area and hence the same node density. Similar to ERBC mode, each sub-corona F_j is assigned a unique channel CF_j such that all nodes that are located on the range of sub-corona F_j will use channel CF_j. By doing that, we can achieve a balanced traffic load of a given SF in all channels. In order to have equal area sub-coronas, each sub-corona will have a different radius r_j. A node n is considered in the range of sub-corona F_j if it satisfies the following condition: $r_{j-1} < dist_n \leq r_j$, where r_j and r_{j-1} are the radii of sub-corona F_j and F_{j-1}, respectively. r_j is calculated as follows

$$r_j = \sqrt{\frac{F_j(R_i^2 - R_{i-1}^2)}{K} + R_{i-1}^2} \tag{5}$$

As shown in Fig. 3, all sub-coronas have the same area and hence the same node density. According to that, all sub-coronas F_j of a given corona C_i need the same number of slots m. Hence, as depicted in Fig. 3, all sub-coronas have the same number of slots m. Unlike ERBC mode (Fig. 2), where each sub-corona has a different area, and hence different density leading to a different number of slots m. After finding the value of r_j, each node can simply find the area of the sub-corona located in A_{F_j}. Then, similar to ERBC mode, Eq.3 and Eq.4 are used to calculate m_j and SID, respectively.

Algorithm 1 ATS-LoRa MAC protocol

Input: node coordinates (X_n,Y_n), Gateway coordinates (X_G, Y_G), number of channels K, and node density d
Output: The Sreading Factor SF_i, the Channel CF_j, and the timeslot SID for a given node n
$CFs \leftarrow [CF_1, CF_2, .., CF_K]$
$R_i \leftarrow [R_1, R_2, .., R_i]$
$dist_n \leftarrow euclideanDistance(X_n, Y_n, X_G, Y_G)$
# Determine node's SF SF_i
for $i \leftarrow 1$ to 7 **do**
  if $dist_n \leq R[i]$ **then**
    $SF_i \leftarrow (i-1)+7$
    $C_i \leftarrow i$
    $break$
  end if
end for
if ERBC mode **then**
  $L_i \leftarrow (R_i - R_{i-1})/K$
  # Find sub-corona ID F_j
  $F_j \leftarrow \frac{dist_n - R_{i-1}}{L_i} \mod K$
  # Find area of sub-corona F_j
  $A_{F_j} \leftarrow \pi((R_{i-1} + F_j \times L_i)^2 - (R_{i-1} + F_{j-1} \times L_i)^2)$
else
  # EABC mode
  $F_j \leftarrow FindChannelID(dist_n, C_i, 1, K)$
  $r_{F_j} \leftarrow \sqrt{\frac{F_j(R_i^2 - R_{i-1}^2)}{K} + R_{i-1}^2}$
  $r_{F_{j-1}} \leftarrow \sqrt{\frac{F_{j-1}(R_i^2 - R_{i-1}^2)}{K} + R_{i-1}^2}$
  $A_{F_j} \leftarrow \pi(r_{F_j}^2 - r_{F_{j-1}}^2)$
end if
$CF_j \leftarrow CFs[F_j]$
$m_j \leftarrow A_{F_j} \times d$
if $m_j < 100$ **then**
  $m_j \leftarrow 100$
end if
$\alpha_j \leftarrow 2\pi/m_j$
Calculate node's theta to the gateway θ_n
$\theta_n \leftarrow \arctan(Y_n - Y_G / X_n - X_G)$
$SID \leftarrow \theta_n / \alpha_j$

3) Equal Area-Based Sector (EABS) mode: In this mode, instead of dividing coronas into sub-coronas, either with the same width (ERBC mode) or with the same area (EABC mode), we divide the network field into K sectors with angle $\beta = 2\pi/K$ and assign a channel to each sector. Fig.4 shows an example of EABS mode. All nodes located in sector Sec_j will use channel CF_j that is assigned for that sector.

