PolarScheduler: Dynamic Transmission Control for Floating LoRa Networks

Ruinan Li, Xiaolong Zheng, Yuting Wang, Liang Liu, Huadong Ma
Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia
Beijing University of Posts and Telecommunications, Beijing, China
{liruinan, zhengxiaolong, wangyutingbupt, liangliu, mhd}@bupt.edu.cn

IEEE INFOCOM 2022 - IEEE Conference on Computer Communications | 978-1-6654-5822-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOM48880.2022.9796656

***Abstract*—LoRa is widely deploying in aquatic environments to support various Internet of Things applications. However, floating LoRa networks suffer from serious performance degradation due to the polarization loss caused by the swaying antenna. Existing methods that only control the transmission starting from the aligned attitude have limited improvement due to the ignorance of aligned period length. In this paper, we propose *PolarScheduler*, a dynamic transmission control method for floating LoRa networks. *PolarScheduler* actively controls transmission configurations to match polarization aligned periods. We propose a V-zone model to capture diverse aligned periods under different configurations. We also design a low-cost model establishment method and an efficient optimal configuration searching algorithm to make full use of aligned periods. We implement *PolarScheduler* on commercial LoRa platforms and evaluate its performance in a deployed network. Extensive experiments show that *PolarScheduler* can improve the packet delivery rate and throughput by up to 20.0% and 15.7%, compared to the state-of-the-art method.**

I. Introduction

Internet of Things (IoT) has shown remarkable prosperity in terrestrial environments including both urban and rural areas [1], [2]. Benefiting from the extensive deployment of Low Power Wide Area Network (LPWAN) such as LoRa (Long Range) [3], [4], recent studies are attempting deploying IoT systems in the aquatic environments such as oceans [5] [6], reservoirs [7], and rivers [8]. Lots of researches lately have proved the capability of LoRa to decode packets with low link quality [9]–[12] and deal with multiple packets concurrently [13]–[17]. With the features of wide coverage, high reliability and low power consumption, LoRa offers great potential to support the communication of floating nodes equipped with anchors on the water surface.

Compared to the networks on land, the link reliability of floating LoRa networks experiences significant degradation, which can be more than 20% Packet Delivery Ratio (PDR) decline [18]. This is because the floating node sways with the waves and therefore has a fast-changing attitude. The attitude changes cause misalignment between the linearly polarized antennas of LoRa transceivers, leading to highly dynamic link Signal-to-Noise Ratio (SNR) and consequent packet failures. Different from nodes on land that have stable polarization alignment, the antenna attitude of a floating node keeps changing, resulting in non-negligible link quality instability.

Adaptive control is a conventional method to cope with link dynamics. However, existing adaptive methods such as ADR [19] and DyLoRa [20] fail to protect the transmission reliability of floating LoRa networks because they focus on the occasional link attenuation changes but ignore the fast-changing link quality caused by polarization alignment. Some resource allocation methods such as EF-LoRa [21] and AdapLoRa [22] focus more on the energy fairness and resource consumption for each node, incapable of adapting to the fast-changing environment as well. Due to continuous changing, the SNR estimated from a gateway can vary even 6dB during a single second. PolarTracker [18] is the first work that focuses on the attitude link degradation problem in floating LoRa networks and proposes an attitude-aware channel access method to control the transmission time. The reliability can be improved by scheduling packets start transmitting from the aligned attitude.

However, we find that only controlling the transmission start time can still have poor performance because of the ignorance of the polarization aligned period length. First, reliability improvement can fail due to the mismatch between packet duration and aligned period. Our measurement results reveal that the SNR can vary 8dB during one packet transmission when LoRa adopts a large SF, which is very common for floating LoRa networks that desire a long communication range. Then the latter part of a packet will be transmitted during the misaligned period, leading to failures of the whole packet. Second, the valuable aligned period is underutilized when the duration of a transmitted packet is shorter than the aligned period. Since PolarTracker only schedules packets with predetermined transmission configurations, the mismatch between packet duration and dynamic aligned period leads to low throughput. Therefore, a method that can dynamically control the overall transmission configurations to fit the aligned periods is necessary but missing in existing literature.

In this paper, we propose *PolarScheduler*, an attitude-aware dynamic transmission control method for floating LoRa networks that actively controls transmission configurations to best fit the aligned period. To improve both reliability and throughput, *PolarScheduler* not only keeps transmissions in aligned periods but also fill aligned periods as full as possible. To achieve this goal, we face three challenges. First, it is non-trivial to model the aligned period when actively changing transmission configurations because the aligned period is dependent to LoRa transmission parameters. When using a larger SF, the SNR required for decoding decreases and

the aligned period extends. Second, how to efficiently obtain the alignment model is not clear. Since LoRa has a large number of transmission configurations and consequently many possible alignment models, probing all the configurations is too costly. Third, how to efficiently and accurately select the optimal transmission configuration for floating nodes with dynamic attitudes is challenging. Given the large number of configurations, selecting the optimal configuration by brute-force searching is obviously inefficient.

To solve the challenges, we propose the design of *PolarScheduler*. First, we propose a V-zone model to describe the aligned period with the consideration of transmission configurations. The V-zone model uses the SNR at the best-aligned attitude and the attitude with the minimum required SNR for the current configuration to describe the feasible transmission period. Second, we propose a V-zone model transformation method to infer models of other configurations based on the probing results on one configuration. Note that though the aligned period length varies with configuration changes, the underlaying physical link quality and attitude motion are the same for a given node. Hence, we can calculate the aligned period changes based on the relationship of the required decoding SNR for different configurations. Third, we propose a two-phase optimal configuration selection algorithm to reduce the overhead of searching among huge number of configurations. In the first phase, we propose a ternary search based algorithm to select the best SF based on the V-zone model. Then in the second phase, we decide the appropriate packet length and transmitting timing based on the aligned period determined by the selected SF. After obtaining the optimal configuration, *PolarScheduler* packages packets with the selected packet length and SF and schedules the transmissions into aligned periods, taking both hardware and software delay into consideration.

The contributions of this work are summarized as follows.

- We propose *PolarScheduler*, a novel dynamic transmission control method for floating LoRa networks to improve both reliability and throughput. *PolarScheduler* not only controls the transmission timing but also actively control the configurations of SF and packet length to best fit the polarization aligned period.
- We propose a V-zone model to capture the relationship between polarization aligned periods and transmission configurations. We design an attitude-based model transformation method to reduce the probing overhead. Based on the V-zone model, we further propose an efficient optimal configuration selection method.
- We implement *PolarScheduler* on commodity LoRa devices and evaluate its performance in real-world scenarios. The experimental results demonstrate that *PolarScheduler* can remarkably improve both the reliability and throughput of floating nodes, compared to the default ALOHA and the state-of-the-art method.

