






Article

A Novel Approach to Improve the Adaptive-Data-Rate Scheme for IoT LoRaWAN

Yousef A. Al-Gumaei ^{1,2,*} , Nauman Aslam ¹ , Mohammad Aljaidi ³, Ahmed Al-Saman ^{4,*} , Ayoub Alsarhan ⁵ 
and Adel Y. Ashyap ⁶ 

¹ Department of Computer and Information Science, Faculty of Engineering and Environment, Northumbria University, Newcastle Upon Tyne NE1 8ST, UK

² Electrical Engineering Department, Faculty of Engineering, Ibb University, Ibb 70270, Yemen

³ Department of Computer Science, Zarqa University, Zarqa 13110, Jordan

⁴ Department of Manufacturing and Civil Engineering, Norwegian University of Science and Technology, 2815 Gjøvik, Norway

⁵ Department of Information Technology, Hashemite University, Zarqa 13133, Jordan

⁶ Department of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), Batu Pahat 86400, Malaysia

* Correspondence: yousef.al-gumaei@northumbria.ac.uk (Y.A.A.-G.); ahmea@alumni.ntnu.no (A.A.-S.)

Abstract: The long-range wide-area network (LoRaWAN) uses the adaptive-data-rate (ADR) algorithm to control the data rate and transmission power. The LoRaWAN ADR algorithm adjusts the spreading factor (SF) to allocate the appropriate transmission rate and transmission power to reduce power consumption. The updating SF and transmission power of the standard ADR algorithm are based on the channel state, but it does not guarantee efficient energy consumption among all the nodes in complex environments with high-varying channel conditions. Therefore, this article proposes a new enhancement approach to the ADR+ algorithm at the network server, which only depends on the average signal-to-noise ratio (SNR). The enhancement ADR algorithm ADR++ introduces an energy-efficiency controller α related to the total energy consumption of all nodes, to use it for adjusting the average SNR of the last records. We implement our new enhanced ADR at the network server (NS) using the FLoRa module in OMNET++. The simulation results demonstrate that our proposed ADR++ algorithm leads to a significant improvement in terms of the network delivery ratio and energy efficiency that reduces the network energy consumption up to 17.5% and improves the packet success rate up to 31.55% over the existing ADR+ algorithm.

Keywords: LoRaWAN; energy efficiency; adaptive data rate; SNR



Citation: Al-Gumaei, Y.A.; Aslam, N.; Aljaidi, M.; Al-Saman, A.; Alsarhan, A.; Ashyap, A.Y. A Novel Approach to Improve the Adaptive-Data-Rate Scheme for IoT LoRaWAN.

Electronics **2022**, *11*, 3521. <https://doi.org/10.3390/electronics11213521>

Academic Editor: Athanasios D. Panagopoulos

Received: 25 September 2022

Accepted: 16 October 2022

Published: 29 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The long-range wide-area network (LoRaWAN) has been prominent among many wireless communication technologies due to its simple architecture and the utilization of unlicensed Industrial Scientific and Medical (ISM) bands [1]. Furthermore, LoRaWAN has a number of advantages: low cost, low power, and long-range transmission distance, which enable the supply of the wide area for several years. In LoRaWAN, the spreading factor (SF) and the transmission power (TP) of the LoRa end nodes are controlled by the adaptive-data-rate (ADR) mechanism [2]. Based on the channel condition, the ADR algorithm updates SF to determine the appropriate data rate and transmission power to guarantee that the LoRa node achieves successful transmission with low energy consumption. The orthogonality of different SFs leads to an increase in the number of nodes in a single-cell network [3]. Several works proposed algorithms to allocate the SFs and TP, using a diverse set of parameters, including the estimation of a channel state similar to the ADR mechanism. In this article, the ADR+ algorithm, which has been implemented by the FLoRa module and was introduced in [4], was improved. Following a rigorous experimental evaluation of ADR and ADR+,

we observe that both algorithms at the network server (NS) have improved the link budget based on the channel condition of an individual node. The updates of SF and TP in the ADR algorithm were improved based on the maximum value of SNR of the last 20 frames, while the ADR algorithm was based on the average SNR of those frames in the ADR+ algorithm. In addition, most of the previous studies focused on the enhancement ADR and ADR+ for the individual node but did not take into consideration the overall energy consumption of all the LoRa nodes. Thus, we introduced a novel approach to improve the packet delivery ratio and energy efficiency of the ADR+ algorithm using an energy-efficiency controller term that depends on the average energy consumed by all LoRa nodes. The proposed energy-efficiency controller enhances the computing of the SNR margin in the ADR+ algorithm, which results in an improvement of the frames delivery ratio and a significant reduction in energy consumption by LoRa nodes in both urban and sub-urban high-varying channels.

The remainder of the paper is structured as follows. In Section 2, we present an overview of the related work. The ADR and ADR+ configuration and the proposed ADR++ algorithm are introduced in Section 3. The simulation results and the performance of the proposed game are given in Section 4. Finally, the concluding remarks of this paper are made in Section 5.

2. Related Work

In this section, we review a few of these studies that follow the way of ADR that are related to our work.

