A Survey of LoRaWAN Adaptive Data Rate Algorithms for Possible Optimization

Charles Lehong
Department of Computer Science
North-West University
Mafikeng, South Africa
lehongcharles@gmail.com

Bassey Isong
Department of Computer Science
North-West University
Mafikeng, South Africa
isong.bassey@ieee.org

Francis Lugayizi
Department of Computer Science
North-West University
Mafikeng, South Africa
francis.lugayizi@nwu.ac.za

Adnan M. Abu-Mahfouz
Council for Scientific and
Inductarial Research (CSIR)
Pretoria, South Africa
a.abumahfouz@ieee.org

Abstract— **The Adaptive Data Rate (ADR) is used in LoRaWAN to ensure optimized data rates and transmission powers for static end devices. However, its implementation is still new, and therefore, its functionalities are limited. Multiple proposals have been made to help improve the ADR but an optimum solution is yet to be found. Therefore, this paper surveyed nine most relevant papers with the goal of identifying the ADR challenges and critically analyzing the solutions put forward to solve them. The critique of the solutions was based on the problems they solve, how they solve them, and the improvements they achieved. Findings show that the poorly designed ADR algorithm results in increased collisions and unfairness in collision probabilities amongst network traffic in LoRaWAN. Furthermore, the findings show that the studies surveyed had limitations in terms of the data rates investigated, the transmission parameters considered, and the region in which they were conducted. To help mitigate some of these challenges, few suggestions are presented in this paper.**

Keywords—LoRaWAN, ADR, Spreading Factor, Transmission Power, Collisions.

I. INTRODUCTION

As the idea of the Internet of Things (IoT) continued to gain popularity and the diversity of interconnected "things" on the internet continued to grow, new technologies to cater for this growth needed to be developed. As a result, Low Power Wide Area Network (LPWAN) emerged as one of such technologies. The rationale for the development of LPWANs was to create networking technologies that were both low cost, energy-efficient and could interconnect thousands of geographically dispersed end devices running on batteries. These were made possible through the development of LPWAN technologies such as Long Range Wide Area Network (LoRaWAN), SIGFOX, Ingenu RPMA, NB-IoT [1] and so on. Among these LPWAN technologies, LoRaWAN is the most widely adopted due to its simplicity, openness, and cost-effectiveness. This technology has a very simple star-of-stars network topology architecture where end devices connect to centralized gateways which forward messages to and from a remote network server. LoRaWAN gateways have the capacity to connect thousands of end devices and their coverage can span multi-kilometers. LoRaWAN end devices are energy-efficient sensors and actuators and can last on batteries for up to 10 years [2] as well as featuring low data rates ranging from 0.3 kbps to 50 kbps [3]. Thus, the technology was built for low demand applications such as smart parking and crop monitoring where speed and reliability are not a priority.

In spite of LoRaWAN benefits, the biggest limiting factor to its performance is the fact that it operates in the freely available Industrial, Scientific, and Medical (ISM) band which is shared with other technologies. Technologies operating in the ISM band are constrained by duty cycle limitations set by regional spectrum regulatory bodies which limit air time usage of the spectrum. Furthermore, LoRaWAN service providers may impose more restrictions which further limit the performance of the network. Because of the constrained network operating conditions together with the size and diversity of LoRa networks, positioning of end devices, and changes in link states due to various reasons such as varying environmental conditions, obstructions, and device movement, it is essential that LoRaWANs have an intelligent network management mechanism capable of dynamically and efficiently managing different communication parameters to ensure an optimized network performance [1]. For this purpose, LoRaWAN employs a mechanism called Adaptive Data Rate (ADR).