The rest of this paper is organized as follows. We introduce the background and motivation of our work in Section II. We then present the V-zone model in Section III and the design of *PolarScheduler* in Section IV. We evaluate *PolarScheduler* in Section V and finally conclude our work in Section VII.

II. Background & Motivation

In this section, we first briefly introduce the basics of LoRa and then discuss the motivation of this work based on our preliminary experiments.

A. LoRa Background

LoRa can configure key parameters to achieve trade-offs among communication distance, data rate, and power consumption [23]–[25]. Among the parameters, Spreading Factor (SF), Bandwidth (BW), and Coding Rate (CR) have the biggest impacts on the transmission reliability and throughput. LoRaWAN is a specification of LoRa communication architecture that adopts pure ALOHA as the Medium Access Control (MAC) protocol. To adapt to different communication environments, LoRaWAN provides a mechanism called Adaptive Data Rate (ADR) [19] which helps to optimize data rates, air-in-time and energy consumption. ADR records the most recent 20 measurements of SNR to monitor the link changes and accordingly controls SF, BW and Transmission Power (TP) of an end device. The difference between the measured SNR and the required SNR for decoding is calculated to decide the set of parameters to apply. ADR can cope with the occasional link quality changes for terrestrial environments, but it cannot deal with the constantly fast-changing link quality in floating LoRa networks.

B. Motivation

Due to the ignorance of SNR jitters caused by the constantly changing polarization alignment state, neither historical or probed SNR can reflect the further link quality. Regarding to the dynamic control dedicated to floating LoRa transmissions, PolarTracker [18] proposes an attitude-aware link quality model that quantitatively describes SNR variations at different attitudes. With the help of the new link quality model, PolarTracker can predict the SNR in the future and then schedule transmissions into the polarization aligned periods.

However, our measurements show that PolarTracker fails to improve the performance when transmitting packets with long duration. First, the reliability cannot be effectively improved when the packet duration is long. We conduct an experiment that the floating sender transmits the LoRa packets with SF=9 and BW=125KHz. Fig. 1 plots the PDR of PolarTracker and LoRaWAN when the packet length varies from 50bytes to 250bytes. We can clearly find that with the increase of packet length, the improvement of PolarTracker is dramatically reduced. Specifically, when the packet length is 250bytes, its PDR improvement is only 4.08%, which can be more than 20% when the packet length is 50bytes. Besides, the PDR of PolarTracker has an increasing variation when the packet length increases.

By in-depth analysis, we find the reason behind the performance degradation is that PolarTracker only controls the

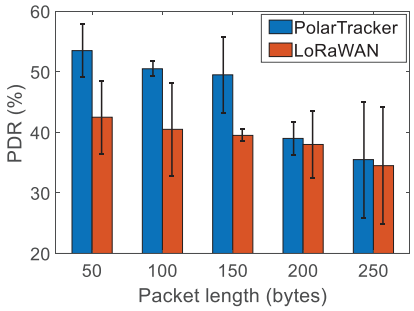


Fig. 1. Reliability degradation of PolarTracker with the increase of packet length.

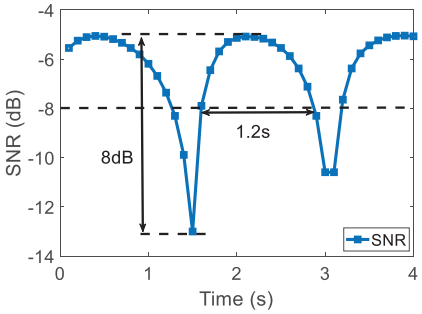


Fig. 2. SNR rapidly changes over time caused by polarization loss.

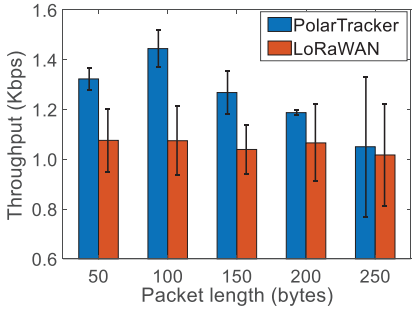


Fig. 3. Throughput of PolarTracker with the increase of packet length.

transmission start time, which cannot guarantee the whole packet is transmitted during the aligned period. When the duration of a packet exceeds the aligned period, the latter part of the packet may suffer from huge signal loss due to polarization misalignment, leading to the reception failure of the whole packet. We plot the SNR variation of a floating node during several seconds in Fig. 2. We can find the SNR can decrease up to 8dB within just one second. Given such a big SNR drop, it is no wonder that the PDR decreases when sending long packets, because the transmission time of a single packet can be 1.25s under the SF and BW setting in our experiment when the packet length is 250 bytes. What's worse, for a floating network, using large SF is very common because a larger SF can lower down the SNR required for decoding and therefore have a longer communication distance. Hence, a packet with long payload and larger SF can even take up to 7s. Only controlling the transmission starting time of the packets with given configurations will fail to improve the reliability.

On the other hand, conservatively using short packets will lead to a low transmission efficiency. Given the fixed preamble overhead, a short packet means less useful payloads can be transmitted. We plot the throughput of the above experiment in Fig. 3. We can find that compared to using the packet length of 50bytes, using the packet length of 100bytes has a lower PDR, but it can improve the throughput by 9.21%, to 1.44kbps. This is because using a short packet length does not make full use of the aligned period. Based on results in Fig. 2, with the experiment settings minimum SNR required for decoding is -8dB, the aligned period can last for 1.2s. But based on our real-world measurement, a packet with the length of 50bytes only takes 0.32s. The remaining 0.88s is wasted.

In a nutshell, existing methods neither guarantee reliability nor improve throughput. The key insight is passively controlling the transmission start time cannot make full use of the good transmission opportunities. If we can actively control the transmission configurations including not only the transmission start time but also the parameters such as SF and packet length, we can fill the aligned period with an appropriate packet to improve the reliability as well as the throughput. Such an idea motivates us to design a new dynamic transmission control method for floating LoRa networks.

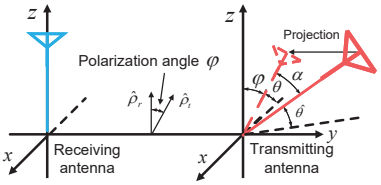


Fig. 4. Illustration of antenna polarization loss.

III. Configuration-aware Link Model

To make full use of the aligned periods, a floating node has to first learn the aligned period length under different transmission configurations. Hence, in this section, we first analyze the link quality under polarization misalignment with the consideration of transmission configurations. We then propose the V-zone model, a configuration-aware link quality model for floating nodes.

A. Polarization Model

In wireless communication, only when the polarization vectors of transmitting and receiving antennas are matched, the receiver can enjoy the largest receiving power. The misalignment between antennas can cause the power loss, which is a property of electromagnetic wave [26]. Existing study [18] has shown that floating LoRa nodes experience continuously changing polarization loss when using linearly polarized antennas.