In [3], the authors estimated the performance of ADR at a network server using the NS-3 simulator and proposed new modifications to the ADR algorithm. The network server increases the spreading factor of nodes when the estimated data-extraction rate (DER) is below 80%. This enhancement leads to a faster convergence data rate for nodes, and significant improvement of the overall packet delivery rate (PDR) in a dynamic network environment. The authors of [5] provide insight into the run-time performance of the official ADR algorithm under dynamic link conditions and various network sizes. They found that if the link conditions change or the number of nodes increases, the convergence time of the ADR mechanism is increased and a large number of packets are therefore lost. The investigation found that the convergence rate can be improved by changing the ADR initialization parameters. Both of the proposed algorithms in [3] and Ref. [5] enhanced the standard ADR algorithm to improve the communication performance in terms of convergence rate. The author of [6] improved the convergence of ADR by using the Gaussian filter to smooth the SNR that was used to update SF and TP. The algorithm was used to reduce the packet losses and therefore the convergence rate of ADR. The proposed approach has been limited to the high variability channel. However, the results only compared with the standard ADR, which has already proven to be an ineffective algorithm in this type of channel by [4].

In [4], a new open-source framework for LoRa simulations based on the OMNeT++ simulator named (FLoRa) was presented to implement the physical and medium access control layers of LoRa. The ADR algorithm was modified such that the maximum signal-to-noise ratio (SNR) measured over last 20 frames was replaced by the average SNR. This modification improved the delivery ratio and energy consumption compared with ADR especially in moderate and high variable channels. The authors of [7] noticed that the ADR link margin parameter can be used to compensate for inaccurate link-quality estimates. This modification led to a better service to nodes while maintaining low levels of energy consumption. These ADR algorithms were based on individual node information but not on network performance.

The authors of [8] have introduced an ordered weighted average (OWA) algorithm to enhance the ADR algorithm in dense LoRa network deployment. The proposed algorithm has shown an improvement on the packet success delivery only in the high channel noise and with the same power consumption of other ADR algorithms. In [9], a collision-

aware ADR (CA-ADR) algorithm was proposed to minimize the collision probability when assigning the data rate. The introduced algorithm considered all the nodes in the network and kept the link performance under control; the results outperform standard solutions when connectivity is good, whereas the algorithm behaves similarly in large areas. In [10], the authors proposed an energy-efficient rate-control scheme based on the uplink channel quality, which allows LoRa nodes to rate up when the channel is better and rate down or increase the transmission power when recognizing failure transmission. The scheme was modelled as a continuous-time Markov chain, and the results show better performance in terms of transmission success probability, transmission rate, frame transmission delay, and energy consumption. An analytical model to evaluate the delivery ratio in a LoRa network has been proposed in [11], in which the impact of several key factors such as multiple gateways and channel variation is considered. The authors of [12] reviewed the ADR improvements that have been suggested in the literature and then described how the proposed E-ADR would handle a node mobility use case. The effect of weather conditions on LoRa IoT has been addressed in [13]. The best equal SINR algorithm BE-LoRa has been proposed in [14] to improve the packet-delivery ratio and minimize the power consumption. The updated SF has been modified based on the received signal strength indicator (RSSI), and the updated transmission power was improved based on the comparison between the optimal SINR of each SF and SNR recorded by last 20 frames. The work was limited to the ideal channel, where the standard deviations in the path loss σ was set to zero. In [15], the authors proposed a new dynamic ADR algorithm to solve the mobility of LoRaWAN and the poor communication quality in harsh environments. An adjustable noise factor β based on the noise factor has been introduced to simulate the varying degrees of impact on RSSI and SNR in the link when environmental conditions change. The above approaches were sought to achieve good performance in power consumption, packet-delivery success rate and convergence. Furthermore, there are two existing approaches that seek to develop the standard adaptive-data-rate algorithm and can be classified into two categories, i.e., link-based approaches, which focus on developing the existing link-adapted ADR algorithm without taking into account the impact of collision [3,6–8,15], and network-aware approaches, which are new development models of power and SF allocation that consider the collision [2,9,10,14]. In this work, we developed the standard ADR+ algorithm by considering the energy consumed per successful transmission computed at a network server to find the best energy-efficiency controller value. The energy-efficiency controller is able to adjust the margin SNR value to ensure efficient power and SF allocation without substantially changing the existing standard ADR.

3. ADR and ADR+ Configuration

The TP and SF are considered as the most important parameters in LoRaWAN. The TP depends on the region, and it varies from 2 to 14 dBm, in 3 dBm steps [4]. The configuration of the SF allows for different transmission data rates, which the lower SF transmits at higher data rate and vice versa. The SFs in the LoRa range from SF7 to SF12, which are semi-orthogonal signals, and each SF has different SNR requirements as a minimum to receive the packet. In fact, selecting different SFs assists the gateway to successfully receive distinct signals at the same time. Moreover, the SNR required to successfully demodulate the frame at the gateway is higher in the lower SF and smaller in higher SF, allowing for longer-range transmission [7]. Table 1 lists the SNR and the bit rate for the LoRaWAN configuration at the bandwidth 125 kHz.

Table 1. LoRaWAN configuration table [16].

