A New-Dynamic Adaptive Data Rate Algorithm of LoRaWAN in Harsh Environment

Chao Jiang, Yue Yang, Xianghui Chen, Jianxin Liao, Weixian Song, and Xihai Zhang, *Member, IEEE*

***Abstract*—The Adaptive Data Rate (ADR) algorithm is used in LoRaWAN, allocating an appropriate transmission rate for terminal equipment to improve channel utilization and reduce power consumption. However, the standard ADR algorithm is only suitable for static terminal equipment. In addition, due to the complexity of the external environment, the selected data rate will not match the new environment and communication packets will be lost. Therefore, this paper proposed a novel and more effective ADR algorithm called New-Dynamic ADR (ND-ADR). The algorithm mainly solves two problems, i.e., the standard ADR algorithm cannot be applied to mobile terminal devices and the poor communication quality and high packet loss rate in harsh environments. In this study, we also developed FLoRaWAN (Frame for LoRaWAN), a simulation framework for the star network topology LoRaWAN in OPNET. Furthermore, we built an OKUMURA-HATA model and additionally introduced a noise factor β to simulate the loss of wireless communication in harsh environments. Finally, extensive simulation results showed that the number of data packets required by different end nodes for rate allocation was different. Compared with the standard ADR algorithm, even for mobile nodes in harsh environments, the ND-ADR algorithm reduced network energy consumption by about 13%, reduced network delay by about 18%, and increased effective throughput by about 15%. Therefore, the improved ND-ADR algorithm is more suitable for wireless communication of removable nodes in harsh environments. Its advantages are better awareness of link environment, faster data rate regulation, improved channel utilization, and further reduction of network energy consumption.**

***Index Terms*—LoRaWAN, ADR, ND-ADR, FLoRaWAN, OKUMURA-HATA.**

I. Introduction

Due to the lack of reliable networks and power infrastructure, it is usually not feasible to use wired networks in harsh environments. The most widely used short-range wireless communications (e.g., Bluetooth and WiFi), cellular networks (e.g., 2G, 3G, 4G, and the recent 5G), and other wireless communication technologies are also not applicable. Meanwhile, Low-Power Wide Area Network (LPWAN) technology is increasingly used in Wireless Sensor Networks (WSN) for data transmission to solve these problems [1][2]. The most significant advantages of LoRa technology are low power consumption, easy networking, low cost, and long transmission distance, which can meet long-term operation and battery power supply can last for several years. Therefore, it is considered to be the most promising representative technology of LPWAN. LoRaWAN [3] is a standard communication protocol constructed by the LoRa Alliance [4] at the MAC layer of the LoRa protocol. Adaptive Data Rate [5] (ADR) is an essential feature of the LoRaWAN, which can allocate an appropriate data rate for terminal equipment to improve channel utilization and reduce power consumption. However, the standard ADR algorithm is only suitable for static terminal equipment. Only the Signal-Noise Ratio (SNR) of the data packets is selected as the calculation basis. The judgment condition is relatively simple, only the node itself is judged, and the consideration of other factors (e.g., vegetation occlusion, temperature, humidity, etc.) is lacking in the network [6].

Methods of controlling transmission parameters (e.g., spreading factor and transmission power) in LoRaWAN have been proposed in most published papers to solve these problems [7-13]. Furthermore, the existing improved methods of the standard ADR algorithms can be classified into two categories, i.e., the link-based approaches [7, 10-13] and the network-aware approaches [8][9]. Slabicki *et al.* proposed the ADR+ algorithm [7]. They pointed out that in the standard ADR algorithm, the maximum SNR in the recent data packets was selected as the calculation basis. The standard ADR algorithm is effective under stable channel conditions, but it will be affected when the channel conditions are changed rapidly (e.g., node movement, vegetation occlusion, temperature, humidity, etc.). The average SNR of recent data packets was used in the ADR+ algorithm as the basis for subsequent calculations to more accurately evaluate the link quality, thereby improving the ADR algorithm's performance in the case of unstable channel conditions. They also ranked the nodes according to the actual locations of the gateways and nodes, and then the spreading factor was assigned to each node in turn based on the optimal distribution. The optimal distribution of the spreading factor was obtained by the genetic algorithm in [8]. Reynders *et al.* [8] used a genetic algorithm to distribute spreading factors and discrete power settings to nodes inside a smaller cell of LoRaWAN. This scheme aimed to improve the packet error rate for nodes far from the base station, thereby creating more packet error rate

This research was supported by Natural Science Foundation of Heilongjiang Province of China (No. C2018023). (Corresponding author: Weixian Song, Xihai Zhang.)

The authors are with the College of Electronic and Information, Northeast Agricultural University, Harbin 150000, China (e-mail: songweixian@163.com, xhzhang@neau.edu.cn).

fairness inside a LoRaWAN cell. Abdelfadeel *et al.* [9] applied a network-aware approach similar to [8]. However, the bandwidth of nodes was assumed to be constant in [8], while the nodes can dynamically choose their bandwidth as in [9]. These were all aimed at the idea of SF distribution in large-scale networks to improve the ADR algorithm [10][11]. Marini *et al.* [10] also used the average SNR of recent data packets as the algorithm benchmark, and used the orthogonality of different SFs to minimize the collision probability when assigning data rates. Cuomo *et al.* [11] combined SF orthogonality and radio range visibility to increase the number of simultaneous users in the system, and two improved algorithms were proposed. One method was to distribute SF among nodes. The other was to ensure a balanced distribution of channel load between nodes to ensure the balance of transmission time in the air, thereby ensuring fairness. In addition, Babaki *et al.* [12] proposed an Ordered Weighted Average (OWA) algorithm to configure transmission parameters for channel noise in dense LoRa network deployment. This algorithm increased the data rate and reduced energy consumption in the case of high channel noise. Li *et al.* [13] proposed a DyLoRa dynamic transmission control system to replace the standard ADR threshold control algorithm. The main idea of the system was to construct a transmission model that used the demodulation Symbol Error Rate (SER) to characterize the Packet Delivery Ratio (PDR) and network energy efficiency based on the demodulation mechanism of LoRa. In this model, DyLoRa used a limited number of data packets to configure the best transmission parameters, which also made DyLoRa only suitable for low data rate LoRa networks with sparse traffic.

Indeed, these measures improved the standard ADR algorithm from different angles and were of great significance to improve the communication performance of the LoRa network. However, they were all research on static nodes, and rarely considered the influence of external environmental factors on LoRa communication. Especially in a harsh environment, the communication performance of the LoRa network will not only be affected by natural conditions (e.g., rain, snow, etc.) [14], but also by large-scale fading during non-line-of-sight propagation [15][16]. Jeftenić *et al.* [14] confirmed that changes in temperature and relative humidity would affect the Received Signal Strength Indicator (RSSI) and SNR in LoRaWAN communication. Due to the hysteresis of the standard ADR algorithm adjustment, the transmission delay and packet loss rate will be increased [6]. Ojo *et al.* [15] found that the presence of vegetation and the difference in density would affect the performance of wireless communication. In fact, given this situation, we can further improve the standard ADR algorithm to suit the harsh communication environment. In other words, environmental factors will affect the RSSI and SNR in the communication links, so it is possible to improve the adjustment speed of ADR through the link's perception of the environment.

The contributions of our work are summarized as follows. In fact, the two main issues that still need to be addressed by the standard ADR algorithm are the dynamic changes in the channel caused by mobile nodes and the perturbation of the standard ADR algorithm by harsh external environments (e.g., rain, snow, etc.). Therefore, we proposed a novel New-Dynamic ADR (ND-ADR) algorithm to adapt to the movement of nodes and complex and variable environmental factors. In addition, we additionally introduced an adjustable noise factor in the proposed channel model to reflect the degree of perturbation of the external environment on the channel transmission. Second, we developed FLoRaWAN, the first LoRaWAN simulation framework created by OPNET [17]. It simulated the LoRaWAN protocol through the three-tier architecture of network model, node model, and process model and supported two-way communication between the server and removable nodes. Finally, we conducted extensive simulation experiments to compare and analyze the communication performance of removable nodes in the standard ADR algorithm, the ADR+ algorithm, and the ND-ADR algorithm. The results showed that the use of link-aware approaches could improve the adjustment capability of ADR, increase channel utilization, and further reduce the energy consumption of the network.

The rest of the paper is organized as follows. Section II presents related technologies. Adaptive network configurations are described in Section III. Results and discussion are given in Section IV. Finally, conclusions are discussed in Section V.

II. Overview Of Related Technology

This section presents a basic introduction of LoRa and LoRaWAN, analyzes the standard ADR algorithm, and finally introduces the path loss model.