As shown in Fig. 4, the receiving antenna of the base station usually keeps static and its unit vector of incident wave electric field $\widehat{\rho}_r$ is parallel to the z-axis. Since only the electric field component parallel to the receiving antenna can be induced and received, the swaying will cause a polarization misalignment angle φ, which is the included angle between X-axis and the transmitting vector projected on X-Z plane, whose unit vector is $\widehat{\rho}_t$. When the transmitting antenna sways with waves, $\widehat{\rho}_t$ keeps changing and leads to a dynamic φ. Then the polarization loss with an angle φ can be measured by polarization loss factor K [27], [28].

$$K = |\widehat{\rho}_t \cdot \widehat{\rho}_r|^2 = cos^2\varphi, \tag{1}$$

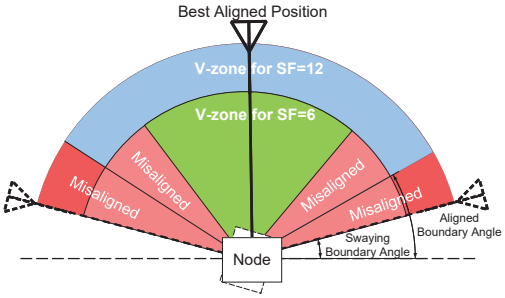


Fig. 5. The V-zone model of different SF.

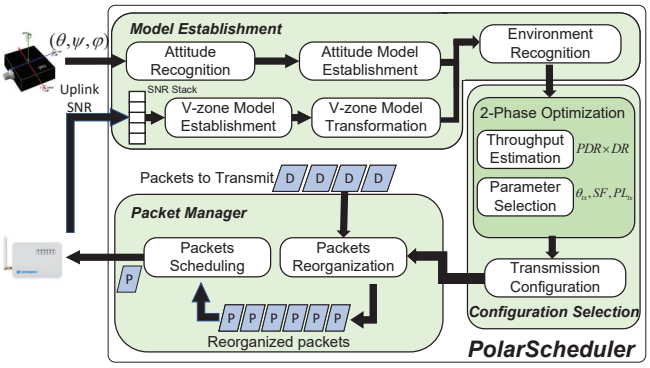


Fig. 6. Overview of *PolarScheduler*.

For expression simplicity, we use the complementary angle of φ to describe the polarization misalignment, which is denoted as $\theta = \pi/2 - \varphi$. Then Eq. (1) can be rewritten as:

$$K = cos^2(\pi/2 - \theta) = sin^2\theta \tag{2}$$

Then the receiving signal strength with polarization angle θ can be described by the following Friis Transmission Formula:

$$P_r(\theta) = P_t \frac{G_t G_r \lambda^2}{(4\pi d)^2} K \tag{3}$$

where λ is the signal wavelength, d is the distance between the transceiver, P_t and P_r are the transmission power and received power, G_t and G_r respectively indicate the gains of the transceiver antennas. Then we can measure the link quality in term of Signal Noise Ratio (SNR) $SNR(\theta)$ as:

$$SNR(\theta) = 10lg(P_r(\theta)) = SNR^* + 10lgsin^2\theta \tag{4}$$

where SNR^* is the SNR at the aligned attitude and $10lgsin^2\theta$ is the SNR loss caused by polarization misalignment.

B. V-zone Model

The above polarization model only describes the link quality at different attitudes but doesn't capture the aligned period length. Hence, we further improve the polarization model to a configuration-aware link quality model. We propose the V-zone model that uses the best-aligned SNR and the aligned boundary angle to describe the feasible region for reliable packet delivery with a specific transmission configuration.

Fig. 5 presents an illustration of the V-zone model. We can find that there are misaligned periods that a node can sway into but cannot meet the minimum requirement on decoding SNR. Only in the feasible region, which is called as V-zone, transmissions can be reliable. The example also reveals that the scope of V-zone for SF=6 is smaller than the scope of V-zone for SF=12 because a smaller SF requires higher decoding SNR and consequently has a more strict requirement on the polarization alignment.

To obtain V-zone, we analyze the minimum SNR required for decoding and then estimate the corresponding attitude. Given an Additive White Gaussian Noise (AWGN) channel, the study [20] has shown the symbol error probability P_b is:

$$P_b(SF) = 0.5 \cdot Q(\sqrt{10^{\frac{SNR}{10}} \cdot 2^{SF+1}} - \sqrt{1.386 \cdot SF + 1.154}) \tag{5}$$

where $Q(\cdot)$ is the standard normal distribution tail function. By applying the polarized $SNR(\theta)$ in Eq. (4) to Eq. (5), we can learn the symbol error probability, P_s, is:

$$P_s(SF, \theta) = 0.5 \cdot Q(\sqrt{10^{\frac{SNR^*}{10} + 2lg(sin\theta)} \cdot 2^{SF+1}} - \sqrt{1.386 \cdot SF + 1.154}) \tag{6}$$

To guarantee the success of a transmission, it is commonly agreed that SER (i.e. symbol error probability) should be lower than 10^{-6} [29]. Then we can obtain the required decoding SNR SNR^{SF}_{min} based on Eq. (6). Given SNR^{SF}_{min}, we then can calculate the aligned boundary angle based on Eq. (4), which is:

$$\theta^{SF}_{aligned} = arcsin\sqrt{10^{(SNR^{SF}_{min} - SNR^*)/10}} \tag{7}$$

Then the V-zone for SF is $[\theta^{SF}_{aligned}, \pi - \theta^{SF}_{aligned}]$.

Though V-zone gives the feasible region for reliable transmission, obtaining the optimal link throughput is non-trivial because there is a large number of configurations to search. Leveraging the V-zone model to achieve better reliability and throughput still needs elaborate designs.

IV. DESIGN

In this section, we first present an overview of *PolarScheduler* and then introduce its major components.

A. Overview of PolarScheduler

The overview of *PolarScheduler* is shown in Fig. 6. Similar to PolarTracker, we use the on-board Inertial Measurement Unit (IMU) to learn the attitude motion of a floating node. Then model establishment component establishes the attitude model utilizing the angular velocity and acceleration. The swaying boundary angle θ_{bound} and swaying duration T_{sway} will be recorded and updated. Along with the attitude recognition, the node sends a few packets following default MAC protocol and records the corresponding attitudes. Then the SNR feedback from the base station will be used to establish the V-zone model of current transmission configuration. Upon obtaining one V-zone model, *PolarScheduler* uses the V-zone model transformation module to estimate V-zone models of other configurations. Then the configuration selection component will estimate the throughput and select the largest

one among all the configurations. A two-phase optimal configuration searching method is used to efficiently identify the best configuration. After deciding the configuration, the packet manager component will reorganize the packets to schedule transmissions into the aligned period. Along with the transmissions, the model establishment component will keep tracking attitude changes and record the SNR to adapt to the swaying behaviour changes.