Configuration	Bit Rate [b/s]	Required SNR [dB]
SF12	293	−20.0
SF11	537	−17.5
SF10	976	−15.0
SF9	1757	−12.5
SF8	3125	−10.0
SF7	5469	−7.5

Based on Internet of Thing (IoT) application, LoRaWAN has three different classes: Class A, Class B, and Class C [1]. Class A is most popular due to the energy consumption advantage compared to other classes. The successful transmission between LoRa and the gateway occurs when the received power is greater than the receiver sensitivity. The received power P_{rec} depends on transmit power P_t and the path loss due to the attenuation and shadowing. In this work, we used the log-distance path loss model with shadowing, which depends on the distance between the node and the gateway [17]. The received power is computed as follows:

$$P_{rec,i}[\text{dBm}] = P_{t,i}[\text{dBm}] - PL(d_0) - 10n\log_{10}\left(\frac{d_i}{d_0}\right) + X_\sigma \quad (1)$$

where $P_{t,i}$ represents transmit power of node i , $PL(d_0)$ denotes the mean path loss at the reference distance d_0 in [dB], d_i represents the distance between node i and the gateway, n denotes the path loss exponent, and X_σ is a zero-mean Gaussian distributed random variable with standard deviation σ [dB].

The data rate and transmission power are the most important parameters in the physical layer that have a direct impact on the network capacity and scalability of LoRa network. Thus, the ADR algorithm described in LoRaWAN specification is able to modify these parameters to improve the link between the nodes and the gateway. In LoRaWAN, a collision occurs when two or more packets are overlapped simultaneously and frequently at the receiver [14]. In such simultaneous receptions of packets, the signal of the desired packet that needs to be demodulated is compared to the summation of the interfering signals of other packets, which refer to the signal-to-interference ratio (SIR) above a margin. The co-SF interference occurs when interference signals and the desired signal have the same SF, and the inter-SF interference occurs when interference signals have different SF than the desired signal. The margin SIR of both co-SF and inter-SF interference is presented in Table 2.

Table 2. SIR margins [18].

SIR [dB]	7	8	9	10	11	12
7	6	−16	−18	−19	−19	−20
8	−24	6	−20	−22	−22	−22
9	−27	−27	6	−23	−25	−25
10	−30	−30	−30	6	−26	−28
11	−33	−33	−33	−33	6	−29
12	−36	−36	−36	−36	−36	6

3.1. ADR+ NS Algorithm

In LoRaWAN, ADR estimates the link budget between a node and a gateway and then allocates a high data rate with a low SF to the node close to the gateway and a low data rate with a high SF to the far node. In addition, it allocates a low TP to the close node and a high TP to the far node. There are two concurrent algorithms running at the NS and node sides. However, the NS ADR algorithm converges faster than the node algorithm, and it is not necessary that both algorithms converge to the same data rate [3].

The ADR algorithm at the node side has a counter that increments or decrements based on the downlink message. A node requests a response from the network server when the counter reaches the (ADR ACK LIMIT). If the node does not receive any response after the delay time (ADR ACK DELAY), it starts to increase the spreading factor SF till the maximum value is reached. This SF increment leads to an increase in the transmission range and the probability to reach the gateway. If the downlink transmission is received, the counter returns to zero.

On the NS side, the maximum SNR of the last 20 records with the required SNR for each SF and error margin (device margin) in dB is used to compute the number of adjustment steps required to improve the link quality. The number of steps is given by

$$steps = \text{floor} \left[\frac{SNR_{max} - SNR_{req} - deviceMarg}{3} \right] \quad (2)$$

The number of steps computed in (2) is used to decrease SF and adjust transmission power (increase/decrease) by 3 dBm. It was found in [4] that estimating the link quality by using the maximum SNR value is ideal when there is no variability in the channel quality, but it is not the optimal approach when the physical channel conditions are variable. Thus, the authors proposed a simple modification by changing the *max* operator in the NS algorithm by the average function *avg* and referred to this algorithm as ADR+.

The part of ADR running at the NS (ADR+ NS) is illustrated in Algorithm 1.

Algorithm 1 NS ADR+ algorithm

```

1:  $SNR_{ADR+} \leftarrow avg(\text{SNR of last 20 frames})$ 
2:  $SNR_{req} \leftarrow \text{demodulation floor (current data rate)}$ 
3:  $deviceMargin \leftarrow 10$ 
4:  $SNR_{margin} \leftarrow (SNR_{ADR+} - SNR_{req} - deviceMarg)$ 
5:  $steps \leftarrow \text{floor}(SNR_{margin}/3)$ 
6: while  $steps > 0$  and  $SF > SF_{min}$  do
7:    $SF = SF - 1$ 
8:    $steps \leftarrow steps - 1$ 
9: end while
10: while  $steps > 0$  and  $TP > TP_{min}$  do
11:    $TP = TP - 3\text{dBm}$ 
12:    $steps \leftarrow steps - 1$ 
13: end while
14: while  $steps < 0$  and  $TP < TP_{max}$  do
15:    $TP = TP + 3\text{dBm}$ 
16:    $steps \leftarrow steps + 1$ 
17: end while

```

3.2. Proposed ADR++

It is shown in the previous subsection that ADR+ can improve the network performance when the channel conditions are variable. Both ADR and ADR+ are based on the link budget of the individual link between the node and the gateway.