ADR is an algorithm used by the LoRaWAN network server to optimize transmission powers and data rates for end devices. LoRaWAN provides transmission parameters Spreading Factors (SF), Bandwidth (BW), Transmission Power (TP), and Coding Rate (CR) which the ADR can manipulate to improve the network's coverage, optimal data rate, energy efficiency, and traffic's robustness to interference. The ultimate result of using ADR in LoRaWAN is improved support for scalability through the addition of new gateways and the increased capacity of the network [4]. Theoretically, the ADR is supposed to be an efficient and reliable network management mechanism, however, various studies have shown that in practice, the ADR suffers from performance issues such as unfairness, slow convergence, and increased collision rate which decrease LoRaWAN reliability and reduce its scalability. Moreover, there are several techniques that have been designed or developed to address the challenges caused by ADR. Therefore, this paper is dedicated to critically studying the challenges and analyzing the goals and limitations of currently proposed ADR techniques for solving these problems. The objective is to identify and provide researchers with important directions to improve ADR in LoRa/LoRaWAN. To the best of our knowledge, this paper is the first of its kind to perform a survey of previous attempts to improve the performance of the ADR algorithm. Thus, the overall contributions of this paper are summarized as follows:

- Provide brief details on the ADR and how it performs its functions.
- Perform a detailed study of currently proposed approaches and techniques to mitigating ADR challenges.
- Provide insights gained from the previous studies to identify new challenges and opportunities for future research.

The rest of this paper is structured as follows: Section II provides a brief background on LoRa/LoRaWAN and the ADR. Section III provides a broad discussion of the results found from studies performed in this paper. Section IV uses insights from our study to provide ideas for possible future work. Finally, section V concludes the paper.

II. BACKGROUND INFORMATION

This section provides a brief background on LoRa, LoRaWAN, and the ADR.

A. *LoRa/LoRaWAN*

LoRa is a proprietary network physical layer owned by Semtech. It is based on the Chirp Spread Spectrum modulation technique and was designed for robust and energy-efficient long-distance communications. The LoRa physical layer can be utilized by any Media Access Control (MAC) layer protocol, but it is popularly used by LoRaWAN [2][5]. LoRaWAN is MAC layer protocol that runs atop of the LoRa physical layer. It defines the network architecture and how gateways and end devices communicate with one another in the network. Its architecture as shown in Fig. 1, is deployed using a star topology for simplicity and cost-effectiveness.

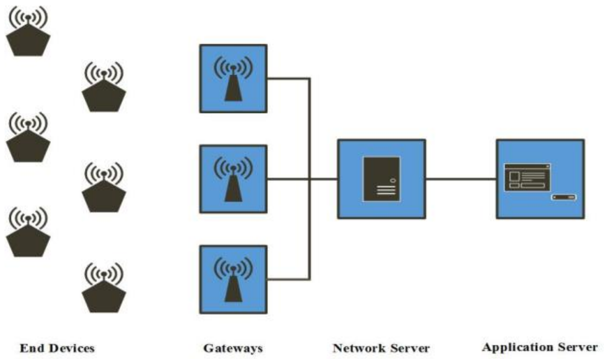

Fig. 1. LoRaWAN network architecture [2]

The end devices are not associated with a particular gateway, instead, all end devices are connected to all gateways in their communication range. When these devices transmit data, the data is received by all the gateways, which then forward the data, via a non LoRaWAN communication link such as Ethernet, towards the network server. To cater for different application types, three different types of device classes are supported: Class A, B, and C [2]. These classes provide a tradeoff between energy consumption and downlink reception delay. Furthermore, the network server handles all the complex network activities such as filtering duplicate data, managing the network, performing security checks, determining the optimal gateway, and performing the ADR. To keep LoRaWANs as energy efficient as possible, the end devices communicate on an ALOHA based method. The devices are asynchronous and spend most of their time sleeping, thus saving more energy and increasing their battery life. LoRaWAN gateways have multichannel multi-modem transceivers that increase their capacity and therefore that of the networks. LoRaWAN also provides security for both the network and the application by ensuring that all the end nodes are authentic and the network operators don’t have access to user data. Furthermore, traffic that traverses the network is end-to-end encrypted using the 128-bit Advanced Encryption Standard. Both LoRa and LoRaWAN were designed for sensors and applications that only need to send small amounts of data every few times a day.

B. *Adaptive Data Rate in LoRaWAN*

The ADR is an algorithm used by LoRaWAN to optimize transmission data rates for static end devices. The algorithm determines the optimal combinations of SF, BW, and TP of end devices to increase their transmission data rates, reduce their transmission airtime, and optimize their energy efficiency. The full specification of the ADR algorithm is given in [6] with the simplified description of the algorithm and its recommended implementation provided by Semtech in [4] and [8] respectively.