Similar to EABC mode, EABS mode divides the network field into sectors with identical areas. Hence, the traffic load is balanced among the channels. As shown in Fig. 4, $Corona - Sector_{ij}$ refers to the intersection between corona C_i and sector Sec_j. All nodes in the same $Corona - Sector_{ij}$ will use the same SF_i assigned to corona C_i and the same channel CF_j assigned to sector Sec_j. To control channel access between them, we divide each $Corona - Sector_{ij}$ into a grid of $q_{ij} = \sqrt{m_{ij}}$ rows and columns. We assume that each square, which is the intersection of a row and a column, has an identifier ranging from 1 to m_{ij}, which defines the time-slot ID SID. In this mode, m_{ij} is calculated as follows

$$m_{ij} = \begin{cases} 100, & \text{if } m_{ij} < 100 \\ \frac{\beta}{2}(R_i^2 - R_{i-1}^2) \times d, & \text{if} m_{ij} \geq 100 \end{cases} \quad (6)$$

To determine node's SID, a node must firstly determine which square it is located in. Hence each node computes its row_n and col_n as follows

$$row_n = \frac{q_{ij}(dist_n - R_{i-1})}{R_i - R_{i-1}}$$
$$col_n = \theta_n \mod \frac{\beta}{q_{ij}} \quad (7)$$

Then, time-slot SID can be retrieved as follows

$$SID = (row_n - 1) \times q_{ij} + col_n \quad (8)$$

Further details about EABS mode can be found in [24].

According to the proposed algorithms, each channel has six frames, corresponding to the number of the available SFs as shown in Fig. 5. The frame size of a given SF on a given channel depends on the node density of the area that uses that channel with that SF. In other words, the number of needed slots m_j differs among channels for a given SF depending on

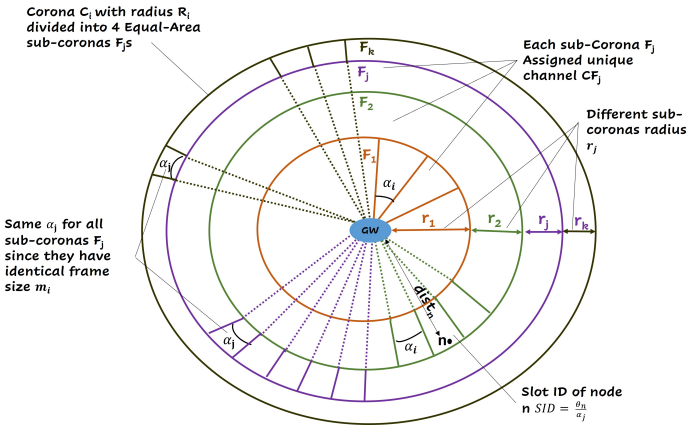

Fig. 3. Equal Area-Based Corona (EABC) mode example of Corona C_i with radius R_i

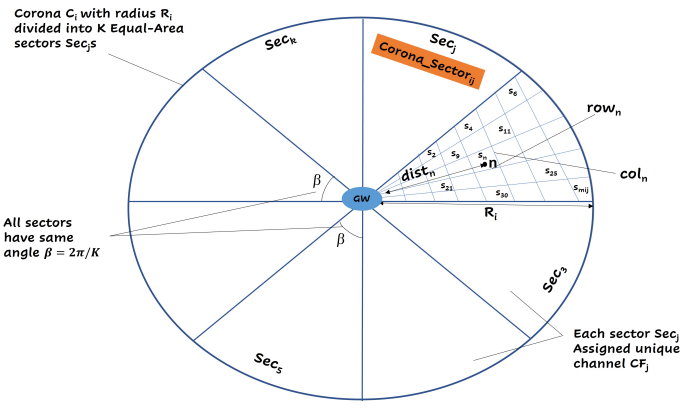

Fig. 4. Equal Area-Based Sector (EABS) mode example of Corona C_i with radius R_i

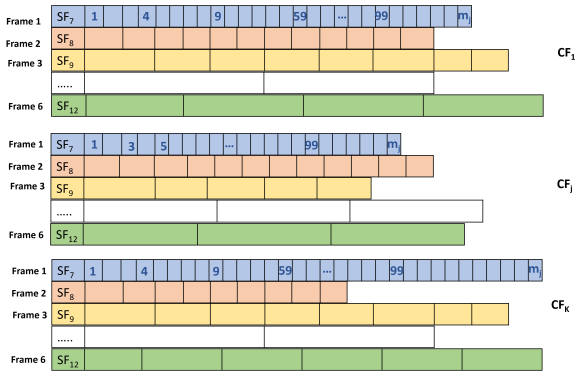

Fig. 5. Time frames per channel CF

the node density of the area that is assigned a given channel CF_j. For example, as shown in Fig. 5, the number of needed slots m_j for SF_7 is different from one channel to another. Furthermore, the slot duration is different from one frame to another depending on the used SF. Hence, the proposed protocol is completely dynamic as it has dynamic frame sizes and dynamic timeslot duration.