Thus, we propose another modification to the ADR+, wherein the average SNR in Algorithm 1 can be adjusted by decrementing it to find the best average value that ensures efficient energy consumption. The average operator in the first line of Algorithm 1 is multiplied by α , where α represents the energy-efficiency controller of the system. We refer to this new algorithm as ADR++, and it will be applied in high-variability, fast-fading conditions.

Assuming the number of LoRa nodes is N , the NS computes the energy consumption per successful transmission of all nodes E that result from the ADR+ algorithm ($\alpha = 1$), which is E^α as

$$E^\alpha = \sum_{i=1}^N \frac{TE_i^c}{S_i} \tag{3}$$

where TE_i^c is the total energy consumed and S_i the number of packets received successfully.

The ADR++ algorithm updates α in ADR+ again after decrementing α , by a positive value, $\Delta\alpha$. The ADR+ will be run but with multiplying line 1 with the new $\alpha = \alpha - \Delta\alpha$. If the energy consumption $E^{\alpha-\Delta\alpha}$ at this new α improves the energy consumed with respect to the previous instance, the factor α is decremented and the procedure is repeated. We continue until the energy consumption level becomes worse than the previous step; then, we declare the last value of the energy-efficiency controller to be the best factor α_{BEST} . The α_{BEST} value will be multiplied by the first line of Algorithm 1; thus, the computation complexity remains the same as ADR+. The way α_{BEST} is determined by the NS can be summarized as in the Algorithm 2 format.

Algorithm 2 NS ADR++ algorithm

- 1: Set $\alpha = 1$ and use it in NS ADR+ algorithm $\text{avg}(\text{SNR of last 20 frames}) \times \alpha$
- 2: Get E^α energy consumption of all nodes N ,
- 3: decrement $\alpha := \alpha - \Delta\alpha$ and use it in NS ADR+ algorithm
- 4: Get $E^{\alpha-\Delta\alpha}$ energy consumption of all nodes N
- 5: **if** $E^{\alpha-\Delta\alpha} < E^\alpha$ **then**
- 6: Go step 3
- 7: **else**
- 8: declare $\alpha_{BEST} = \alpha$;
- 9: **end if**

Figure 1 represents the whole operation of the proposed ADR++ algorithm in the LoRaWAN node and the network server.

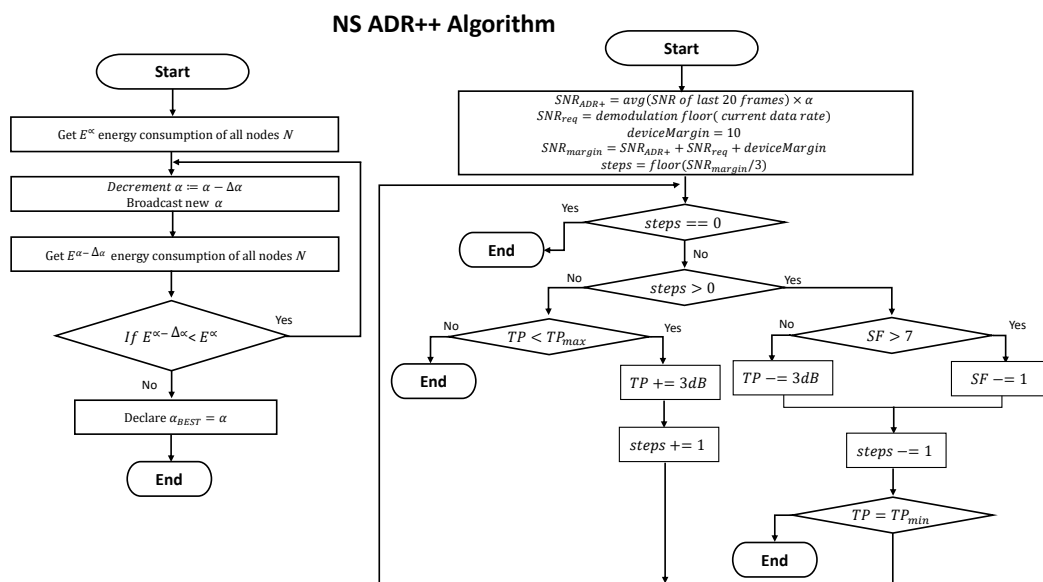


Figure 1. Flowchart of ADR++ algorithm operation in LoRaWAN network server.

4. Results

We evaluate the performance of the proposed ADR++ algorithm in LoRa networks through simulations. We employed FLoRa to evaluate the performance of LoRa networks with ADR+ and with our proposed ADR++ algorithm. Each LoRa node initially used

the maximum spreading factor SF12 and the maximum transmission power level 14 dBm. A uniform distribution of nodes is assumed within a network area of $480 \text{ m} \times 480 \text{ m}$, which adequately represents an urban deployment scenario, and $9800 \text{ m} \times 9800 \text{ m}$, which represents the sub-urban deployment scenario. The urban represents the built-up environment wherein deployments are partially indoor, while the sub-urban corresponds to a few tall-building environments. The default path loss parameters $PL(d_0), d_0, \sigma, n$ used in the two cases shown in Table 3 have been derived from the measurements carried out in [19].