A. LoRa and LoRaWAN

LoRa [4] is low-power long-distance communication technology, mainly for applications such as the Internet of Things (IoT) or Machine to Machine (M2M), and is an essential wireless technology for Low Power Wide Area Networks (LPWAN). LoRa technology includes the LoRaWAN protocol and LoRa protocol. LoRaWAN refers to the MAC layer networking protocol, while LoRa is just a physical layer protocol. The main advantage of LoRa technology is that the modulation parameters can be flexibly configured in the design stage according to the needs of data rate, distance, and power consumption. There are three main parameters of the LoRa modulation function [18]: Spread Factor (SF), Bandwidth (BW), and Code Rate (CR).

SF represents the ratio between chip rate and symbol rate. In general, SF can be a value from 7 to 12. Higher SF can improve the signal's anti-interference capability, it means higher sensitivity and long-distance, but decreases data rates and resultant higher energy expenditure [19]. BW limits the lowest frequency and highest frequency of the signal allowed to pass through the channel, which can be understood as a frequency passband. Increasing BW can increase the effective data rate to shorten the transmission time but at the cost of sacrificing some RSSI [20]. CR is the ratio of the valuable part (non-redundant) in the data stream, so it is a number from 0 to 1. FEC (Forward Error Correction) technique is used in LoRa

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3118051, IEEE Internet of Things Journal

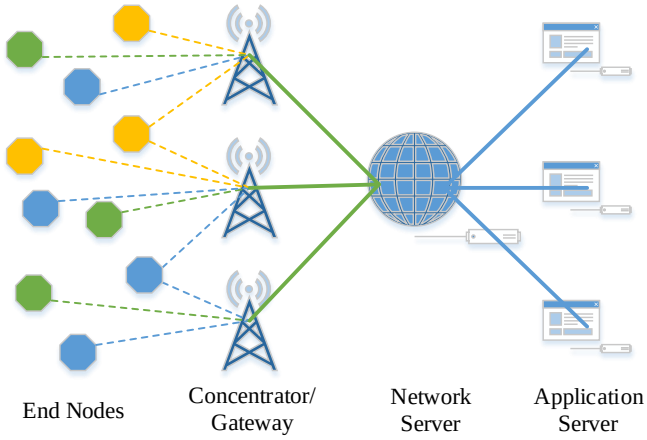


Fig. 1. LoRaWAN network architecture.

technology for forward data error detection and correction. By adding redundant data coding, FEC can detect and correct erroneous information. The fixed number of codes in LoRa coding is 4, and the range of redundant coding factor is [1,4], then the range of corresponding code rate CR is {4/5,4/6,4/7,4/8}. Higher CR provides better immunity, and the reliability of the link is higher, but it will increase the transmission time of data packets [21].

LoRaWAN [3] is a Media Access Control (MAC) layer protocol designed for LoRa long-distance communication network. It is based on LoRa modulation technology and achieves a high level of noise immunity through Chirp Spread Spectrum (CSS) modulation technology. Therefore, it can realize high-efficiency remote communication with little energy consumption in a relatively harsh environment. LoRaWAN mainly includes two parts: communication protocol and network architecture. The LoRaWAN network architecture is shown in Fig. 1.

In general, a complete LoRaWAN network includes four parts: end nodes, gateway, Network Server (NS), and application server. In the network of LoRaWAN, a star network topology is adopted between the gateway and the end nodes. Due to the long-distance characteristics of LoRaWAN, single-hop transmission can be used between them. In addition, the end nodes can send to multiple gateways at the same time. The gateway forwards the LoRaWAN protocol data between the NS and the nodes. The gateway communicates with the LoRaWAN network server through a standard IP link (e.g., Ethernet, 3G/GPRS, and WiFi).

B. ADR Algorithm

The purpose of ADR is to effectively adjust the data rate for static nodes with different transmission distances to improve the throughput of the network and maximize the battery life of the nodes [5]. Thus, all nodes can dynamically change the data rate of uplink and downlink transmission. The ADR speed regulation mechanism is divided into two parts. The NS side is responsible for increasing the transmission rate of the nodes, and the node side is accountable for reducing its transmission rate. The ADR algorithm on the node side is generally formulated by the Lora Alliance [4]. In contrast, the network operator itself can develop the algorithm on the NS side, and the ADR function mainly depends on the operation of the NS side.

Algorithm 1 Standard ADR algorithm

Initialize: $SF \in [7,12]$, $TP \in [2\text{dBm}, 17\text{dBm}]$
$SNR_{req} \leftarrow$ demodulation floor (current data rate)
$SNR_{max} \leftarrow$ max (SNR of last 20 frames)
$SNR_{margin} \leftarrow SNR_{max} - SNR_{req} - Margin\_dB$
$NStep \leftarrow \text{int}(SNR_{margin} / 3)$
while $NStep > 0$ **and** $SF > SF_{min}$ **do**
    $SF \leftarrow SF - 1$
    $NStep \leftarrow NStep - 1$
end while
while $NStep > 0$ **and** $TP > TP_{min}$ **do**
    $TP \leftarrow TP - 3$
    $NStep \leftarrow NStep - 1$
end while
while $NStep < 0$ **and** $TP < TP_{max}$ **do**
    $TP \leftarrow TP + 3$
    $NStep \leftarrow NStep + 1$
end while
Output: TP and SF

Nevertheless, Kim *et al.* [22] proposed a congestion classifier based on logistic regression, which improved adaptive data rate control on the node side of the LoRaWAN network. It is intended to solve channel congestion when judging the wireless state by receiving the acknowledgment (ACK) messages between the gateway and the nodes in LoRaWAN communication. The algorithm controls the data transmission rate according to the degree of channel congestion, but there must be an effective feedback channel, i.e., an ACK is required for each transmission. However, Pop *et al.* [23] found that the downlink traffic would affect the uplink throughput, so that this mechanism would reduce the transmission efficiency of the network. Therefore, the improvement of the ADR algorithm on the NS side is a crucial and effective way.

The implementation process of the standard ADR algorithm on the NS side is described in Algorithm 1. The NS will determine the data rate based on the n packets received recently whenever uplink data packets from end nodes are received (n = 20 in standard ADR).

Precisely, when the ADR mode is enabled, the NS controls the transmission rate and power of the terminal device through the MAC layer command LinkADR. The difference between the maximum value of SNR in the received data packets and the required SNR under the corresponding SF is calculated, and subtract margin $Margin\_dB$ (by default 5), get the SNR margin SNR_{margin} to judge whether the current rate needs to be adjusted. The minimum values of SNR and RSSI required for

demodulation at different data rates are listed in Table I [24].

If the margin SNR_{margin} is negative, it means the current data rate used by the nodes is too high, or the transmission power is low, resulting in low SNR of the uplink signal. Therefore, it is necessary to increase the transmission power or reduce the data rate.

If the margin SNR_{margin} is positive, it indicates that the current link environment is fine. We can choose to increase the data rate of the nodes or reduce the transmission power to reduce power consumption to extend the service life of the nodes. Set the step length to 3, and calculate $NStep$, which means the number of times to modify the configuration. It is increasing the data rate each time until it reaches the highest value. If there is still margin after calculation, continue to reduce the transmit power. If the value $NStep$ is 0, no adjustment is needed, and the original configuration needs to be output. The algorithm steps to be followed by the standard ADR are shown in Fig. 2.

C. OKUMURA-HATA Model

The quality of the signal received by the receiving antenna is related to its frequency characteristics and the propagation path. Usually, we use the Friis transmission equation [25] to study a radio frequency transmitting and receiving system in free space and the relationship between sending power, receiving power, antenna gain, and transmission distance. Due to the lack of obstruction, radio waves usually propagate in free space only by direct radiation. In contrast, radio waves have various propagation paths (e.g., direct, reflected, bypass, penetration, etc.) in the natural environment. Also, one of the most exciting features of LoRa modulation is its ability to send data to distances as far as 25 km at low power. According to the Friis transmission equation, it is counterintuitive because transmitting data over large distances requires a transmitter with high transmission power. Therefore, we need a more reliable path loss model to simulate better and analyze the transmission performance of LoRaWAN networks. We use the Okumura-Hata model [26] because it is one of the most popular and accurate models, especially for urban and suburban areas, and the Okumura-Hata model is also increasingly used for LoRaWAN and even for data rate adaptation [27-30]. Moreover, according to practical tests on the performance of extensive LoRa networks in specific cities, Jörke *et al.* [31] further proposed new path loss models for 868 MHz and 433 MHz LoRa smart city scenarios based on the path loss models such as Okumura-Hata.