B. Frame sizes comparison

In ATS-LoRa protocol, with the three operational modes, the increase in the frame size is not related to the increase in the total number of connected nodes. Alternatively, the increase of the frame size is related to the increase in the number of nodes that use common transmission parameters, such as common SF and channel CF, so to mitigate collisions among them. Fig.6 shows a comparison between the frame size of SBTS-LoRa protocol [13] and the proposed protocol with its three modes. To better clarify the frame sizes, Fig.6.a shows the frame sizes of ATS-LoRa in ERBC, EABC, and EABS modes as a function of the number of connected nodes. Fig.6.b shows the frame sizes of the SBTS-LoRa protocol as a function of the number of connected nodes. The number of nodes is ranging from 1000 to 10000 nodes. The field radius is around 9 Km. Fig.6 shows the frame sizes of the last and largest corona that has $SF = 12$. As shown in Fig. 6.a, the frame size of all three modes is identical for small networks (< 2000). However, for larger number of connected nodes, ATS-LoRa with EABC or EABS modes has lower frame sizes than the ATS-LoRa with ERBC. This is because the former modes divide the coronas into equal-area sub-coronas or sectors, respectively. Hence, the number of nodes that use common SFs and channels and thus require timeslots separation, is more balanced. In this simulation, we consider the last corona because it is the most challenging one as it has the highest SF with the longest end to end delay and highest energy consummation. Hence, with a payload size equals $50B$ and a number of connected nodes $N = 10000$, the waiting time between two successive transmissions is only 15.7 minutes in EABC mode and 17 minutes in ERBC and EABS modes.

IV. Performance Evaluation

In this section, ATS-LoRa with its three modes is evaluated using OMNET++ simulator [25] and FLoRa framework [26]. The nodes are distributed at random around the gateway, with a maximum distance corresponding to the field radius R. The radius of the field and the radius of SF coronas R_i are field-dependent, which means that they vary depending on the deployed environment. Accordingly, this simulation assumes a suburban environment and employs the well-known log-distance path loss model with shadowing [27] as expressed in the following equation

$$PL(d) = \overline{PL}(d_0) + 10n \log\left(\frac{d}{d0}\right) + X_\sigma \qquad (9)$$

where $\overline{PL}(d_0)$ is the mean path loss for distance d_0, n is the path loss exponent, and X_σ is a zero-mean Gaussian distributed random variable with standard deviation σ. The values used in the simulations for these parameters are shown in Table I. **As shown in Table I, eight frequency channels are used in this research, the three default channels that must be implemented in every LoRa node namely, 868.1 MHz, 868.3 MHz, 868.5 MHz, and the optional frequency channels, which are 867.1 MHz, 867.3 MHz, 867.5 MHz, 867.7 MHz, and 867.9 MHz. All these channels support all spreading factors, ranging from SF7 to SF12 with a bandwidth of 125 kHz. Furthermore, the duty cycle of all these channels is 1% [7] [28].** ATS-LoRa protocol assumes a single packet transmission per frame with a packet length of 20 bytes. The protocol imposes a minimum frame size of $100 \times ToA_{SF}$ as explained in section III. However, there is no limit for the maximum frame size, as the frame size remain reasonable even with large number of connected nodes, as demonstrated in section III-B, thanks to the possible parallel frames per channel and SF.