Table 3. Parameters of the propagation model [19].

Scenario	d_0 [m]	$PL(d_0)$ [dB]	n	σ
Urban	40	127.41	2.08	3.57
Sub-urban	1000	128.95	2.32	7.08

A single gateway is located at the center and connected to the network server. The LoRa nodes and the gateway adhered to a 1% duty cycle restriction. We evaluate the performance with the standard deviations in the path loss (σ) set to represent the typical variability ($\sigma = 3.57$) in urban and ($\sigma = 7.08$) in sub-urban scenarios. The bandwidth (BW) is 125 KHz, the frequency is 868 MHz, the device Margin is 10 dB, and the code rate is 4/5. The SNR_{req} for each SF is as in Table 4.

Table 4. Required SNR for each SF.

Spreading Factor SF	7	8	9	10	11	12
SNR_{req} dB	-7.5	-10	-12.5	-15	-17.5	-20

The simulation time for each experiment is 12 days, and the simulations included a warm-up period of 2 days during which statistics were not collected while the network reached steady state. We run 10 iterations of each experiment according to the independent replication method of OMNET++.

The state-based energy consumer module in FLoRa is used to model the energy expenditure of the LoRa nodes. It computes the energy consumed depending on the time spent by the LoRa radio in the main three states: transmit, receive, and sleep state [4]. The energy consumed in the transmit state depends on the values of the instantaneous current of each transmission power level [19]. The sleep state occurs after transmitting or receiving a frame, and the current drawn during the receive and sleep modes is derived from the Semtech SX1272/73 data sheet with a supply voltage of 3.3 V [20].

Figure 2 represents the relation between the energy consumed and the energy-efficiency controller of 100 nodes in the urban scenario, and it is constructed by letting Algorithm 1 complete all the steps at each value of α using $\Delta\alpha = 0.1$. We terminate decrementing the energy-efficiency controller if the energy consumed by all the nodes is worse than the previous step. It is shown that the energy consumed decreases by gradually decreasing α from 1 (as used by ADR+) until $\alpha = 0.7$ is reached, and then it increases when $\alpha = 0.6$. In the proposed ADR++ algorithm, the decrementing was terminated at $\alpha = \alpha_{BEST} = 0.7$, which represents the minimum energy consumption of all the nodes.

Next, we applied the ADR++ algorithm when the network size varies from 100 to 500 in steps of 100 nodes with $\alpha = \alpha_{BEST}$ and compared the output with ADR+. We evaluated the performance by using two metrics, which are

- The delivery ratio: the number of packets correctly received by the network server divided by the total number of messages sent by the end nodes.
- The energy consumed per successful transmission: the total energy used by all the LoRa nodes divided by the total number of messages successfully received by the network server.

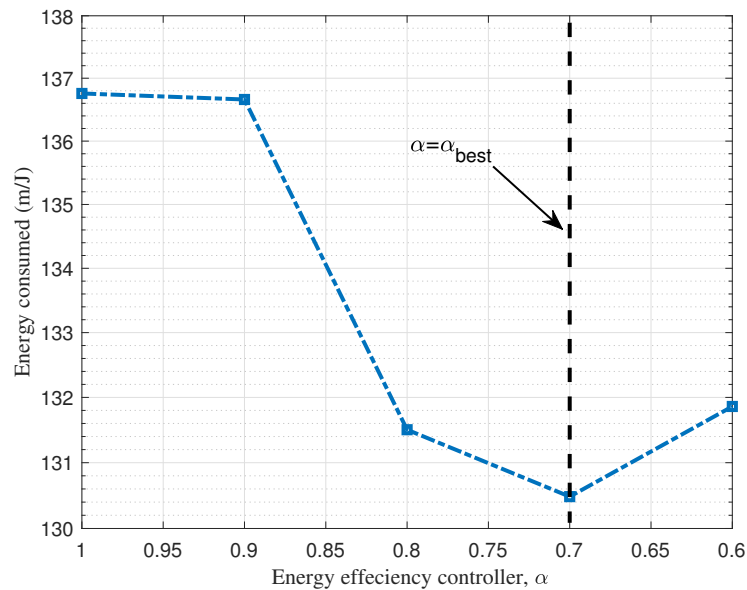


Figure 2. Energy consumed vs. energy-efficiency controller α in urban scenario with 100 LoRa nodes.

In general, the delivery ratio is decreased and the energy consumption is increased by increasing the number of nodes. It can be observed that the solution of ADR++ with $\alpha = \alpha_{BEST}$ offers a significant improvement in the packet delivery ratio with respect to the solution offered by ADR+, as shown in Figure 3a. In the ADR++ algorithm, LoRa nodes are assigned to SFs that guarantee sending packets to the gateway successfully using minimum transmission powers. Figure 3b shows how the power consumption was reduced when the LoRa nodes used ADR++ with $\alpha = \alpha_{BEST}$ compared with the ADR++ algorithm. We note that the number of steps computed in ADR+ were based on the average of the last 20 SNR records of the individual node, while the individual node depends on the overall system energy consumption in the ADR++ algorithm. Choosing α_{BEST} has implications of a max–min fair outcome, for which we start with $\alpha = 1$, which assigns spreading factors and transmission powers based on ADR+ and continues decrement α to improve energy consumption until the latest result is worse than the previous one.