B. Model Establishment

Attitude motion model: To capture the swaying behaviors and establish the attitude motion model, *PolarScheduler* uses the on-board IMU to capture the motion of the transmitter antenna. The attitude recognition module leverages the Euler angles provided by IMU to calculate $\hat{\theta}$, which is the angle between the antenna and the x-y plane.

$$\hat{\theta} = \frac{\pi}{2} - arccos(cos(\Psi)cos(\gamma)), \tag{8}$$

where Ψ and γ are the roll and pitch angle from the Euler angles provided by IMU.

Apart from the instant attitude of a node, we also concern the swaying pattern to capture the aligned period length. By obtaining the angular velocity and acceleration, the swaying estimation module is able to estimate the swaying boundary and swaying duration of the node. The sampling rate of IMU is 200Hz. We can obtain an attitude angle measurement every $5ms$, which is able to describe the attitude motion model.

V-zone model establishment: During the initial phase, a node transmits several short packets and record the corresponding attitude angle θ_t. Then the node can learn the received SNR SNR_{re} from the feedback of the base station. To retrieve the polarization angle, not only the attitude of the transmitting antenna is needed, but the relative location of receiving antenna as well. With the projection angle α between the antenna and the x-z plane as depicted in Fig. 4, the polarization angle θ can be obtained as:

$$\theta = arcsin(\frac{sin(\hat{\theta})}{cos(\alpha)}). \tag{9}$$

Based on Eq. (4) and Eq. (9), we can obtain the relationship among the best-aligned SNR $\hat{SNR^*}$, the projection angle α and the received SNR SNR_{re} by the following equation:

$$SNR_{re} = \hat{SNR^*} + 10lg(\frac{sin\theta_t}{cos\alpha})^2 \tag{10}$$

In the design, we apply the average of 5 estimated $\hat{SNR^*}$ and α as smooth treatment. Then comparing with the minimum SNR required for decoding SNR_{min}^{SF}, we can learn the aligned boundary angle in V-zone model for the current configuration.

V-zone model transformation: We propose a V-zone model transformation method that can infer the V-zone models of other configurations based on the probed V-zone model of current configuration.

Among the parameters that we want to control in a configuration, it is SF that influences the V-zone scope decided by the different minimum required decoding SNRs. Since the underlaying attitude motion model and the physical link quality at different attitudes are actually the same for different models, we can use Eq. (5) to calculate the difference of the minimum required decoding SNR between two configurations. Then based on Eq. (7), given the aligned angle $\theta_{aligned}^{SF=i}$ and the corresponding $SNR_{min}^{SF=i}$ of current V-zone model, we can calculate the aligned angle of configuration $SF = j$ as:

Preamble		Header		Payload	
		if Implicit Header Disabled			
Pre-preamble	Mandatory Preamble	PHY Header	PHY Header CRC	Payload	CRC
6-65535 Symbols	4.25 Symbols	0-3 bytes (encoded in CR 4/8)		0-255 bytes	0-2 bytes

Fig. 7. The structure of a LoRa packet.

$$\theta_{aligned}^{SF=j} = arcsin\frac{sin\theta_{aligned}^{SF=i}}{\sqrt{10^{(SNR_{min}^{SF=i} - SNR_{min}^{SF=j})/10}}} \tag{11}$$

Based on Eq. (11), we can then obtain all the V-zone models of all the configurations without probing.

C. Configuration Selection

To improve both reliability and throughput, *PolarScheduler* actively controls the transmission configuration, including transmission timing, SF, and packet length. We don't control parameters such as TP and BW due to practical concerns. Adjusting BW for a single node will cause unpredictable interference for the whole network. TP is also seldom changed in a floating LoRa network to keep the planned coverage. We first model the reliability and throughput and then introduce our configuration selection method.

Reliability estimation: We use *PDR* to measure the reliability. Given the estimated SER in Eq. (6), we can estimate the error probability of different parts of a packet, including preamble, header and payload, based on the structure of a LoRa packet, which is shown in Fig. 7.

A LoRa preamble usually includes 8-symbol pre-preamble, 4.25-symbol mandatory preamble including 2-symbol Sync word and 2.25-symbol SFD. Since preamble is detected by correlation, the process can be treated as receiving a symbol modulated with an SF of $(SF + log_2(8+2+2.25))$, i.e., $SF + log_2(12.25)$. Then the preamble detection probability $P_{preamble}$ is:

$$P_{preamble} = 1 - P_s(SF + log_2(12.25), \theta) \tag{12}$$

As for the header part, the LoRa header utilizes a fixed coding rate of 4/8 that can correct 1 bit error for each four bits [30]. Given the header length L_h, the header decoding probability P_{header} can be estimated as:

$$P_{header} = \left[(1-P_s)^4 + 3(1-P_s)^7 P_s\right]^{\lceil L_h/4SF \rceil} \tag{13}$$

The coding rate of payload is specified by the user-defined parameter CR, with the values of 1,2,3, and 4 that refer to the redundancy of 4/5, 4/6, 4/7, and 4/8, respectively. Among

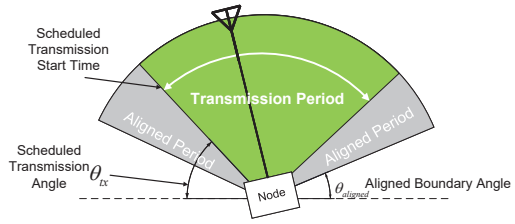
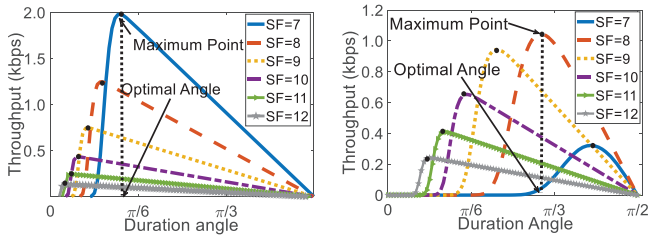


Fig. 8. Short packets cannot make full use of the aligend period.