When using the LoRaWAN protocol for communication in harsh environments, to obtain the value of the received signal strength P_r, we introduce the OKUMURA-HATA model [26] in OPNET, which uses the propagation loss in urban areas as the standard, and other regions use correction factors for correction. Modeling the path loss L_b in dB:

$$L_b = 69.55 + 26.16\lg f - 13.82\lg h_b - a(h_m) + (44.9 - 6.55\lg h_b)(\lg d)^{\gamma} + \delta \quad (1)$$

TABLE I
SNR AND RSSI REQUIRED FOR DEMODULATION AT DIFFERENT DATA RATES (BW 125KHZ) [24].

Data Rate	Spreading Factor	RSSI (dB)	SNR (dB)
DR0	SF12	-136	-20
DR1	SF11	-133	-17.5
DR2	SF10	-132	-15
DR3	SF9	-129	-12.5
DR4	SF8	-126	-10
DR5	SF7	-123	-7.5

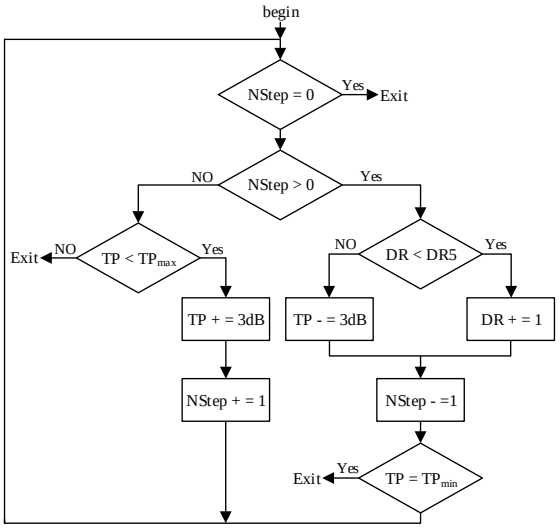


Fig. 2. Standard ADR algorithm flow chart.

where $a(h_m)$ is the correction factor for effective mobile antenna height, which is a function of the size of the coverage area. For a small to medium-sized city, the mobile antenna correction factor is given by:

$$a(h_m) = (1.1\lg f - 0.7)h_m - (1.56\lg f - 0.8) \quad (2)$$

and for a large city, it is given by:

$$a(h_m) = 8.29(\lg 1.54h_m)^2 - 1.1 \quad \text{for } f \le 300\text{MHz} \quad (3)$$

$$a(h_m) = 3.2(\lg 11.75h_m)^2 - 4.97 \quad \text{for } f \ge 300\text{MHz} \quad (4)$$

where γ is the long-distance propagation correction factor, and the expression is as follows:

$$\gamma = \begin{cases} 1 & d \le 20 \\ 1 + \left(0.14 + 1.87\times10^{-4} f + 1.07\times10^{-3} h_b\right)\left(\lg\frac{d}{20}\right)^{0.8} & d \ge 20 \end{cases} \quad (5)$$

In the network scenario designed in this paper, the distance between the gateway and the end nodes is $d \le 20$, so γ =1. where d is the distance between the gateway and the end nodes (in km); f is the frequency of the carrier (in MHz); L_b is the path propagation loss value (in dB); h_b is the effective height of the gateway (in meters); h_m is the effective height of the end nodes (in meters); δ is a zero-mean Gaussian distributed random variable with a standard deviation of σ, which is related to the actual environment,

i.e., $\delta \sim \mathrm{N}\left(0,\sigma^2\right)$.

III. Adaptive Network Configuration

In this section, we mainly introduce the ND-ADR algorithm. Meanwhile, we develope a new simulation framework FLoRaWAN (Frame for LoRaWAN), to better analyze the communication performance of the ND-ADR algorithm.

A. ND-ADR Algorithm

As mentioned earlier, when LoRaWAN is used for data transmission in a harsh environment, it will be affected by external natural factors (e.g., vegetation, rain, snow, etc.)[14][15]. Therefore, when the ADR mode is enabled, NS will consume more time for the n = 20 ($1 \le n \le 20$) data packets that need to be received. In addition, the loss of some data packets will cause the nodes and the network server to call and wait for confirmation continuously, which will take up more channel resources. However, these are not the purpose of using ADR.

Therefore, based on the standard ADR algorithm, we combine the RSSI calculated by the OKUMURA-HATA model and the average value SNR_{avg} of the recently received n data packets as the adjustment index. In addition, we do not think that n = 20 is the best calculation basis. Choosing the value of n too low (e.g., n = 1) will cause significant changes in the network, and picking too high (e.g., n = 20) will cause some problems such as transmission delay. Therefore, the value of n is particularly critical. Consequently, we will use a set of dynamic selection algorithms to determine the most suitable value of n in the virtual environment. We call these improved new ADR algorithms the New-Dynamic ADR (ND-ADR) algorithm. The specific algorithm flow is in Algorithm 2.

We set the initial empirical value of n to 3 according to [32](step 3 of Algorithm 2), which we think is an initial scientific value. In addition, the main idea of the ND-ADR algorithm is to combine RSSI and average value SNR_{avg} as the demodulation basis instead of selecting only the maximum value SNR_{max} as the adjustment basis in the standard ADR algorithm. Table I shows the minimum values of SNR and RSSI required to demodulate different SFs, i.e., the RSSI obtained through the path loss can be demodulated to be SF_{RSSI}, and the average value SNR_{avg} can be demodulated to SF_{SNR}. When the values of SF_{RSSI} and SF_{SNR} are different (Step 7 of Algorithm 2), we think that the external environment has changed, and the number of selected data packets n = 3 is too low to be used as a basis for calculation. At this time, the value of n needs to be increased, then return to step 3 and execute it again. When step 3 is completed, it is necessary to select the most recently received n+1 data packets to calculate the average value SNR_{avg} and RSSI.

Algorithm 2 **ND-ADR** algorithm

1: **Initialize**: $SF \in [7,12]$, $TP \in [2\mathrm{dBm}, 17\mathrm{dBm}]$
2: SNR_{req} ← demodulation floor (current data rate)
3: SNR_{avg} ← average (SNR of last n frames)
4: RSSI ← calculation (OKUMURA-HATA model)
5: SF_{SNR} ← demodulation floor (current SNR_{avg})
6: SF_{RSSI} ← demodulation floor (current RSSI)
7: **if** $SF_{SNR} > SF_{RSSI}$ **or** $SF_{SNR} < SF_{RSSI}$ **then**
8: n ← n+1
9: go to 3
10: **else**
11: $SF \leftarrow SF_{RSSI}$ **or** SF_{SNR}
12: $SNR_{margin} \leftarrow SNR_{avg} - SNR_{req} - Margin\_dB$
13: $NStep \leftarrow \mathrm{int}(SNR_{margin} / 3)$
14: **while** $NStep > 0$ **and** $SF > SF_{min}$ **do**
15: $SF \leftarrow SF - 1$
16: $NStep \leftarrow NStep - 1$
17: **end while**
18: **while** $NStep > 0$ **and** $TP > TP_{min}$ **do**
19: $TP \leftarrow TP - 3$
20: $NStep \leftarrow NStep - 1$
21: **end while**
22: **while** $NStep < 0$ **and** $TP < TP_{max}$ **do**
23: $TP \leftarrow TP + 3$
24: $NStep \leftarrow NStep + 1$
25: **end while**
26: **Output**: TP and SF

B. Simulation Setup

Fig. 3 shows a network scenario of static nodes with a size of 500m * 100m built-in FLoRaWAN. We believe that the network capacity of LoRaWAN in this network scenario can meet the communication requirements of normal-scale ClassA mode nodes. To simplify the model, facilitate the observation and analysis of network indicators, we only arbitrarily arranged 10 ClassA mode sensor nodes and a server model in the scene. Similarly, the network scenario of dynamic nodes in FLoRaWAN is shown in Fig. 4. Compared with Fig. 3, the network scenario of moving nodes has two differences in Fig. 4. The node model becomes a randomly moving model with an arrow, which can be set to different movement trajectories. The other difference is adding a random movement module called Mobility Config, which is used to configure random movement parameters (e.g., movement speed, movement range, etc.) for the mobile nodes. A randomly moving node will randomly move to a location within a specified range of coordinates and wait for 1 second before continuing to move, and so on, during the simulation time according to the movement speed set in Mobility Config. Here the node's movement speed is set to a random value in the interval (5,10)

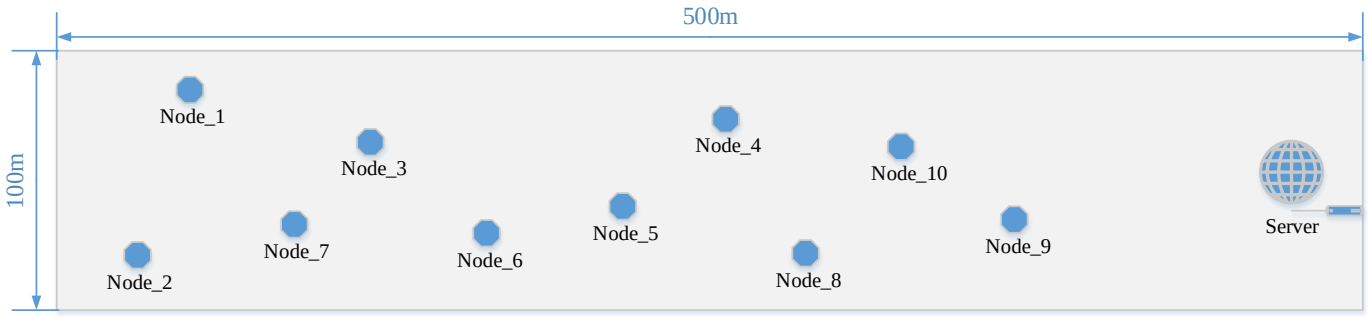


Fig. 3. Network scenario of static nodes.