Figure 4 shows the final assignment of SFs to the LoRa nodes. It is obvious that the number of nodes assigned to SF9 and SF10 increases by decrement α and decreases with other SFs that improve the data rate and the delivery ratio.

Furthermore, we applied the ADR++ algorithm in a sub-urban environment and compared the results of the delivery ratio and energy consumed with traditional ADR+. In the sub-urban scenario, the energy consumed is decreased by decrement α , and in the urban scenario. The energy consumed is decreased by decreasing the efficiency controller from 1 (as used by ADR+) until reaching $\alpha = 0.5$, and then it increased when $\alpha = 0.4$. Thus, the ADR++ declare $\alpha = 0.5$ is the best.

Figure 5a shows that the ADR++ algorithm with $\alpha = \alpha_{BEST}$ offers a significant improvement in the packet delivery ratio to above 70% with a different network size compared with ADR+, which achieved less than 60%. Moreover, the power consumption was significantly reduced when the LoRa nodes used ADR++ with $\alpha = \alpha_{BEST}$ compared with ADR++ algorithm, as shown in Figure 5b.

Both the ADR+ and ADR++ algorithms improve the link budget by appropriately selecting the spreading factor and transmission power for each link. The ADR and ADR++ approaches do not take into consideration the possibility of collisions that can significantly decrease the delivery ratio in dense networks [4]. However, the proposed ADR++ algorithm computes the energy consumed per successful transmission of all nodes in different steps of α and then use the best energy-efficiency controller α_{best} to guide each

individual node to the best performance. Updating the SF of each link has a direct impact to the received signal strength (RSS) in which increasing SF will reduce the required RSSI at the receiver by 3 dB.

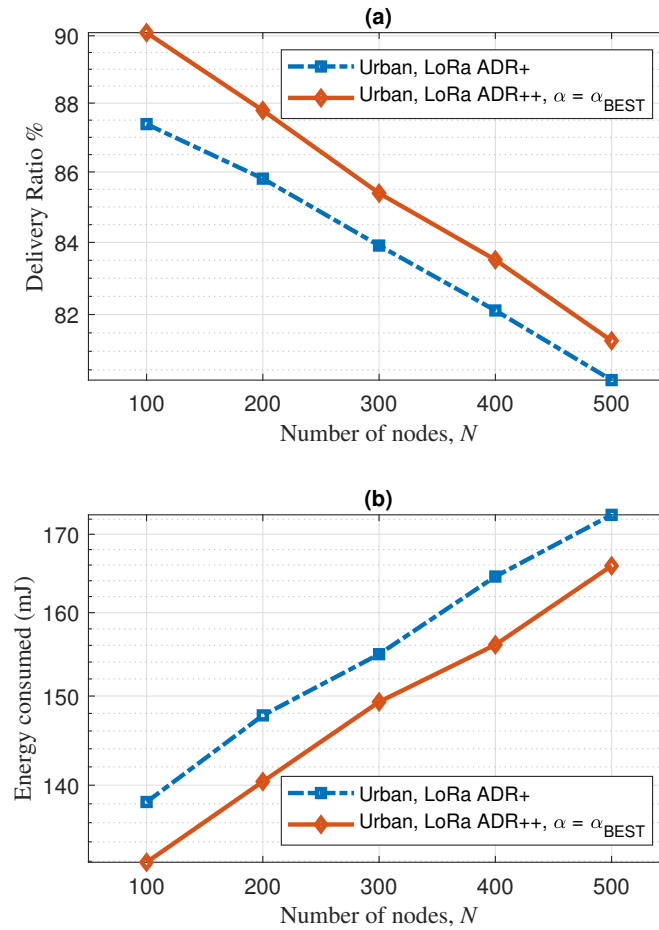


Figure 3. Simulation comparison between ADR+ and ADR++ at α_{BEST} with different number of nodes in urban scenario: (a) delivery ratio; (b) energy consumed.

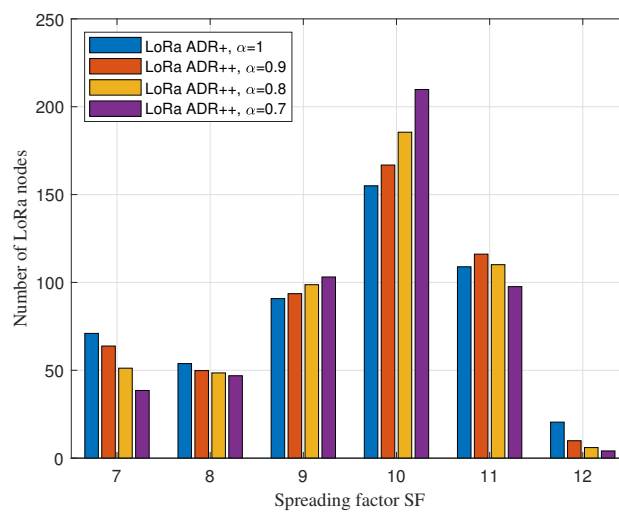


Figure 4. Number of LoRa nodes in each SF with different value of α in urban scenario with 500 LoRa nodes.