(a) SNR = 0dB (b) SNR = -10dB

Fig. 9. Theoretical throughput with different transmission angle under different SNR.

these settings, CR of 4/5 and 4/6 can only detect errors and CR of 4/7, 4/8 can correct 1 bit error for every 4 data bits. The payload decoding probability with the payload length L_p can be estimated as:

$$P_{payload} = \begin{cases} (1-P_s)^{\lceil L_p/SF \rceil} & \text{CR=1,2} \\ ((1-P_s)^4 + 3(1-P_s)^{(CR+3)} P_s)^{\lceil L_p/4SF \rceil} & \text{CR=3,4} \end{cases} \quad (14)$$

Then the PDR of a LoRa packet can be calculated as follows.

$$PDR(SF,\theta) = P_{preamble} \times P_{header} \times P_{payload}, \quad (15)$$

Throughput estimation: We use $PDR \times DR$ to measure the throughput. Though using a small part of the aligned period can achieve a high PDR, giving up the rest aligned period leads to a low throughput. For example, as shown in Fig. 8, $\theta_{aligned}$ is the aligned boundary angle and the node starts the transmission from the attitude with the angle of θ_{tx}. Based on the data rate estimation given by Semtech [30], the DR when transmitting only within θ_{tx} can be estimated as:

$$DR(SF,\theta_{tx}) = \frac{BW \times SF}{2^{SF}} \times \frac{\pi/2 - \theta_{tx}}{\pi/2 - \theta_{aligned}} \quad (16)$$

To show the improve room of using optimal configurations, we plot the theoretical throughput with different transmission angle under different SNR in Fig. 9. We can clearly find that a maximum throughput exists in both SNR settings. Initially, the throughput increases with the increase of θ_{tx}. This is because θ_{tx} is in the misaligned region and the PDR is low. Increasing θ_{tx} across the aligned boundary angle will lead to a sharp increase of throughput due to the big PDR improvement. But when keeping increasing θ_{tx}, the utilized aligned period is shorten and the throughput decreases. When θ_{tx} continues increasing to the right-side misaligned region, throughput will significantly decrease due to the low PDR. From the results, we can also find that for different link quality, the optimal configuration in terms of SF is different. Hence, carefully identifying and selecting the optimal configuration is crucial.

Optimal configuration selection: The optimal configuration will maximize $PDR(SF,\theta) \times DR(SF,\theta)$ within $\theta_{aligned}$. However, searching all the configurations is inefficient. To efficiently get the optimal configuration, we propose a two-phase optimal configuration selection method. We only decide the optimal SF in the first phase and then determine the packet length in the second phase to reduce the searching space.

In the first phase, we design a ternary search based method to further reduce the searching overhead. Ternary search is an algorithm that can solve the problem of searching the maximum or minimum of functions with a single peak. Specifically, we first divides the whole feasible angles within V-zone into three sections evenly. Given the two junction points between the three sections, we will compare the metric of two attitude angles at junction points. The algorithm will narrow down the feasible angles by giving up the section on the side with a smaller metric. The procedure will be repeated until the range of the feasible angles is smaller than the value we demand. Then the maximum point is found.

In practical implementation, due to the limited computation resource, the ternary search can consume up to $0.5s$ on a LoRa node. If the ternary search is adopted every time the best-aligned SNR changes, it will be too costly. Since the best-aligned SNR variation is not frequent, we can adopt periodical configuration selection or trigger the configuration selection when the environment significantly changes. In our current implement, we periodically estimate the optimal configuration every $30s$ for simplicity. To avoid ternary searching blocking packet transmissions, we schedule the ternary search into the misaligned periods when the computation resources are free.

In the second phase of optimal configuration selection, we decide the suitable packet length to fit the aligned period $T_{aligned}$ best. In LoRa, each chirp consists of 2^{SF} chips carrying SF data bits. With a bandwidth of BW, the duration of a single chirp T_{sym} can be estimated as $T_{sym} = \frac{2^{SF}}{BW}$. A LoRa packet mainly consists of a preamble with the length of 12.25 symbols and the payload with a varying number of symbols. Then the packet duration is $T_{pkt} = (12.25 + S_{pl}) \cdot T_{sym}$, where S_{pl} is the number of symbols in the payload, which is:

$$S_{pl} = 8 + max\left(4CR\left\lceil \frac{8PL - 4SF + 28 + 16CRC}{4SF} \right\rceil, 0\right) \quad (17)$$

where PL is the packet length, CRC is cyclic redundancy check which uses a symbol mandatorily for uplink packets.

Based on Eq. (17), the maximum payload size S^*_{pl} that can be transmitted within the duration of aligned period $T_{aligned}$ are estimate by the following equation:

$$S^*_{pl} = \left\lfloor \frac{1}{2} \cdot [(\left\lfloor \frac{T_{aligned}}{T_{sym}} - 0.25 \right\rfloor - 21) \cdot \frac{SF}{4CR} + SF - 11] \right\rfloor \quad (18)$$

By analyzing the packet length, we can decide the transmission start time. Hence, we have obtained the transmission configuration including transmission start time T_{start}, the best SF SF^* and the best packet length decided by S^*_{pl}.

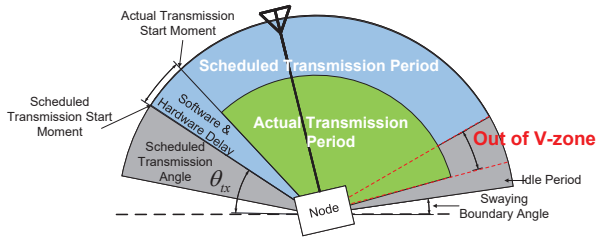
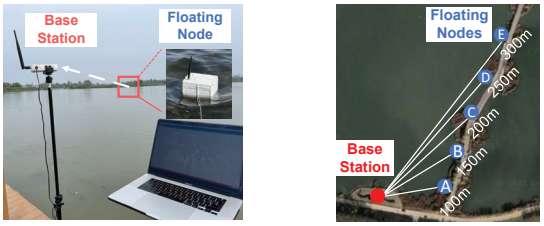
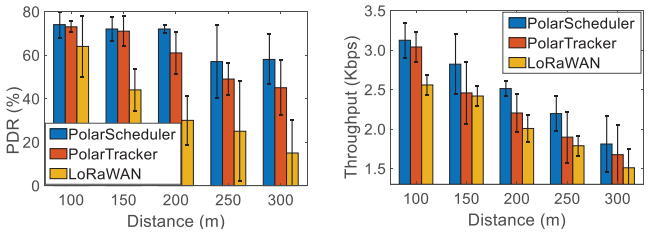


Fig. 10. Mismatch of the scheduled transmission period and the actual transmission period caused by hardware and software delay.



(a) Scene (b) Deployment

Fig. 11. Experiment setting.



(a) PDR (b) Throughput

Fig. 12. Overall performance comparison.

D. Packet Manager

After obtaining the optimal transmission configuration, the packet manager component schedules the transmissions accordingly. We first encode the data with the parameter of SF^* and then reorganize the packets to fill the aligned period fully. If an encoded data block has a length larger than S^*_{pl}, we will partition it into multiple packets. If the encoded data block has a much smaller length than S^*_{pl}, we will combine it with other data blocks to package into a long packet.