The process of ADR in LoRaWAN is as follows: the end devices initially transmit their data using the highest transmission power possible. They notify the network server that they want to use the ADR by setting the ADR bit in one of the uplink packets. Immediately after the ADR is requested, the network server starts collecting 20 uplink messages from the end device. With the 20 collected uplink messages, the following metadata is recorded and used as input to the ADR:

- Frame counter values
- Signal-to-Noise (SNR) ratio values.

From the recorded SNR values, the maximum SNR (SNR_{max}) is determined and with the SNR_{max} value, a margin is computed using the following equation:

$$SNR_{margin} = SNR_{max} - SNR_{limit} - margin_{default} \tag{1}$$

where SNR_{limit} is the minimum required SNR to demodulate the received signal and the $margin_{default}$ is the device-specific installation margin of the network and is typically 10dB in most networks. From the newly computed margin, the NStep which represents the number of times the algorithm is executed is calculated using the following equation:

$$NStep = int\ (SNR_{margin} / 3) \tag{2}$$

where *int* represents the integer part of the resulting value.

- When the NStep value = 0, the device is already using an optimal data rate and TP, therefore the algorithm exits.
- When the NStep value < 0, the end device's data rate and TP are inadequate. The algorithm solves this problem by gradually increasing the TP until it reaches the maximum. The data rate is not adjusted in this instance as all LoRaWAN end devices support automatic data rate decay.
- When the NStep >0, the data rate and the TP can still be optimized. The algorithm first increases the data rate until the maximum is reached and then reduces the TP until it reaches the minimum.

Once the optimized transmission parameters are determined, the network server waits for the next uplink from the end device and then responds with a message containing the newly recommended transmission parameters. The end device then switches its data rate to the newly recommended one. To ensure that connection with the new parameters is not lost, the end device increments the ADR_ACK_CNT counter each time an uplink frame counter is incremented.

- If no response from the network server is received when the ADR_ACK_CNT reaches ADR_ACK_LIMIT uplinks, the end device sets the ADRACKReg bit. Once the ADRACKReg bit is set, there is a number of ADR_ACK_DELAY frames during which the network server must respond with a downlink frame. Once a downlink is received by the end device, ADR_ACK_CNT is reset.
- If no downlink frame is received from the network server by the time the ADR_ACK_DELAY is reached, the end device assumes that the connectivity is lost and tries to regain it by first increasing the TP to its maximum and then reducing the data rate by one step until a response from the network server is received.

If still no response is received, the end device keeps stepping down the data rate every ADR_ACK_DELAY frames until a response is received or until the data rate reaches its minimum. Once the device reaches its lowest data rate, it re-enables all its uplink channels.

III. ANALYSIS OF EXISTING ADR ALGORITHMS AND DISCUSSIONS

Though LoRaWAN is still in its infancy stage, it has gained considerable attention from researchers around the globe. Most of this attention is geared towards finding ways to improve its performance, making ADR algorithm one of the main topics of interest. The main goal of this section is to provide answers to the following research questions (RQs) pertaining to the ADR and to discuss their findings.

RQ1: What are the currently proposed approaches to solving ADR performance issues?

RQ2: How do the currently proposed approaches in RQ1 improve the ADR performance and to what extent were they validated?

To provide answers to the RQs, analyses were performed on nine most relevant papers that focused on developing improved versions of the ADR algorithm. For RQ1, we will briefly discuss the proposed solutions and the problems they attempt to solve. For RQ2, we will describe how these solutions work and how they were validated.

A. What are the currently proposed approaches to solving ADR performance issues?

This section presents the existing approaches of ADR in LoRaWAN. These approaches are discussed as follows: Slabicki *et al.* [9] developed an OMNET++ based open-source LoRa network simulation tool called FLoRa. Using FLoRa, they performed an extensive performance evaluation of the ADR under dense LoRaWAN deployments. The results found show that, although the ADR performs best in stable network conditions, its performance highly degrades in varying channel conditions. To solve this problem, they modified the ADR to support link adaptation and then showed that this modification does improve ADR performance. As a step further, the authors also showed that a network-aware approach, wherein the ADR has global knowledge of the network (e.g. end devices' locations) can be used to further improve the performance of the ADR.