TABLE I
Simulation Parameters

Parameter	Value	Comments
$\overline{PL}(d_0)$	128.95	Mean path loss for distance $d_0 = 1000m$
n	1.5	Path loss exponent
σ	0	
CF Carrier Frequencies (MHz)	{868.1, 868.3, 868.5, 867.1, 867.3, 867.5, 867.7, 867.9}	1% duty cycle
SF	7 to 12	Spreading factors
TP	14 dBm	Transmission powers
CR	4/5	Coding rate
BW	125kHz	Bandwidth
R	8921m	Field radius
N	1000 - 4000	Number of nodes
Simulation time	11	Days

According to the proposed protocol, the network field is divided into six coronas that are assigned, each, a given SF. Each corona's radius corresponds to the maximum eligible distance for nodes to use the assigned SF. We conduct a simulation using OMNET++ simulator to determine the maximum eligible distance for a given SF such that the corresponding

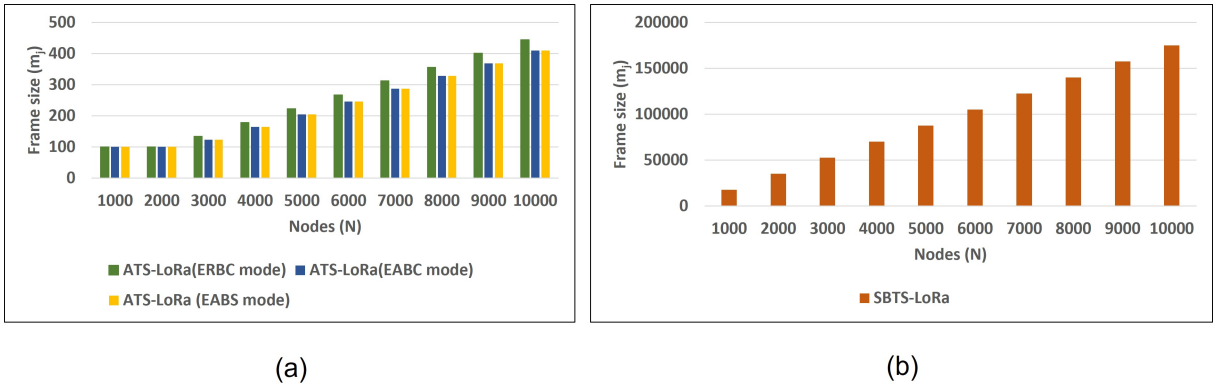

Fig. 6. Comparison between the frame sizes of ATS-LoRa protocol in ERBC, EABC, and EABS modes and SBTS-LoRa protocol [13]

transmissions are not received below the sensitivity level of the gateway. Specifically, the simulation consists of one node and one gateway. Firstly, a node is assigned a given SF value starting from the smallest one, SF7, to the largest one, SF12. Then, on each run, the node is moving farther from the gateway while sending packets to the gateway. The maximum eligible distance for a given SF is the maximum distance for the node in which the gateway can successfully receive packets above the gateway sensitivity level. The simulation is repeated for all the SF values. Table II exhibits the maximum allowable distance for each SF with a transmission power equals 14 dBm and a path loss model parameters as indicated in Table I. Furthermore, Table II shows the sensitivity levels that is configured at the gateway as specified in [29]. The following sections evaluate the performance of the proposed protocol with its three modes compared to SBTS-LoRa [13] and LoRaWAN-ADR in terms of the probability of collision, the end-to-end delay, the network throughput, and the energy consumption.

A. The probability of collisions

Fig.7 shows the probability of collision as function of the total number of connected nodes. **Furthermore, Eq.10 demonstrates how the probability of collision is measured for this simulation.**

$$Collisions_{Prob} = \frac{Col_{PKT}}{(Col_{PKT} + RCV_{PKT})} \tag{10}$$

where Col_{PKT} and RCV_{PKT} are the total number of collided packets and the total number of received packets during the simulation time, respectively. As shown in Fig.7, ERBC and EABC modes achieves the lowest collision rates. On the other hand, EABS mode achieves the highest collision rate that close to the collision rate of LoRaWAN. This is due to the slots assignment procedure among nodes. In fact, in all modes, the total number of the available time-slots per channel m_{ij} is related to the estimated node density of a given sub-corona or $Corona - Sector_{ij}$ (see Eq.3 and Eq.6). However, in ERBC, EABC and SBTS-LoRa, the assignment of time-slots corresponds also to node's relative position angle "theta" to the gateway. Hence, there is a low probability to have two or more nodes on the same sub-corona and with the same "theta". As shown in Fig.7, the collision rate is zero in ERBC and EABC modes with $N = 1000$, which is a quite large number of nodes. Furthermore, in ERBC and EABC modes, the collision rates are slowly increasing with the increase of the number of connected nodes compared to EABS mode. In other words, although each $Corona-Sector_{ij}$ in EABS mode have a frame size that corresponds to its need, similar to ERBC and EABC modes, two or more nodes can have the same time-slot ID SID if they are located on the same square (see Eq.8). In other words, the assignment of time-slot IDs does not depend on nodes' theta like other modes. In EABS mode, the time-slot assignment depends on which square a node is located. In fact, the probability of having more nodes with the same SID is increasing with the increase of the number of nodes. Furthermore, nodes start their transmissions usually at the beginning of their time-slots without any random back off. Hence, destructive collisions are happened with nodes located on the same $Corona - Sector_{ij}$ and use the same SID. This explains the highly increasing collision rate of EABS mode.