After packaging packets, the packet manager will start loading the packet ahead of the scheduled transmission start time due to the software and hardware delay. Based on our measurement, the time difference between the time initiating transmission and the time that the packet is transmitted on the air is about $0.2s$. As shown in Fig. 10, if ignoring the hardware and software delay, the scheduled transmission period will mismatch the actual transmission period. The latter part of the packet may still falls on the period which is supposed to be idle, causing a reliability drop. Hence, to avoid the ill effect of the delay, we load the packet into the buffer ahead of the scheduled transmission start time to align with the desired transmission period. As long as the attitude recognition module recognizes the transmission period, the packet can be transmitted immediately to fit the aligned period.

V. Evaluation

A. Experiment Setup

We implement *PolarScheduler* on commodity LoRa platform. Specifically, we use Dragino LG01-P with HopeRF's RFM96W [31] as the gateway and Dragino Shield with Semtech SX1278 [32] as the floating node. The antenna gain of the transceiver is 3dBi. To acquire the physical attitude information, we fixed the IMU and the node together in a waterproof box. We conduct experiments in the real-world environment. The experiment scene is shown in Fig. 11(a). In our experiments, we deploy a five-node floating LoRa network on the water with distances of $100m$, $150m$, $200m$, $250m$, and $300m$ from the gateway, as shown in Fig. 11(b).

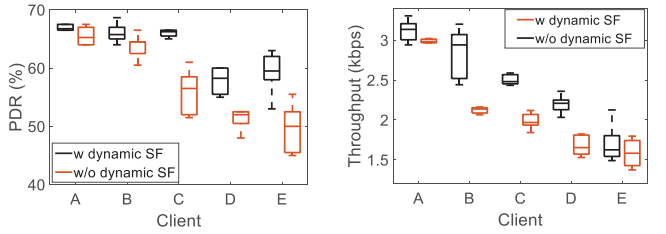
During the experiments, LoRa nodes operate in the frequency band of 433MHz. Unless otherwise specified, the default parameters adopted in our evaluation are: BW = 250kHz, CR = 4/5, and TP = 5dBm. To compare with existing methods, we also implement LoRaWAN [3] and PolarTracker [18] on our platforms. For LoRaWAN and PolarTracker, the packet length is set to 200bytes, and SF is 7.

B. Overall Performance

We first present the overall performance comparison with PolarTracker and LoRaWAN in our deployed floating network. Each node transmits for 30 minutes with a packet transmission interval of $500ms$. We alternatively execute different methods to keep the floating environment similar.

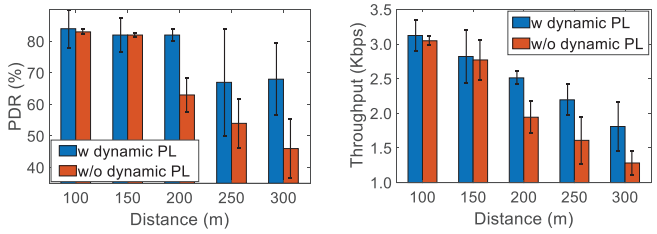
The experimental results are shown in Fig. 12. From Fig. 12(a), we can clearly find *PolarScheduler* achieves the best performance at all the distances. At distance $100m$, $150m$, $200m$, $250m$, and $300m$, the mean PDR of *PolarScheduler* is 94.2%, 92.7%, 92.1%, 77.6%, and 78.3%, respectively. The default LoRaWAN has the worst performance due to the ignorance of polarization misalignment. It achieves the mean PDR of 84.3%, 64.7%, 50.0%, 45.4%, and 35.3% at distance from $100m$ to $300m$. Compared with LoRaWAN, *PolarScheduler* increases the PDR by 20.5%, 43.8%, 84.0%, 71.2%, 122.1% at the five distance respectively. By controlling the transmissions start at the aligned attitude, PolarTracker can obviously improve the reliability and achieve the PDR of 93.3%, 91.6%, 81.5%, 69.2%, and 65.5% at the five distances from $100m$ to $300m$. But with the distance increases, the improvement of PolarTracker decreases. This is because PolarTracker only passively adopt to the polarization misalignment but cannot adjust transmissions to suit for the lower and more dynamic link qualities at long distance. By actively adjusting the transmission configurations, *PolarScheduler* can further improve PolarTracker's PDR by 1.1%, 1.1%, 13.5%, 11.6%, 20.0% respectively for the five distances.

Fig. 12(b) presents the corresponding throughput during experiments. At distance $100m$, $150m$, $200m$, $250m$, and $300m$,



(a) PDR (b) Throughput

Fig. 13. Performance of dynamic parameter selection.



(a) PDR (b) Throughput

Fig. 14. Performance of dynamic packet length control under different distances.

the throughput of *PolarScheduler* is 3.124kbps, 2.821kbps, 2.512kbps, 2.195kbps, 1.810kbps respectively. At five distances, the throughput of *PolarScheduler* is 22.1%, 16.7%, 25.2%, 22.8%, 20.0% higher than the throughput of LoRaWAN, and 2.8%, 14.8%, 14.0%, 15.7%, 8.1% higher than the throughput of PolarTracker. By comparing the PDR improvement, we can find that at the short distance such as 100*m*, compared to our method, PolarTracker achieves a similar PDR but has much lower throughput. Such results reveal that PolarTracker can control the packet in the aligned period but cannot fully fill the whole period. Thanks to the V-zone model, *PolarScheduler* can capture the aligned period length and adjust the transmissions to make full use of the aligned period.

C. Performance of PolarScheduler's Modules

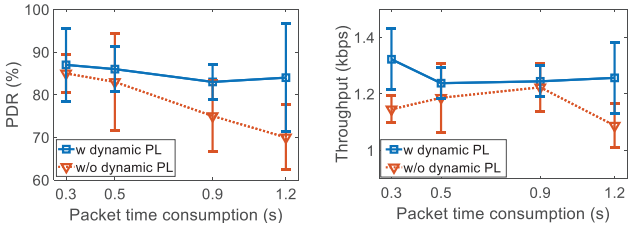
We then evaluate the key designs of *PolarScheduler*, including transmission parameter selection, packet length control, transmission scheduler, and V-zone transformation.

1) Parameter Selection: We first investigate the performance of *PolarScheduler* with and without dynamic SF selection. The experiments are also conducted on the deployed network. *PolarScheduler* without dynamic SF use a fixed SF=7. Each node transmits packets with an interval of 500*ms*.

We plot the results in Fig. 13. We can find that for all five nodes, using dynamic SF can significantly improve the performance. Node A has a good PDR for both methods because its link quality is less influenced by polarization due to its short distance. But when the distance is longer than 150*m*, polarization will significantly influence the reliability. For Node E, the method without dynamic SF can only achieve a PDR of 59.2%, while *PolarScheduler* with dynamic SF can improve the PDR to 78.3%. Due to the improvement of reliability, the throughput improvement of *PolarScheduler* with dynamic SF is 4.3%, 33%, 26.9%, 33.8%, 17.3% for node A, B, C, D, and E respectively.