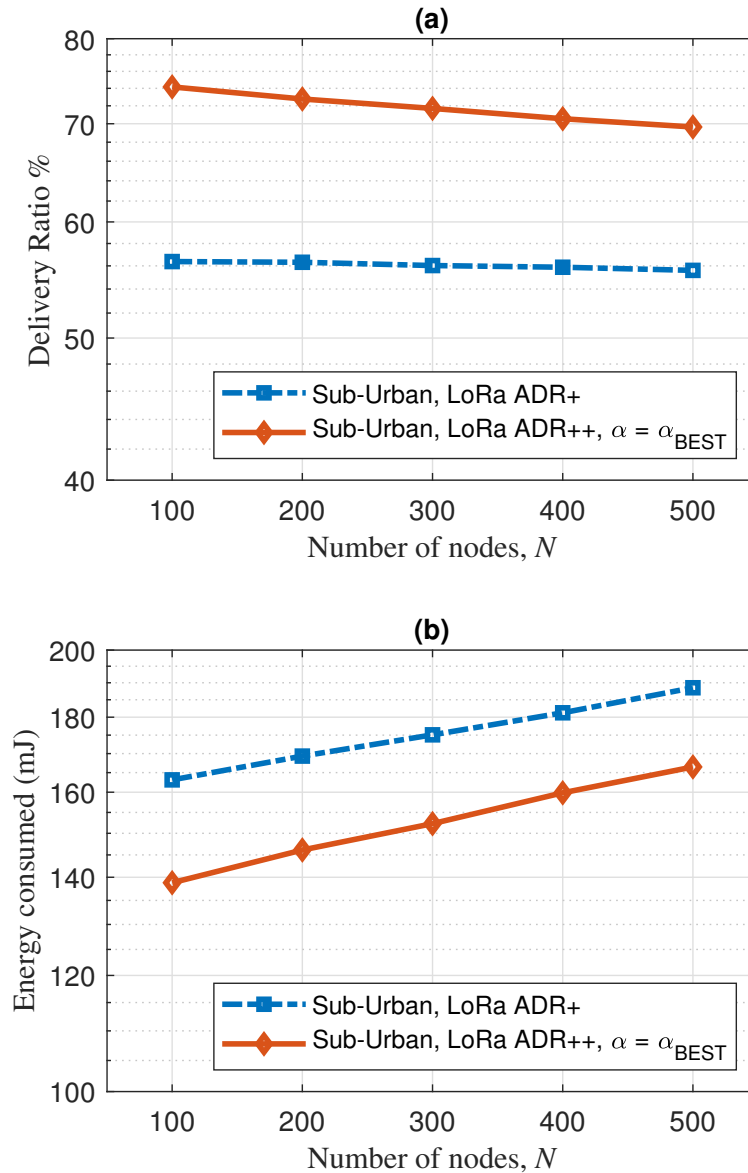


Figure 5. Simulation comparison between ADR+ and ADR++ at α_{BEST} with different number of nodes in sub-urban scenario: (a) delivery ratio; (b) energy consumed.

The summary of performance improvement of ADR++ at $\alpha = \alpha_{BEST}$ shown in Table 5.

Table 5. Numerical results from ADR+ and ADR++ and enhancement ratio in both scenarios obtained when number of nodes $N = 100$.

	Urban		Sub-Urban	
	Delivery %	Energy (mJ)	Delivery %	Energy (mJ)
ADR+	87.39	138.2	56.38	163.1
ADR++	90.09	131.9	74.17	138.8
Enhancement	3.08%	4.77%	31.55%	17.5%

5. Conclusions

In this article, the performance of LoRa networks has been enhanced using adaptive communications and an energy-efficiency controller. In particular, the ADR+ mechanism that was proposed as an efficient mechanism for highly-varying channels was considered. A modification to the link quality by adding an energy-efficiency controller that depends on the overall energy consumption has been introduced. This modification resulted in an improvement in the link budget of the LoRa nodes in terms of the delivery ratio and energy consumption. Thus, extending this work with a general energy-efficiency controller to be applied with ideal and moderate channels is another direction for future research in this area.

Author Contributions: Data curation, Y.A.A.-G.; Formal analysis, Y.A.A.-G., N.A. and A.A.-S.; Investigation, Y. A.A.-G., M.A., A.A. and A.Y.A.; Methodology, Y.A.A.-G.; Project administration, Y.A.A.-G. and N.A.; Resources, Y.A.A.-G., M.A. and A.A.; Software, Y.A.A.-G.; Visualization, Y.A.A.-G., N.A., M.A., A.A.-S. and A.Y.A.; Writing—original draft, Y.A.A.-G. and A.Y.A.; Writing—review & editing, N.A., M.A., A.A.-S., A.A. and A.Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: We thank the University of Northumbria and Cara for their support.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SNR	Signal-to-noise-ratio
SIR	signal-to-interference ratio
RSSI	Received signal strength indicator
ADR	Adaptive data rate
SF	Spreading factor
LoRaWAN	Long-range wide-area network
ISM	Industrial Scientific and Medical
OWA	Ordered weighted average
TP	Transmission power
DER	Data-extraction rate
PDR	Packet delivery rate
NS	Network server