TABLE II
MAX ELIGIBLE DISTANCE FOR EACH SPREADING FACTOR SF[29]

SF	MAX eligible distance (m)	Sensitivity (S)
7	2450	-124
8	3306	-127
9	4450	-130
10	5998	-133
11	7316	-135
12	8921	-137

B. The end-to-end delay

In this simulation, the end-to-end delay is calculated as the difference between the start time of a packet transmission and the end time of its reception for all successfully received packets by the gateway. Eq. 11 shows the end-to-end delay in seconds

$$End\_Delay(s) = \frac{\sum(End\_time_{PKT} - Start\_time_{PKT})}{RCV_{PKT}} \tag{11}$$

where $Start\_time_{PKT}$ and End_time_{PKT} are the start time of a packet transmission and the end time of its

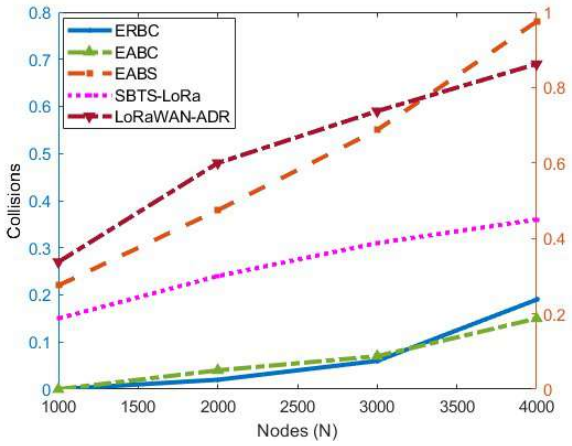

Fig. 7. Probability of collisions

reception, respectively. Fig.8 shows the end-to-end delay as function of the number of connected nodes. As shown in Fig. 8, all modes of the ATS-LoRa protocol achieve lower end-to-end delay than SBTS-LoRa and LoRaWAN protocols. In fact, the used SF of each node has a significant impact on the end-to-end delay of their transmissions. Smaller SFs have shorter ToA and thus less delay. As a result, from a delay standpoint, it is preferable to use lower SFs whenever they are allowable. To better explain the end delay, Fig.9 shows the distribution of SFs among nodes in both ATS-LoRa and SBTS-LoRa protocols. As shown in the figure, SBTS-LoRa algorithm use larger SFs more than the ATS-LoRa. This explains the longer delay of SBTS-LoRa algorithm. Furthermore, the end-to-end delay is only considered for the successfully received packets by the gateway. That's why the end-to-end delay decreases with large number of nodes as the number of successfully received packets will be reduced due to high collision rate. More precisely, the closest nodes to the gateway will be the ones that will succeed to deliver their packets to the gateway as they have the shortest time on air which will explain the reduced end-to-end delay. Hence, EABS mode achieves the lowest end to end delay as it has the highest collision rate.