2) Packet Length Control: We then investigate the performance improvement brought by packet length control. All five nodes transmit packets with an interval of 500*ms* with the payload length of 250bytes. SF is set to 8.

The results are shown in Fig. 14. Compared to the performance without packet length control, *PolarScheduler* with packet length control can improve the PDR by 26.0%, 20.3%, 39.4% respectively at 200*m*, 250*m*, 300*m*. The corresponding



(a) PDR (b) Throughput

Fig. 15. Performance of packet length control under different packet time consumption.

throughput can be improved by 29.2%, 36.3%, 41.4%. The results show that our packet length control method can effectively adjust the packet length to fit the aligned period length. It's worth noting that we set SF to 8 in this experiment and therefore the PDR of node B increases, compare to the results in Fig. 14(a) that use the SF of 7.

To further verify the efficiency of packet length control, we vary the length of data packets to obtain different packet duration. Then we measure the PDR and throughput of *PolarScheduler* with and without dynamic packet length control. The results are presented in Fig. 15. In the experiments, we set the SF to 9 and change the packets payload length, which leads to packet time consumptions of 0.3*s*, 0.5*s*, 0.9*s* and 1.2*s* respectively. As shown by the results, with the packet length control the PDR can reach 87.2%, 86.4%, 83.7%, and 84.1% respectively, which shows an improved reliability in all conditions. For the one without the packet management, it shows a similar PDR when packet duration is 0.3*s* and 0.5*s*. However, when the packet duration increases, the PDR falls to 75.0% and 70.4% respectively to 0.9*s* and 1.2*s*. The main reason behind is that the transmission time exceeds the aligned duration, which leads to packet failures.

As for the transmission efficiency, *PolarScheduler* with packet length control can reach a throughput higher than 1.2kbps in all conditions. Without packet length control, the throughput is only 1.09kbps when the packet duration is 1.2*s*, causing by the poor reliability with large packets. When the time consumption is 0.3*s*, as it can not fill the full aligned period and make use of the aligned period, the throughput is 1.15kbps, which is still lower than the method with packet length control. To summarize, the packet management module

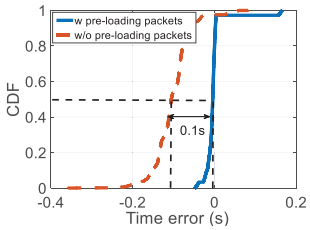


Fig. 16. The accuracy of pre-loading based transmission start time control.

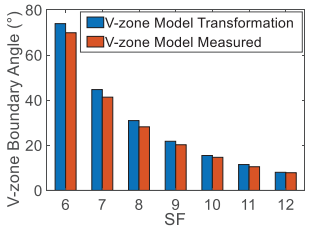


Fig. 17. The accuracy of V-zone model transformation.

can improve both reliability and throughput.

3) Transmission Scheduler: The transmission scheduler controls the transmission start time. If we start calling the transmission command at the scheduled attitude, the transmission actually will miss the schedule because of the hardware and software delay caused by packet loading. Hence, *PolarScheduler* loads packets ahead of the scheduled attitude based on the attitude motion speed. We measure the mismatch error by calculating the difference between the actual transmission start time and the time arriving the scheduled starting attitude. The results are shown in Fig. 16. We can find that *PolarScheduler* can achieve a quite small mismatch error, the median of which is $0.009s$. For comparison, we also plot the mismatch error when *PolarScheduler* doesn't use packet scheduler to load packet in advance. From Fig. 16, we can find that the median mismatch error can be as large as $0.1s$, which can be used for transmitting 52bytes data under SF=7. Hence, packet scheduler is very helpful to *PolarScheduler* to obtain the expected improvement on PDR and throughput.

4) Accuracy of V-zone Transformation: The V-zone transformation method is the key to avoid the probing cost when establishing V-zone models under different configurations. Though effective to reduce overhead, its accuracy should be also satisfying for the following transmission control decision. We investigate the accuracy of V-zone model by comparing the boundary angles obtained by our V-zone transformation method and the ground truth that is obtained by probing the minimum required SNR. Fig. 17 shows the results. We can find that the estimation error of our transformation method is within 3° and decreases with the increase of SF. When SF is 12, the error is only 0.8°. The results demonstrate the accuracy of our V-zone model transformation method. Besides, it is worth noting that all the results obtained by V-zone model transformation are slightly larger than that of the real-world measurement. A higher boundary angle means a smaller inner angle of the V-zone and a shorter aligned period. This is our conservative design that guarantees a reliable aligned period used for transmission.

VI. Related Work

Though LoRa is a promising technology with the ability to connect thousands of nodes, its performance can be poor due to the diversity of communication environments. Adaptive control is a conventional method for LoRaWAN to improve transmission reliability. For example, ADR [19], [33] employed in LoRaWAN is an effective mechanism to optimize data rates, air-in-time and energy consumption. Based on ADR, EXPLoRa-SF [34] attempts to allocate SF so as to equalize the time on air of the packets and deal with different traffic conditions. EF-LoRa [21] and AdapLoRa [22] aims to optimize LoRa network life-time with dynamic resource adaptation. DyLoRa [20] builds an energy efficiency model under different transmission parameters and control the transmission accordingly. Aiming at the reliability reduction caused by dynamic environment, a few MAC protocols are proposed to improve the reliability [35]–[38].

However, these methods fail to protect the transmission reliability of floating LoRa networks because they focus on the occasional link attenuation changes but ignore the fast-changing link quality caused by polarization loss. The polarization mismatch will reduce the reliability, which has been proved by previous studies [39]–[41]. PolarTracker [18] is the first work focusing the link degradation problem caused by polarization in floating LoRa networks. PolarTracker proposes an attitude-aware channel access method to control the transmission start from the aligned attitude. But it fails to improve the performance when transmitting long packets with long duration due to the ignorance of aligned period length. Different from PolarTracker, our method not only controls the transmission start time but also actively control the transmission configurations to make full use of the aligned period. By filling the aligned period by an appropriate packet, our method can improve the reliability as well as throughput.

VII. Conclusion

In this paper, we propose *PolarScheduler*, a novel dynamic transmission control method for floating LoRa networks to improve both reliability and throughput. Besides the transmission start time, *PolarScheduler* also actively controls the transmission configuration of SF and packet length to best fit the polarization aligned period. We propose a V-zone model to capture the polarization aligned periods under different transmission configurations. Based on our model, *PolarScheduler* adopts a model transformation method to efficiently obtain all models based on the probed results of current configuration. We also design an efficient optimal configuration selection method to make full use of aligned periods. Our evaluation on commercial LoRa platforms demonstrates that *PolarScheduler* can improve the reliability and throughput by up to 20.0% and 15.7%, compared to the state-of-the-art approach.