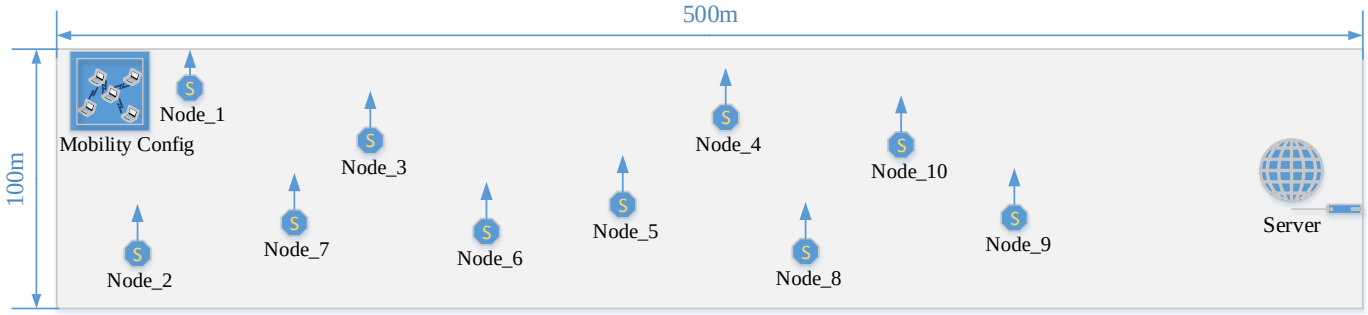


Fig. 4. Network scenario of mobile nodes.

in m/s, and the movement range is the network scenario range.

The OKUMURA-HATA channel model was used in FLoRaWAN to calculate the path loss L_b of each node. The system background noise α of each node was also considered during data transmission, and then got RSSI and SNR_{avg}.

The system background noise α is composed of background thermal noise N_b and background ambient noise N_a, in dBm:

$$\alpha = N_a + N_b \tag{6}$$

$$N_b = (T_{rx} + T_{bk}) \cdot B_{rx} \cdot k \tag{7}$$

$$N_a = B_{rx} \cdot (1.0 \times 10^{-26}) \tag{8}$$

where T_{bk} is background temperture, which is usually set to 290K; k is Boltzmann Constant, $k = 1.379 \times 10^{-23}\ \mathrm{J \cdot K^{-1}}$; B_{rx} is receiver Bandwidth; T_{rx} is receiver temperature, and the expression is as follows:

$$T_{rx} = (NF - 1.0) \times 290 \tag{9}$$

where NF is the Noise Figure of the receiver, typically, the NF of an optimized LoRa gateway receiver is 2.0 dB.

In addition, we also introduced an adjustable noise factor β ($0 \le \beta \le 1$) based on the background noise α, and each node can be set the value of the noise factor β separately. The purpose was to simulate the varying degrees of impact on the RSSI and SNR in the link when the external environment (e.g., rain, snow, etc.) was changed [15].

Therefore, the total noise N of the system can be written as (in dBm):

$$N = \alpha \cdot (1 + \beta) \tag{10}$$

The closer the β value is to 1, the larger the additional noise value loaded on the background noise α, the greater the disturbance to the RSSI and SNR, and the worse the external environment.

The topology of FLoRaWAN is shown in Fig. 5. Where *src* is the service source module, which generates ON/OFF services and service data packets; *sink* is the statistics module, which is used to count the received data (e.g., delay, throughput, etc.); *manager* is the middle layer of the MAC layer and routing, responsible for the data management and forwarding of the upper and lower layers; *LoRa_MAC* is the MAC layer module, which implements the ClassA protocol process of LoRaWAN; *rr_0* and *rt_0* are the transceiver modules respectively.

According to (5), the carrier frequency f, the distance d between the gateway and the end nodes, the effective height h_b of the gateway, and the effective height h_m of the end nodes. The simulation parameters are set in TABLE II.

Finally, in the FLoRaWAN simulation framework, we evaluated the LoRaWAN network performance based on the standard ADR algorithm, ADR+ algorithm, and ND-ADR algorithm. The three indicators of energy consumption, transmission delay, and throughput of the LoRaWAN network have been focused on.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3118051, IEEE Internet of Things Journal

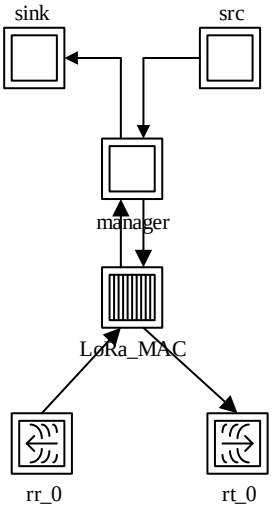


Fig. 5. FLoRaWAN network topology.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Carrier frequency f	868MHz
Gateway effective height h_b	30m
Node effective height h_m	1.5m
Bandwidth (BW)	125kHz
Coding Rate (CR)	4/5
Spreading Factor (SF)	SF6 to SF12
Transmission power	2 - 17 dBm
Payload length	23 Bytes

IV. RESULTS AND DISCUSSION

According to the simulation parameters in TABLE II, we evaluated the LoRaWAN network performance of three different ADR algorithms when the noise factor β was 0, 0.4, and 0.8 in the network scenario of static nodes in Fig. 3. Similarly, we also evaluated the LoRaWAN network performance of three ADR algorithms when the noise factor β was 0 and 0.8 based on the dynamic node model in Fig. 4. Furthermore, we also focused on the optimal number of data packets n dynamically allocated by the ND-ADR algorithm for each static node when β was different values. Finally, we set the duration of each simulation to 2 hours and conducted multiple simulation experiments.

A. β is 0

When the noise factor β of all nodes was 0, it indicated that the external environment was good. At this time, the transmission loss of the network only needed to be considered the loss attenuation of the Hata model.

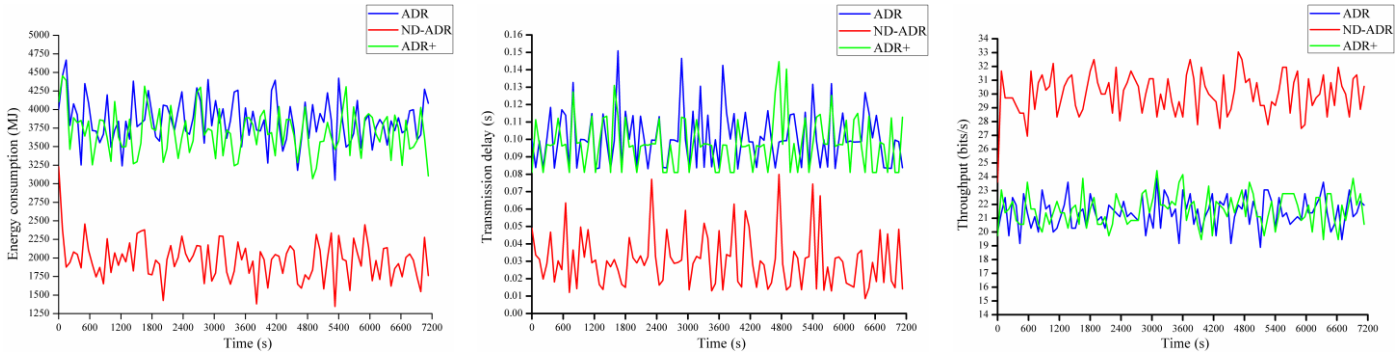
In Fig. 6, the channel transmission is not affected by the external environment. Compared with the standard ADR algorithm, the energy consumption (Fig. 6a), transmission delay (Fig. 6b), and effective network throughput (Fig. 6c) of the ADR+ algorithm are slightly improved. In contrast, the network performance under the ND-ADR algorithm has been significantly improved.