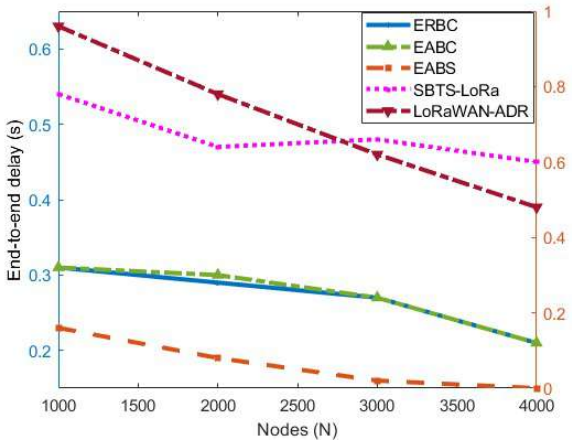

Fig. 8. The end-to-end delay

C. Network throughput

Network throughput can be defined as the total number of received packets during the whole simulation time as demonstrated in Eq. 12

$$Throughput(Packets/s) = \frac{RCV_{PKT}}{Sim\_time} \tag{12}$$

Fig.10 shows the network throughput as function of the number of connected nodes. The results show that the end-to-end delay and collision rate have a significant impact on network throughput. Indeed, as more traffic is generated in the network, the network throughput increases in proportion to the number of connected nodes until it reaches a maximum. The network throughput then begins to deteriorate as the probability of collision takes over the successfully received packets, resulting in a decrease in network throughput. As shown in Fig.10, The ATS-LoRa protocol outperforms the SBTS-LoRa and LoRaWAN protocols in terms of throughput. Specifically, ERBC and EABC modes achieve the highest network throughput since they have the lowest collision rate. In ERBC and EABC modes, the network throughput reaches 54 packets/second when the number of connected nodes equals 3000 nodes. The optimal throughput results can be attributed to the efficient distribution of the transmission parameters that resulted in the lowest collision rate and end-to-end delay. On the other hand, although the EABS mode has the worst collision rate among other ATS-LoRa modes, it still has better network throughput than SBTS-LoRa and LoRaWAN protocols due to the efficient distribution of SFs compared to them as ATS-LoRa protocol uses the minimum SF whenever possible. Hence, the Time on Air (ToA) is reduced. Furthermore, ATS-LoRa has smaller frame sizes compared to SBTS-LoRa protocol, as shown in Fig.6. Hence, the waiting time between successive data transmissions is reduced.

D. The energy consumption

Fig.11 depicts the energy required to successfully transmit one bit as a function of the network size. Eq. 13 shows the total energy consumed to deliver one successful bit of a payload

$$EpB(J) = \frac{Total\_Energy}{RCV_{PKT} \times Payload} \tag{13}$$

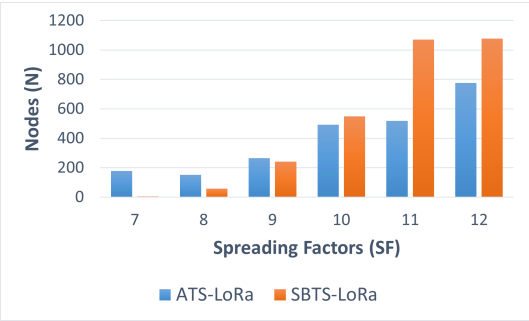

Fig. 9. Comparison between SF distribution in SBTS-LoRa and ATS-LoRa protocols

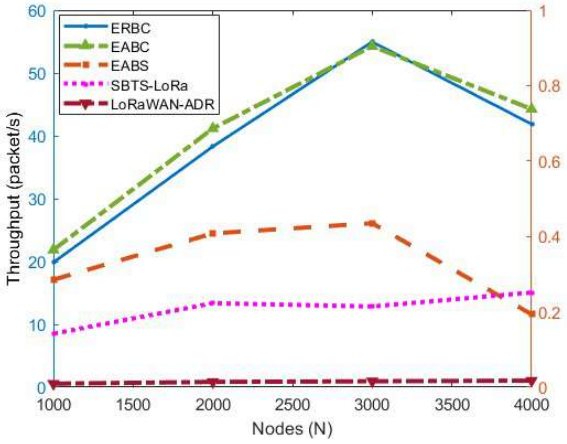

Fig. 10. Network throughput

where $Total\_Energy$ **is the total energy consumed in** J **and** $Payload$ **is the packet payload in bits. As shown in Fig.11,** the ATS-LoRa protocol with EABS mode consumes the highest energy since it has the highest collision rate. ERBC and EABC modes of ATS-LoRa protocol achieve similar energy consumption levels, which is expected, as they have similar collision rates, end-to-end delay, and network throughput. It is, however, notable that ERBC and EABC modes are showing almost stable energy consumption with the increase in the number of connected nodes. This behaviour is critical as it demonstrates the scalability and robustness of these modes. The results also show that ATS-LoRa protocol consumes more energy compared to the SBTS-LoRa protocol. However, this is justified as the number of successfully delivered packets is higher in the ATS-LoRa protocol,as shown in Fig.10. We assume one packet transmission per frame in this simulation. Unlike SBTS-LoRa, the frame size in ATS-LoRa is determined by the node density of a given sub-corona and is not fixed. As a result, the ATS-LoRa protocol has a small frame size without compromising the node's duty cycle.