References

- LoRa™ Alliance. LoRaWAN™ Specification v1.0.2. In *LoRaWAN™ 1.0 Specification*; 2016; Volume 1.0.2; pp. 1–91. Available online: https://lora-alliance.org/resource_hub/lorawan-specification-v1-0-2/ (accessed on 24 September 2022).
- Hauser, V.; Hegr, T. Proposal of adaptive data rate algorithm for LoRaWAN-based infrastructure. In Proceedings of the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Prague, Czech Republic, 21–23 August 2017; pp. 85–90. [CrossRef]
- Finnegan, J.; Farrell, R.; Brown, S. Analysis and Enhancement of the LoRaWAN Adaptive Data Rate Scheme. *IEEE Internet Things J.* **2020**, *7*, 7171–7180. [CrossRef]
- Slabicki, M.; Premsankar, G.; Di Francesco, M. Adaptive configuration of lora networks for dense IoT deployments. In Proceedings of the NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018. [CrossRef]
- Li, S.; Raza, U.; Khan, A. How Agile is the Adaptive Data Rate Mechanism of LoRaWAN? In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 206–212. [CrossRef]
- Anwar, K.; Rahman, T.; Zeb, A.; Saeed, Y.; Khan, M.A.; Khan, I.; Ahmad, S.; Abdelgawad, A.E.; Abdollahian, M. Improving the convergence period of adaptive data rate in a long range wide-area network for the internet of things devices. *Energies* **2021**, *14*, 5614. [CrossRef]

7. de Jesus, G.G.M.; Souza, R.D.; Montez, C.; Hoeller, A. LoRaWAN Adaptive Data Rate with Flexible Link Margin. *IEEE Internet Things J.* **2020**, *8*, 6053–6061. [[CrossRef](#)]
8. Babaki, J.; Rasti, M.; Aslani, R. Dynamic spreading factor and power allocation of lora networks for dense IoT deployments. In Proceedings of the 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, UK, 31 August–3 September 2020. [[CrossRef](#)]
9. Marini, R.; Cerroni, W.; Buratti, C. A Novel Collision-Aware Adaptive Data Rate Algorithm for LoRaWAN Networks. *IEEE Internet Things J.* **2020**, *8*, 2670–2680. [[CrossRef](#)]
10. Jeon, W.S.; Jeong, D.G. Adaptive Uplink Rate Control for Confirmed Class A Transmission in LoRa Networks. *IEEE Internet Things J.* **2020**, *7*, 10361–10374. [[CrossRef](#)]
11. Toro-betancur, V.; Premsankar, G.; Slabicki, M.; Francesco, M.D. Modeling Communication Reliability in LoRa Networks with Device-level Accuracy. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021.
12. Benkahla, N.; Tounsi, H.; Song, Y.Q.; Frikha, M. Review and experimental evaluation of ADR enhancements for LoRaWAN networks. *Telecommun. Syst.* **2021**, *77*, 1–22. [[CrossRef](#)]
13. Elijah, O.; Kamal, S.; Rahim, A.; Member, S.; Sittakul, V.; Al-samman, A.M.; Cheffena, M.; Bin, J.; Tharek, A.R. Effect of Weather Condition on LoRa IoT Communication Technology in a Tropical Region: Malaysia. *IEEE Access* **2021**, *9*, 72835–72843. [[CrossRef](#)]
14. Al-Gumaei, Y.A.; Aslam, N.; Chen, X.; Raza, M.; Ansari, R.I. Optimizing Power Allocation in LoRaWAN IoT Applications. *IEEE Internet Things J.* **2022**, *9*, 3429–3442. [[CrossRef](#)]
15. Jiang, C.; Yang, Y.; Chen, X.; Liao, J.; Song, W.; Zhang, X. A New-Dynamic Adaptive Data Rate Algorithm of LoRaWAN in Harsh Environment. *IEEE Internet Things J.* **2021**, *9*, 8989–9001. [[CrossRef](#)]
16. Alliance, L. LoRaWAN™ 1.0.3 Regional Parameters. 2018. Available online: <https://lora-alliance.org/wp-content/uploads/2020/11/lorawan-regional-parameters-v1.1ra.pdf> (accessed on 24 September 2022).
17. Ur-Rehman, O.; Zivic, N. *Wireless Communications*; Cambridge University Press: New York, NY, USA, 2018; pp. 7–21. [[CrossRef](#)]
18. Goursaud, C.; Gorce, J.M. Dedicated networks for IoT: PHY / MAC state of the art and challenges. *IoT* **2015**, *15*, e3. [[CrossRef](#)]
19. Bor, M.C.; Roedig, U.; Voigt, T.; Alonso, J.M. Do LoRa low-power wide-area networks scale? In Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Malta, Malta, 13–17 November 2016; pp. 59–67.
20. Semtech Corporation. *SX1272/73 Datasheet*, Verion 4; Semtech Corporation: Camarillo, CA, USA, 2017.