Specifically, we averaged the network energy consumption (Fig. 7a), transmission delay (Fig. 7b), and effective throughput (Fig. 7c). Compared with the standard ADR algorithm, the network energy consumption of the ADR+ algorithm was reduced by about 2%, the transmission delay was reduced by about 4%, and the throughput was increased by less than 1%. However, the network energy consumption of the ND-ADR algorithm was reduced by about 49%, the network delay was reduced by about 70%, and the effective network throughput was increased by about 41%.

Similarly, when the network scenario turned into a dynamic node model, and other conditions remained unchanged, the three ADR algorithms showed a significant increase in network energy consumption (Fig. 8a) and transmission delay (Fig. 8b) and a significant decrease in effective throughput (Fig. 8c) compared to Fig. 6, which indicated that the dynamic channel did seriously affected the adaptive adjustment of data rate and thus the network communication performance. However, compared to the standard ADR algorithm, the network performance of the ADR+ algorithm and the ND-ADR algorithm was both improved. Specifically, we also averaged the three indicators in Fig. 9 (a, b, and c). Compared with the standard ADR algorithm, the network energy consumption of the ADR+ algorithm was reduced by about 1%, the transmission delay was reduced by about 2%, and the throughput was improved by about 1%. While the network energy consumption of the ND-ADR algorithm was reduced by about 34%, the transmission delay was reduced by about 39%, and the throughput was increased by about 31%.

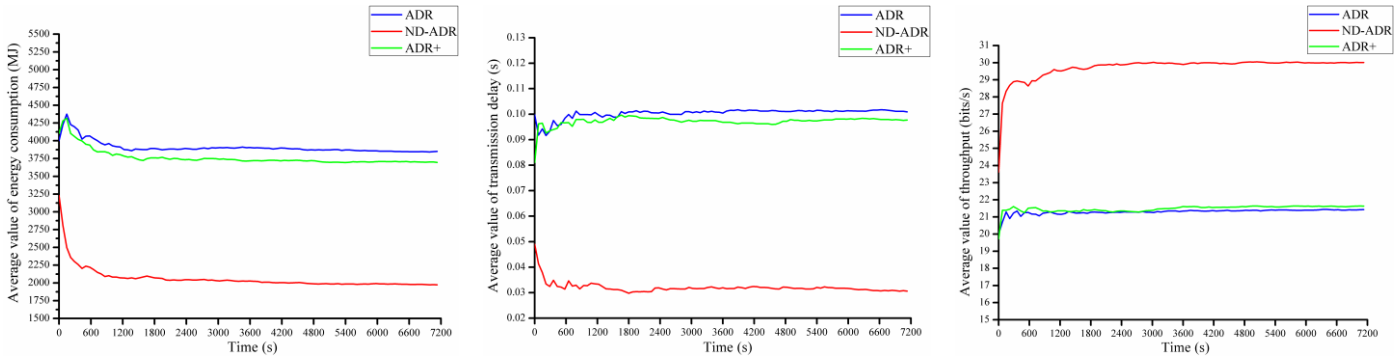
We believe that the ADR+ algorithm based on the average value of SNR as the adjustment basis can have a more accurate assessment of link performance. Still, because the channel conditions are relatively stable, there is no sizeable abnormal fluctuation in RSSI and SNR, i.e., the applicability of the standard ADR algorithm is promising. Therefore, compared with the standard ADR algorithm, the ADR+ algorithm is the same in network energy consumption, transmission delay, and throughput, with only a slight improvement. However, the ND-ADR algorithm reduces the number of selected data packets in a stable channel environment, so the network performance under the ND-ADR algorithm has been dramatically improved. On the contrary, the channel of the mobile nodes changes dynamically in this scenario. Although the ADR+ algorithm can evaluate the link performance more accurately, it still needs to receive 20 packets as the basis for calculation, which significantly increases the probability of packet loss during the fast movement of the nodes. Hence, the improvement of network performance of the ADR+ algorithm in this scenario is not apparent. Similarly, although the ND-ADR algorithm can dynamically adjust the number of packets received according to the link quality, this can only reduce the probability of packet loss during the node's movement and cannot be avoided. Therefore, the network performance of the ND-ADR

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3118051, IEEE Internet of Things Journal



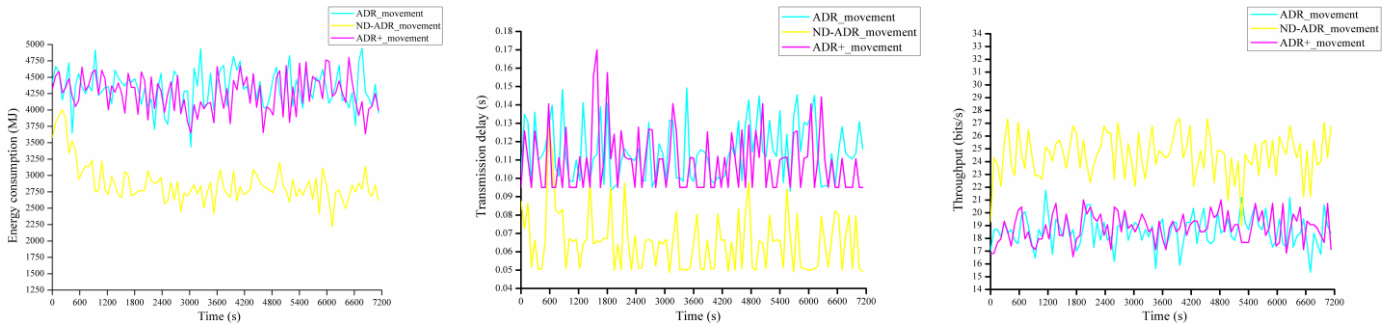
(a) Energy consumption (b) Transmission delay (c) Throughput

Fig. 6. When β is 0, the network performance of three different ADR algorithms in the static nodes scenario.



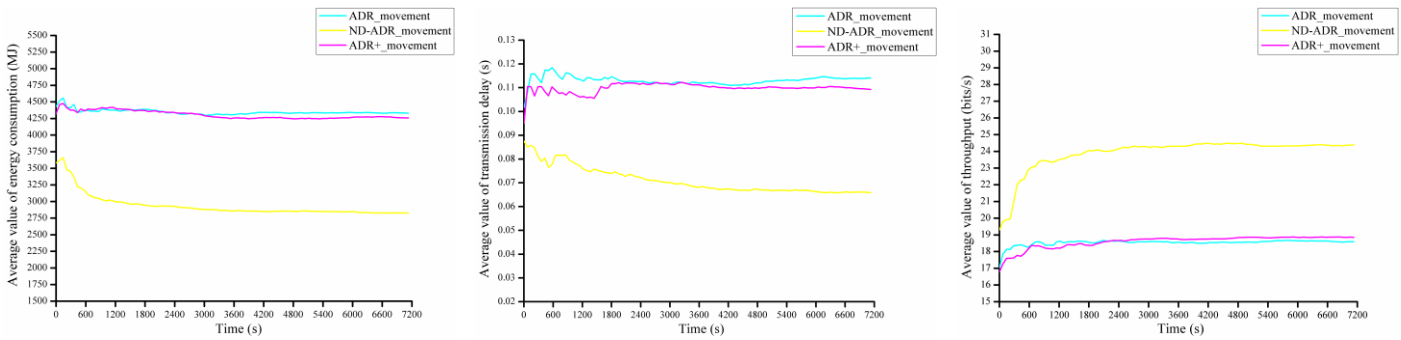
(a) Average value of energy consumption (b) Average value of transmission delay (c) Average value of throughput

Fig. 7. When β is 0, the average value of network performance indicators in the static nodes scenario.



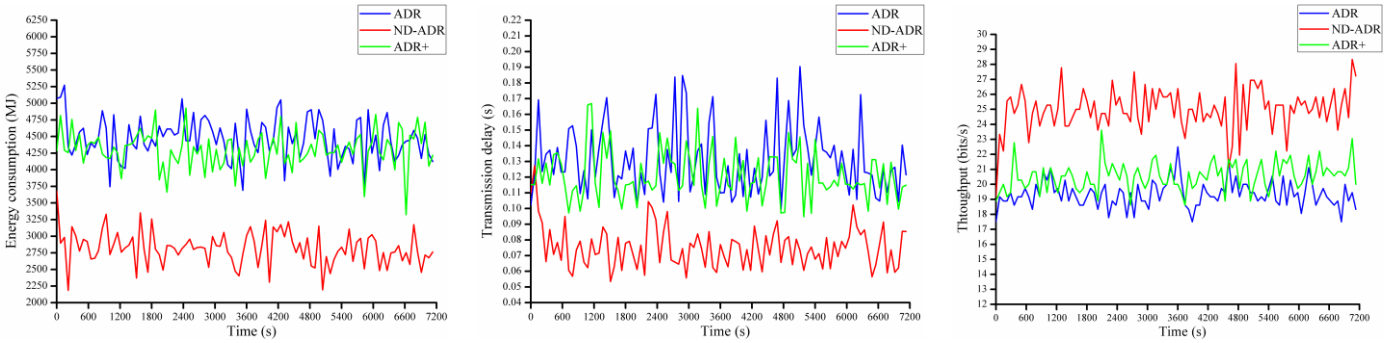
(a) Energy consumption (b) Transmission delay (c) Throughput

Fig. 8. When β is 0, the network performance of three different ADR algorithms in the mobile nodes scenario.