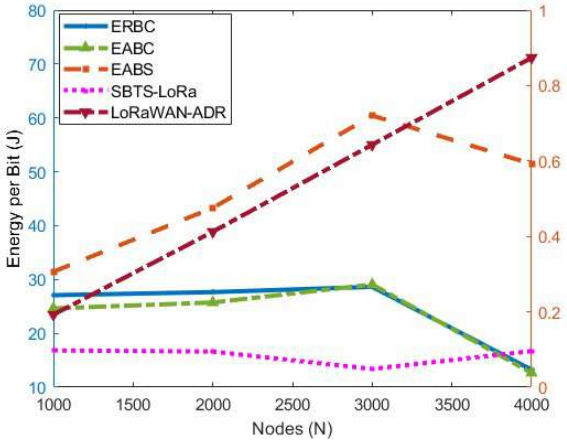

Fig. 11. Energy consumption

V. Conclusion

This paper presents a novel TDMA MAC protocol for LoRa networks that autonomously configures nodes with the appropriate transmission parameters (e.g., SF, Channel, times-lot) utilizing the node's relative position to the gateway. To achieve this, the proposed protocol divides the network field into sections of corona and sectors and assigns specific parameters for each one of them. A node located in a specific range of corona or sector will use the parameters that are virtually assigned to that section. The corona radius is dynamic and reflects the maximum allowable distance for a node to use a given SF such that its transmission is received above the sensitivity level of the gateway. According to that, the Packet Error Rate (PER) in the proposed protocol is almost zero. The assignment of channels and timeslots is done using three variant approaches, namely, Equal Radius-based Corona (ERBC), Equal Area-based Corona (EABC), and Equal Area-based Sector (EABS) showing outperforming results in terms of end-to-end delay and throughput compared to SBTS-LoRa and ADR LoRaWAN. For future work, further improvements could be applied to enhance the approach in determining the optimal SF ranges. In other words, we aim at leveraging some AI techniques to determine the optimal SF for each node without compromising the end-to-end delay, the collision rate, and the energy consumption.

References

[1] G. Premsankar, B. Ghaddar, M. Slabicki, and M. D. Francesco, "Optimal Configuration of LoRa Networks in Smart Cities," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7243–7254, Dec. 2020, conference Name: IEEE Transactions on Industrial Informatics.

[2] F. Deng, P. Zuo, K. Wen, and X. Wu, "Novel soil environment monitoring system based on RFID sensor and LoRa," *Computers and Electronics in Agriculture*, vol. 169, p. 105169, Feb. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169919319222

[3] S. W. Prakosa, M. Faisal, Y. Adhitya, J.-S. Leu, M. Köppen, and C. Avian, "Design and Implementation of LoRa Based IoT Scheme for Indonesian Rural Area," *Electronics*, vol. 10, no. 1, p. 77, Jan. 2021, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2079-9292/10/1/77

[4] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7815384/

[5] "About LoRaWAN® | LoRa Alliance®." [Online]. Available: https://lora-alliance.org/about-lorawan

[6] Z. Sun, H. Yang, K. Liu, Z. Yin, Z. Li, and W. Xu, "Recent Advances in LoRa: A Comprehensive Survey," *ACM Transactions on Sensor Networks*, Jun. 2022, just Accepted. [Online]. Available: https://doi.org/10.1145/3543856

[7] Semtech, "LoRaWAN 1.1 Regional Parameters," 2017.

[8] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things," *Computer Communications*, vol. 153, pp. 1–10, Mar. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0140366419314677

[9] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8030482/

[10] D. Zorbas and X. Fafoutis, "Time-Slotted LoRa Networks: Design Considerations, Implementations, and Perspectives," *IEEE Internet of Things Magazine*, vol. 4, no. 1, pp. 84–89, Mar. 2021, conference Name: IEEE Internet of Things Magazine.