Acknowledgment

The work is supported by the Funds for Creative Research Groups of China under Grant 61921003, the National Natural Science Foundation of China under Grant 62072050 and Grant 61932013, the A3 Foresight Program of NSFC under Grant 62061146002 and the Funds for International Cooperation and Exchange of NSFC under Grant 61720106007. Xiaolong Zheng is the corresponding author.

REFERENCES

[1] S. Fang, L. Da Xu, Y. Zhu, J. Ahati, H. Pei, J. Yan, and Z. Liu, "An integrated system for regional environmental monitoring and management based on internet of things," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1596–1605, 2014.

[2] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart community: an internet of things application," *IEEE Communications magazine*, vol. 49, no. 11, pp. 68–75, 2011.

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

[4] M. Chen, Y. Miao, Y. Hao, and K. Hwang, "Narrow band internet of things," *IEEE Access*, vol. 5, pp. 20 557–20 577, 2017.

[5] L. Parri, S. Parrino, G. Peruzzi, and A. Pozzebon, "A lorawan network infrastructure for the remote monitoring of offshore sea farms," in *Proceedings of IEEE I2MTC*, 2020.

[6] P. S. Dehda, S. Jayram, A. M. Abu-Mahfouz, and K. Ouahada, "A sea rescue operation system based on lora," in *Proceedings of IEEE icABCD*, 2019.

[7] W. Du, Z. Xing, M. Li, B. He, L. H. C. Chua, and H. Miao, "Optimal sensor placement and measurement of wind for water quality studies in urban reservoirs," in *Proceedings of ACM/IEEE IPSN*, 2014.

[8] J. Cecílio, P. M. Ferreira, and A. Casimiro, "Evaluation of lora technology in flooding prevention scenarios," *Sensors*, vol. 20, no. 14, p. 4034, 2020.

[9] C. Li, H. Guo, S. Tong, X. Zeng, Z. Cao, M. Zhang, Q. Yan, L. Xiao, J. Wang, and Y. Liu, "Nelora: Towards ultra-low snr lora communication with neural-enhanced demodulation," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 56–68.

[10] S. Tong, Z. Shen, Y. Liu, and J. Wang, "Combating link dynamics for reliable lora connection in urban settings," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 642–655.

[11] L. Liu, Y. Yao, Z. Cao, and M. Zhang, "Deeplora: Learning accurate path loss model for long distance links in lpwan," in *IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[12] X. Xia, Y. Zheng, and T. Gu, "Litenap: Downclocking lora reception," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2632–2645, 2021.

[13] X. Xia, N. Hou, Y. Zheng, and T. Gu, "Pcube: scaling lora concurrent transmissions with reception diversities," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 670–683.

[14] J. Jiang, Z. Xu, F. Dang, and J. Wang, "Long-range ambient lora backscatter with parallel decoding," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 684–696.

[15] Z. Xu, P. Xie, and J. Wang, "Pyramid: Real-time lora collision decoding with peak tracking," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 2021, pp. 1–9.

[16] S. Tong, J. Wang, and Y. Liu, "Combating packet collisions using non-stationary signal scaling in lpwans," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 234–246.

[17] X. Xia, Y. Zheng, and T. Gu, "Ftrack: Parallel decoding for lora transmissions," *IEEE/ACM Trans. Netw.*, vol. 28, no. 6, pp. 2573–2586, 2020.

[18] Y. Wang, X. Zheng, L. Liu, and H. Ma, "Polartracker: Attitude-aware channel access for floating low power wide area networks," in *Proceedings of IEEE INFOCOM*, 2021.

[19] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in lorawan: Recent solutions and major challenges," *Sensors*, vol. 20, no. 18, p. 5044, 2020.

[20] Y. Li, J. Yang, and J. Wang, "Dylora: Towards energy efficient dynamic lora transmission control," in *Proceedings of IEEE INFOCOM 2020*, 2020.

[21] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, "Towards energy-fairness in lora networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 788–798.

[22] W. Gao, Z. Zhao, and G. Min, "Adaplora: Resource adaptation for maximizing network lifetime in lora networks," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. IEEE, 2020, pp. 1–11.

[23] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do lora low-power wide-area networks scale?" in *Proceedings of ACM MSWIM*, 2016.

[24] G. Premsankar, B. Ghaddar, M. Slabicki, and M. Di Francesco, "Optimal configuration of lora networks in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7243–7254, 2020.

[25] C. LI and Z. CAO, "Lora networking techniques for large-scale and long-term iot: A down-to-top survey," *ACM Computing Surveys (CSUR)*, 2022.

[26] W. L. Stutzman, *Polarization in electromagnetic systems*. Artech house, 2018.

[27] C. A. Balanis, *Antenna theory: analysis and design*. John wiley & sons, 2016.

[28] L. Shangguan and K. Jamieson, "Leveraging electromagnetic polarization in a two-antenna whiteboard in the air," in *Proceedings of ACM CoNEXT*, 2016.

[29] O. Afisiadis, M. Cotting, A. Burg, and A. Balatsoukas-Stimming, "On the error rate of the lora modulation with interference," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1292–1304, 2020.

[30] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on lora networking: Research problems, current solutions, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 371–388, 2019.

[31] "Dragino." [Online]. Available: https://www.dragino.com

[32] SX1278, "Datasheet sx1276/77/78/79," 2020. [Online]. Available: https://semtech.my.salesforce.com

[33] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," *ACM TOSN*, vol. 15, no. 2, pp. 1–35, 2019.

[34] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "Explora: Extending the performance of lora by suitable spreading factor allocations," in *Proceedings of IEEE WiMob*, 2017.

[35] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "Lmac: Efficient carrier-sense multiple access for lora," in *Proceedings of ACM MobiCom*, 2020.

[36] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of lorawans through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, 2018.

[37] D. Zorbas, P. Maillé, B. O'Flynn, and C. Douligeris, "Fast and reliable lora-based data transmissions," in *Proceedings of IEEE ISCC*, 2019.

[38] C. Pham, "Investigating and experimenting csma channel access mechanisms for lora iot networks," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[39] H. Tolhoek, "Electron polarization, theory and experiment," *Reviews of modern physics*, vol. 28, no. 3, p. 277, 1956.

[40] T. S. See, T. M. Chiam, M. C. Ho, and M. R. Yuce, "Experimental study on the dependence of antenna type and polarization on the link reliability in on-body uwb systems," *IEEE transactions on antennas and propagation*, vol. 60, no. 11, pp. 5373–5380, 2012.

[41] M. Badi, J. Wensowitch, D. Rajan, and J. Camp, "Experimental evaluation of antenna polarization and elevation effects on drone communications," in *Proceedings of ACM MSWIM*, 2019.