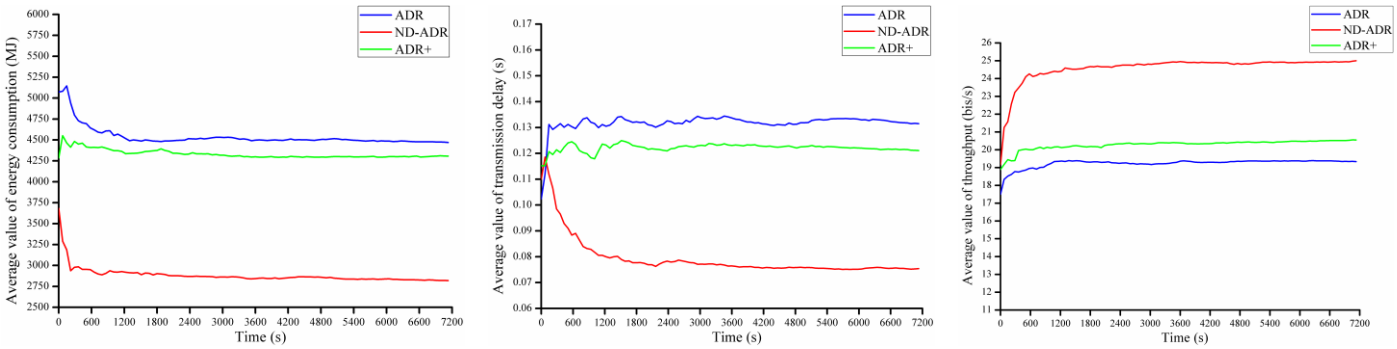
(a) Average value of energy consumption (b) Average value of transmission delay (c) Average value of throughput

Fig. 9. When β is 0, the average value of network performance indicators in the mobile nodes scenario.



(a) Energy consumption (b) Transmission delay (c) Throughput

Fig. 10. When β is 0.4, the network performance of three different ADR algorithms in the static nodes scenario.



(a) Average value of energy consumption (b) Average value of transmission delay (c) Average value of throughput

Fig. 11. When β is 0.4, the average value of network performance indicators in the static nodes scenario.

algorithm in this scenario is still significantly improved. However, it is not as large as the improvement when the channel condition is stable.

B. β is 0.4

When the noise factor β of all nodes was set to 0.4, it indicated that the external environment has begun to be changed but was not severe. At this time, the transmission loss of the network needed to be considered the loss attenuation of the Hata model and the disturbance of the noise factor to the channel transmission.

As can be seen in Fig. 10, when the channel condition is changed moderately, the ADR+ algorithm and the ND-ADR algorithm have different degrees of improvement in terms of network energy consumption (Fig. 10a), transmission delay (Fig. 10b), and effective network throughput (Fig. 10c). Similarly, we averaged the three indicators in Fig. 11 (a, b, and c). Compared with the standard ADR algorithm, we could more intuitively find that the network energy consumption of the ADR+ algorithm was reduced by about 5%, the transmission delay was reduced by about 7%, and the throughput was increased by about 5%. On the other hand, the network energy consumption of the ND-ADR algorithm was reduced by about 35%, the transmission delay was reduced by about 42%, and the throughput was increased by about 29%.

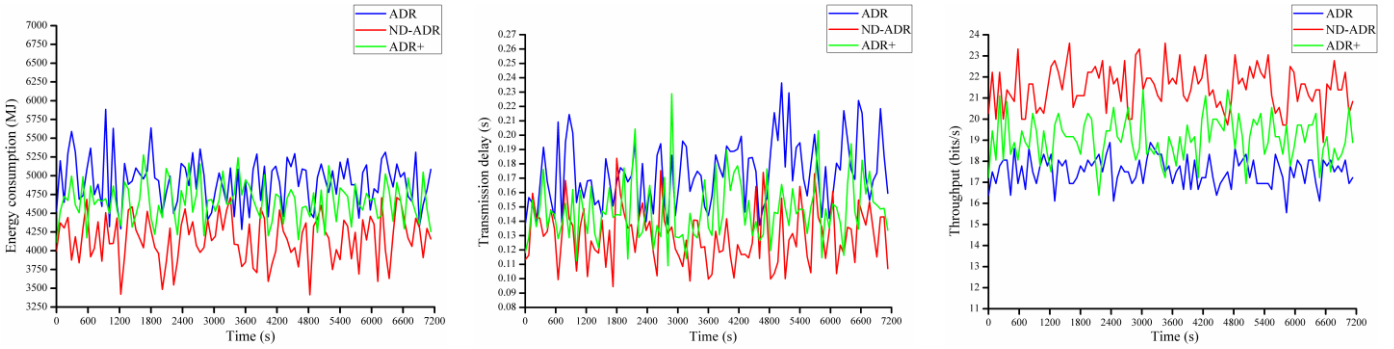
Obviously, when the channel transmission is not only blocked by vegetation but also affected by environmental noise, the ADR+ algorithm has improved significantly in terms of network energy consumption, transmission delay, and throughput. At the same time, the ND-ADR algorithm may increase the number of selected data packets when assessing link quality to cope with the beginning of changing channel conditions. Therefore, the network energy consumption, transmission delay, and throughput (compared with stable channel conditions) under the ND-ADR algorithm have been changed significantly. However, compared with the standard ADR algorithm and the ADR+ algorithm, the network performance of the ND-ADR algorithm is still better.

C. β is 0.8

The external environment has been changed seriously when the noise factor β of all nodes was 0.8. Therefore, compared with the channel attenuation of the Hata model, it was more necessary to pay attention to the disturbance of the noise factor to the channel transmission at this time.

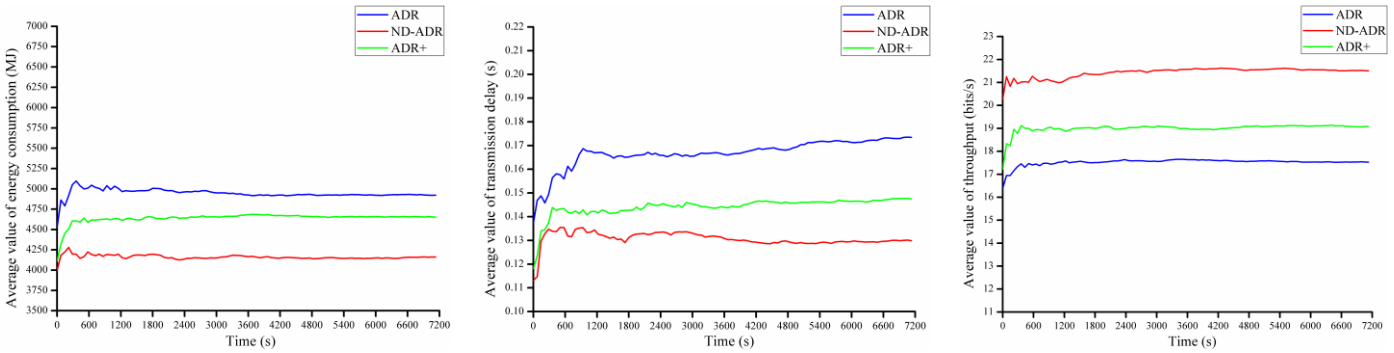
Fig. 12 and Fig. 14 show that when the channel conditions are changed highly, the ADR+ algorithm and the ND-ADR algorithm are changed in terms of network energy consumption (Fig. 12a), transmission delay (Fig. 12b), and effective network throughput (Fig. 12c) in the static nodes scenario. Meanwhile, they are also changed significantly in network energy consumption (Fig. 14a), transmission delay (Fig. 14b), and throughput (Fig. 14c) of the mobile nodes scenario. Similarly, the average values of the three indicators

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3118051, IEEE Internet of Things Journal



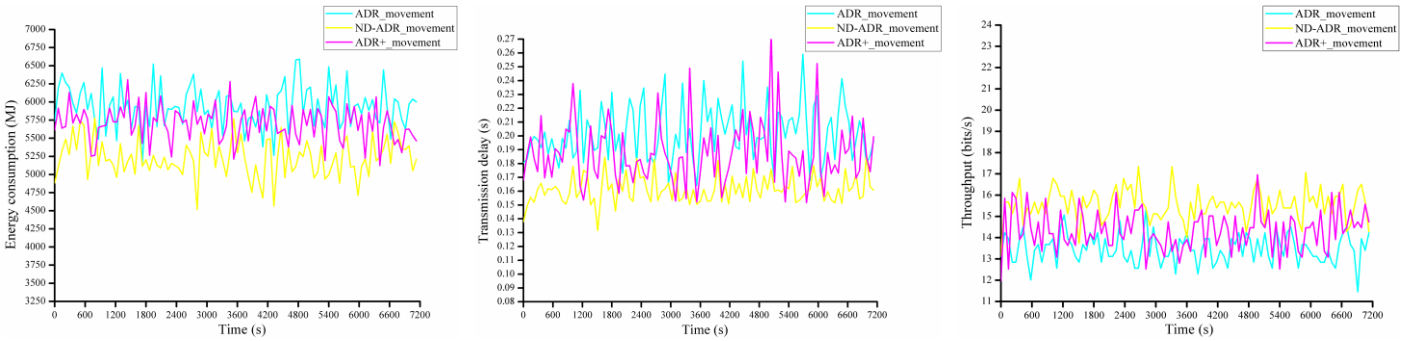
(a) Energy consumption (b) Transmission delay (c) Throughput

Fig. 12. When β is 0.8, the network performance of three different ADR algorithms in the static nodes scenario.