[11] K. Z. Islam, D. Murray, D. Diepeveen, M. G. K. Jones, and F. Sohel, "LoRa-based outdoor localization and tracking using unsupervised symbolization," *Internet of Things*, vol. 25, p. 101016, Apr. 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660523003396

[12] B. Li, Y. Xu, Y. Liu, and Z. Shi, "LoRaWAPS: A Wide-Area Positioning System Based on LoRa Mesh," *Applied Sciences*, vol. 13, no. 17, p. 9501, Jan. 2023, number: 17 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3417/13/17/9501

[13] H. Alahmadi, F. Bouabdallah, and A. Al-Dubai, "A novel time-slotted LoRa MAC protocol for scalable IoT networks," *Future Generation Computer Systems*, vol. 134, pp. 287–302, Sep. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X22001261

[14] L. Leonardi, F. Battaglia, and L. L. Bello, "RT-LoRa: A Medium Access Strategy to Support Real-Time Flows Over LoRa-Based Networks for Industrial IoT Applications," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 812–10 823, Dec. 2019, conference Name: IEEE Internet of Things Journal.

[15] J. Lee, Y. S. Yoon, H. W. Oh, and K. R. Park, "DG-LoRa: Deterministic Group Acknowledgment Transmissions in LoRa Networks for Industrial IoT Applications," *Sensors*, vol. 21, no. 4, p. 1444, Jan. 2021, number: 4 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1444

[16] A. Triantafyllou, P. Sarigiannidis, T. Lagkas, I. D. Moscholios, and A. Sarigiannidis, "Leveraging fairness in LoRaWAN: A novel scheduling scheme for collision avoidance," *Computer Networks*, vol. 186, p. 107735, Feb. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620313232

[17] A. Triantafyllou, D. Zorbas, and P. Sarigiannidis, "Time-slotted LoRa MAC with variable payload support," *Computer Communications*, vol. 193, pp. 146–154, Sep. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422002444

[18] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low Overhead Scheduling of LoRa Transmissions for Improved Scalability," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3097–3109, Apr. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8516298/

[19] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch, "$FREE$ —Fine-Grained Scheduling for Reliable and Energy-Efficient Data Collection in LoRaWAN," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 669–683, Jan. 2020, conference Name: IEEE Internet of Things Journal.

[20] Z. Xu, J. Luo, Z. Yin, T. He, and F. Dong, "S-MAC: Achieving High Scalability via Adaptive Scheduling in LPWAN," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Jul. 2020, pp. 506–515, iSSN: 2641-9874.

[21] C. Garrido-Hidalgo, L. Roda-Sanchez, F. J. Ramírez, A. Fernández-Caballero, and T. Olivares, "Efficient online resource allocation in large-scale LoRaWAN networks: A multi-agent approach," *Computer Networks*, vol. 221, p. 109525, Feb. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912862200559X

[22] D. Zorbas and B. O'Flynn, "Autonomous Collision-Free Scheduling for LoRa-Based Industrial Internet of Things," in *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. Washington, DC, USA: IEEE, Jun. 2019, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8792975/

[23] K. Lasri, Y. Ben Maissa, L. Echabbi, O. Iova, and F. Valois, "Probabilistic and distributed traffic control in LPWANs," *Ad Hoc Networks*, vol. 143, p. 103121, Apr. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870523000410

[24] H. Alahmadi, F. Bouabdallah, A. Al-Dubai, and B. Ghaleb, "Autonomous adaptive frame size for time-slotted lora mac protocol," in *International Wireless Communications and Mobile Computing Conference (IWCMC 2023)*, Marrakesh, Morocco, Jun. 2023.

[25] OMNeT++ Discrete Event Simulator, 2018. [Online]. Available: https://omnetpp.org/

[26] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of lora networks for dense IoT deployments," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. Taipei, Taiwan: IEEE, Apr. 2018, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/8406255/

[27] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128601003024

[28] "Frequency Plans." [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/frequency-plans/

[29] S. Corporation, "Sx1272/73 datasheet rev 3.1," 2017. [Online]. Available: shorturl.at/klBDI