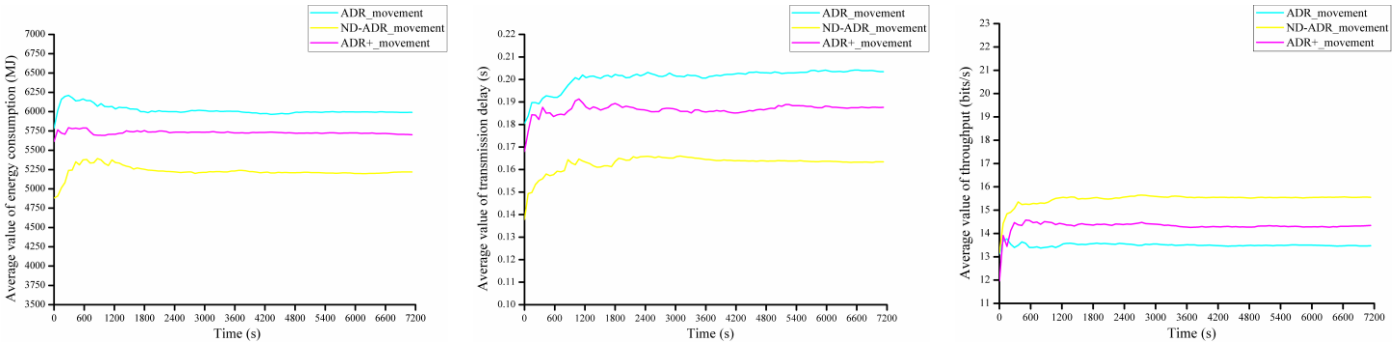
(a) Average value of energy consumption (b) Average value of transmission delay (c) Average value of throughput

Fig. 13. When β is 0.8, the average value of network performance indicators in the static nodes scenario.



(a) Energy consumption (b) Transmission delay (c) Throughput

Fig. 14. When β is 0.8, the network performance of three different ADR algorithms in the mobile nodes scenario.



(a) Average value of energy consumption (b) Average value of transmission delay (c) Average value of throughput

Fig. 15. When β is 0.8, the average value of network performance indicators in the mobile nodes scenario.

were shown in Fig. 13 (a, b, c) and Fig. 15 (a, b, c), respectively. Based on network scenario of static nodes, we could find that the network energy consumption of the ADR+ algorithm was reduced by about 8%, the transmission delay was reduced by about 12%. The throughput was reduced by about 8%. On the other hand, the network energy consumption of the ND-ADR algorithm was reduced by about 18%, the transmission delay was reduced by about 23%, and the throughput was increased by about 21%. In dynamic network scenario, the network energy consumption of the ADR+ algorithm was reduced by about 6%, the transmission delay was reduced by about 8%, and the throughput was improved by about 7%. The network energy consumption of the ND-ADR algorithm was reduced by about 13%, the transmission delay was reduced by about 18%, and the throughput was improved by about 15%.

As we see from Fig. 13b, the channel quality is seriously affected by noise factors. As a result, the transmission delay of the three algorithms was aggravated, and there was still an upward trend. However, Fig. 13c shows that the throughput of the three algorithms only slowly decreases, and the movement is slow and steady. We believe this is due to the powerful demodulation capability of LoRa, which can maintain a specific communication capability even in a harsh environment. Compared to the network scenario with static nodes in Fig. 13, in Fig. 15, we also find a further degradation in the transmission performance of the dynamic channel, and this degradation is particularly pronounced with the standard ADR algorithm. In addition, compared to the static channel (Fig. 3), the transmission performance of the dynamic channel under the ADR+ algorithm degrades more at β = 0.8 than it does at β = 0. That is the opposite in the ND-ADR algorithm, where the degradation of the boosting capability of the ND-ADR algorithm in the dynamic network scenario is particularly significant when the channel condition is good (i.e., β =0) compared to the static nodes scenario. In contrast, the degradation of this boosting capability is significantly reduced when the channel environment is severely disturbed by the noise factor (i.e., β =0.8).

We believe that this phenomenon precisely demonstrates that the ND-ADR algorithm has a better dynamic adjustment capability than the ADR algorithm and the ADR+ algorithm when the channel condition is unstable. The ND-ADR algorithm selects fewer received packets and configures the optimal transmission parameters based on stable channel conditions (i.e., β = 0). However, more received packets need to be added to the ND-ADR algorithm to address packet loss due to node movement and a more accurate evaluation of dynamic link quality. The communication performance improvement capability decreases significantly at this time. Still, this decreasing trend will be slowed down as the channel environment becomes worse (i.e., β is 0, 0.4, and 0.8 in that order) and tends to level off. In other words, the ND-ADR algorithm can maintain a considerable improvement in LoRaWAN network performance even when dynamic channel transmission is performed in a harsh environment.

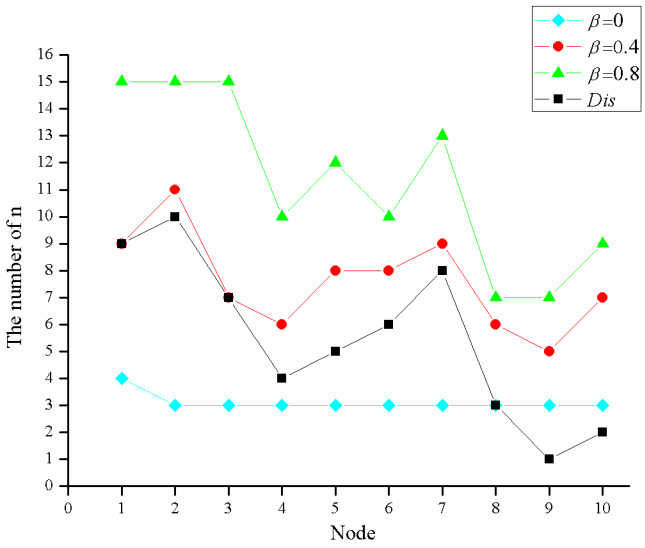


Fig. 16. The value of n when the β value is different.

In addition, we also focused on the value of the number of data packets n selected by the ND-ADR algorithm when the noise factor β was 0, 0.4, and 0.8 for the static nodes scenario, respectively, in Fig. 16.

In Fig. 16, *Dis* represents the ranking of the distances between the nodes and the server in Fig. 3. In this way, the value of n for nodes with different distances at different β values can be observed more intuitively.

As seen in Fig. 16, when the noise factor β is 0, we find that node 1 selects n = 4 data packets, and the remaining nodes only selecte 3 data packets. Therefore, we believe that nodes can pick fewer data packets for data transmission due to the stable external environment. However, non-line-of-sight propagation causes channel attenuation, which will render individual data packets abnormal and cause the ND-ADR algorithm to configure different SFs based on RSSI and SNR_{avg}. Therefore, it is reasonable to increase the number of data packets to re-evaluate the link quality.

When the value β is 0.4, we find that the number of data packets selected by the nodes increases significantly and shows a trend similar to the distance. When the value β is 0.8, the number of data packets specified by the nodes further increases, but we find that nodes 1, 2, and 3 that are far from the server all select the same and the most significant number of data packets, and the value of n by other nodes are not affected by the distance. In other words, the relationship between the changing trend of the importance of n and space is no longer close. We believe that this is due to poor channel conditions. Compared with path loss, the deterioration of the external environment has a more severe impact on channel attenuation.

V. Conclusion

In this paper, we proposed a novel ND-ADR algorithm to address the impact of mobile nodes and harsh environments on data rate regulation. To this end, we developed a simulation framework for the LoRaWAN protocol. We simulated channel transmission in two network scenarios, static and mobile nodes, and introduced an adjustable noise to reflect the perturbation of the external environment on channel transmission. Finally, we evaluated the LoRaWAN network performance of three different ADR algorithms in both network scenarios. Extensive simulation results showed that when the communication channel condition of static nodes was good (i.e., β =0), compared with the standard ADR algorithm, the network energy consumption of the ND-ADR algorithm was reduced by about 49%, the transmission delay was reduced by about 70%, and the effective throughput was increased by about 41%, which significantly improved the communication performance of the LoRaWAN network in static nodes scenario. Moreover, when mobile nodes communicated in a harsh channel environment (i.e., β =0.8), the network energy consumption of the ND-ADR algorithm was still reduced by about 13%, the transmission delay was reduced by about 18%, and the effective throughput was still increased by about 15% compared with the standard ADR algorithm. That demonstrates that the ND-ADR algorithm can maintain a significant improvement in LoRaWAN network performance even when dynamic channel transmission is performed in harsh environments.

Furthermore, the ND-ADR algorithm has better dynamic adjustment ability under the dynamic channel. It can quickly respond to the dynamic changes of the channel and effectively configure transmission parameters to improve communication performance. Therefore, the ND-ADR algorithm is more suitable for LoRaWAN communication in mobile nodes and harsh environments. In the future, we believe that this work can provide a reference for the application and development of LoRaWAN mobile networks in harsh environments, and we will further work on analyzing adaptive regulation strategies for LoRaWAN in multi-gateway scenarios.

References

[1] M. L. Liya and D. Arjun, "A Survey of LPWAN Technology in Agricultural Field," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2020, pp. 313-317.

[2] A. Osorio, M. Calle, J. D. Soto, and J. E. Candelo-Becerra, "Routing in LoRaWAN: Overview and Challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 72-76, Jun. 2020.

[3] N.Sornin and A.Yegin, "LoRaWAN 1.1 Specification," LoRa Alliance Technical Committee, LoRa Alliance, Tech. Rep., Oct. 2017. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawan-tm_specification_-v1.1.pdf, Accessed on: Aug. 2021.

[4] LoRa Alliance, "LoRaWAN What is it? A technical overview of LoRa and LoRaWAN," Nov. 2015. [Online]. Available: https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRaW-AN101.pdf, Accessed on: Aug. 2021.

[5] Semtech Corporation, "LoRaWAN – simple rate adaptation recommended algorithm," 2016. [Online]. Available: https://www.thethingsnetwork.org/forum/uploads/default/original/2X/7/7480e044aa93a54a910dab8ef0adfb5f515d14a1.pdf, Accessed on: Aug. 2021.

[6] S. Li, U. Raza, and A. Khan, "How agile is the adaptive data rate mechanism of LoRaWAN?," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 206-212.

[7] 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*, 2018, pp. 1-9.

[8] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1-6.

[9] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Fair adaptive data rate allocation and power control in LoRaWAN," in *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, 2018, pp. 14-15.

[10] R. Marini, W. Cerroni, and C. Buratti, "A Novel Collision-Aware Adaptive Data Rate Algorithm for LoRaWAN Networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2670-2680, Feb. 2021.

[11] 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 *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2017, pp. 1-8.

[12] J. Babaki, M. Rasti, and R. Aslani, "Dynamic Spreading Factor and Power Allocation of LoRa Networks for Dense IoT Deployments," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1-6.

[13] Y. Li, J. Yang, and J. Wang, "Dylora: Towards energy efficient dynamic lora transmission control," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, 2020, pp. 2312-2320.

[14] N. Jeftenić, M. Simić, and Z. Stamenković, "Impact of Environmental Parameters on SNR and RSS in LoRaWAN," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2020, pp. 1-6.

[15] M. O. Ojo, D. Adami, and S. Giordano, "Network Performance Evaluation of a LoRa-based IoT System for Crop Protection Against Ungulates," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020, pp. 1-6.

[16] O. Elijah, T. A. Rahman, H. I. Saharuddin, and F. N. Khairodin, "Factors that impact LoRa IoT communication technology," in *2019 IEEE 14th Malaysia International Conference on Communication (MICC)*, 2019, pp. 112-117.

[17] A. S. Sethi and V. Y. Hnatyshin, *The practical OPNET user guide for computer network simulation*. CRC Press, 2012. pp. 21-28.

[18] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[19] Semtech Corporation, "LoRa Modulation Basics," [Online]. Available: http://wiki.lahoud.fr/lib/exe/fetch.php?media=an1200.22.pdf, Accessed on: Aug. 2021.

[20] M. Bor and U. Roedig, "LoRa transmission parameter selection," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2017, pp. 27-34.

[21] U. Noreen, A. Bounceur, and L. Clavier, "A study of LoRa low power and wide area network technology," in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2017, pp. 1-6.

[22] D.-Y. Kim, S. Kim, H. Hassan, and J. H. Park, "Adaptive data rate control in low power wide area networks for long range IoT services," *Journal of computational science*, vol. 22, pp. 171-178, 2017.

[23] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks?," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, 2017, pp. 1-6.

[24] Semtech, "SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver," SX1276/77/78/79 datasheet, Aug. 2016. [Online]. Available:https://www.semtech.com/products/wireless-rf/lora-transceiv-ers/sx1261/DS_SX1276-7-8-9_W_APP_V7.pdf, Accessed on: Aug. 2021.

[25] G. Stuber, "Principles of Mobile Communication," Kluwer Academic Publishers, Dordrecht, 1996, 2/e 2001, pp. 85-91.

[26] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317-325, 1980.
[27] D. Bankov, E. Khorov, and A. Lyakhov, "LoRaWAN modeling and MCS allocation to satisfy heterogeneous QoS requirements," *Sensors*, vol. 19, no. 19, p. 4204, 2019.
[28] 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.
[29] R. Kerkouche, R. Alami, R. Féraud, N. Varsier, and P. Maillé, "Node-based optimization of LoRa transmissions with Multi-Armed Bandit algorithms," in *2018 25th International Conference on Telecommunications (ICT)*, 2018, pp. 521-526.
[30] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda, "Toward a Self-Deployment of LoRa Networks: Link and Topology Adaptation," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2019, pp. 1-7.
[31] P. Jörke, S. Böcker, F. Liedmann, and C. Wietfeld, "Urban channel models for smart city IoT-networks based on empirical measurements of LoRa-links at 433 and 868 MHz," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1-6.
[32] N. Benkahla, H. Tounsi, S. Ye-Qiong, and M. Frikha, "Enhanced ADR for LoRaWAN networks with mobility," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019, pp. 1-6.

C. JIANG received the B. E. degree in electrical engineering and automation from Taizhou Institute of Sci.&Tech, Taizhou, China, in 2016. He is currently pursuing the M. D. degree in computer science and technology at Northeast Agricultural University, Harbin, China. His current research interest is wireless communication technology.

Y. YANG graduated from Northeast Agricultural University in Harbin, China in 2020 with a B. E. degree in computer science and technology. She is currently pursuing a M. D. degree in computer science and technology at Northeast Agricultural University in Harbin, China. Her research interest is Deep Learning and Internet of Things.

X. H. CHEN received the B. E. degree in information and computing science from Northeast Agricultural University, Harbin, China, in 2020. He is currently pursuing the M. D. degree in computer science and technology at Northeast Agricultural University, Harbin, China. His current research interest is deep learning and data analysis

J. X. LIAO received the B. E.degree in electronic Information Science and Technology from Shandong University of Science and Technology, Taian, China, in 2020. He is currently pursuing the M. D. degree in computer science and Technology at Northeast Agricultural University, Harbin, China. His current research interest is Internet of Things.

W. X. Song received a B. S. degree in underwater acoustic electronic information and a M.S. degree in underwater acoustic engineering from Harbin Engineering University, Harbin, China, in 2004 and 2009, respectively. He received a PH.D. degree in agricultural electrification and automation from Northeast Agricultural University, Harbin, China, in 2019. He is currently a lecturer in the College of Electronic and Information, Northeast Agricultural University, Harbin, China. His current research interests include Internet of Things and Information processing.

X. H. Zhang (M'11) was born in Harbin, China, in 1979. He received B. S. and M. S. degrees in electrical engineering from Northeast Agricultural University, Harbin, China, in 2002 and 2005, respectively. He received a PH.D. degree in control science and engineering from Harbin Institute Technology, Harbin, China, in 2013. He is currently a professor in the College of Electronic and Information, Northeast Agricultural University, Harbin, China. His current research interests include Internet of Things, embedded systems, as well as deep learning and its application to computer vision and pattern recognition.