Kima *et al.* [10] developed an improved version of the ADR algorithm that makes data rate optimization decisions based on the knowledge of the network's congestion status. The network's congestion status is determined using logistic regression and when congestion is detected, the end devices are notified of a waiting period for an acknowledgment request. This helps the end devices to avoid assuming the loss of connectivity when acknowledgments get lost due to collisions and thus get rid of the unnecessary use of inefficient data rates. Kim and Yoo [11] also proposed a contention-aware approach to implementing the ADR. In their approach, they used the gradient projection method to determine the optimal distribution of SFs that leads to the maximum throughput for a given network. The results of this implementation ensured that end devices use SFs evenly, thereby promoting network load balancing, increasing fairness in collision probability amongst end devices, reducing network contention, and consequently maximizing network throughput. Moreover, since the rate adaptation of the ADR performs poorly in mobile end-devices due to its slow convergence time and its inability to handle variations in link states, Benkahla *et al.* [12] developed an Enhanced ADR (E-ADR). E-ADR is an improved version of the ADR that supports rate adaption for mobile end devices but also works for static end devices. The algorithm is solely controlled by the network server and therefore, unlike in the ADR, the network server has the capability to increase and decrease transmission parameters accordingly. The algorithm adapts data rates for end devices based on their locations and predetermined trajectory. The E-ADR was designed to minimize airtimes, energy consumptions, and packet error rates in mobile end devices.

Como *et al.*[13], in their work proposed two versions of the ADR algorithm. Firstly, they developed EXPLoRa-SF, an algorithm that equally distributes SFs to groups of end devices based on their RSSI values. The algorithm's goal is to create orthogonal sub-channels within the same bandwidth channel to reduce transmission interferences. Moreover, the second algorithm known as EXPLoRa-AT was built atop functionalities of the EXPLoRa-SF to include the intelligence to equalize the airtimes of transmissions belonging to the different SF groups with the goal of maintaining an equal utilization of the transmission channels. The main goals of these algorithms are to improve the data extraction rate and the throughput of LoRa networks.

Abdelfadeel *et al.* [14] in the same vein developed a Fair Adaptive Data Rate (FADR) algorithm to solve two specific problems in LoRa networks: the unfairness in collision probabilities experienced by the end devices' traffic due to their distance from the gateway; and the capture effect experienced by the end devices' traffic due to varying signal strengths. The proposed algorithm combines fair SF distribution and TP control to ensure that all the traffic in LoRa networks experiences the same fair collision and capture effect probabilities irrespective of their distances from the gateways. Reynders *et al.* [15] in their work also made two main contributions to the optimization of LoRa networks. The contributions include deriving the optimal SF distribution to achieve minimized packet error rates and the proposal of a TP and SF control algorithm for LoRaWAN. The TP and SF control algorithm uses channel constraining to group end devices by channel use and then uses the optimized SF distributions to optimize the end devices' data rates. The algorithm also has a power control intelligence that automatically optimizes the end devices' TPs. The overall objective of the algorithm was to balance packet error rates for both near and far-end nodes and to enhance fairness in transmissions' collision probabilities.

Hauser and Tomáš [16], also carried out an evaluation study on various major open-source implementations of the ADR. The main areas of interest were to find out how these ADR algorithms affect traffic collisions, energy consumption, and network scalability. Using insights gained from this study, the authors then made three major modifications to the ADR functionality targeted at: i) the algorithm's data rate adjustment logic; ii) the calculation of the SNR value used for link margin estimation and; iii) the determination of the NStep when the margin is greater than zero. These modifications were made to improve the ADR in terms of packet error rate reduction and to enhance network scalability support. Bianchiy *et al.* [17] extended their work in [13] to develop a data rate adaptation algorithm called EXPLoRa-C, where C stands for Capture and which employs a sequential waterfalling technique to assign SFs to end devices. This means that some end devices may end up using lower data rates even when higher data rates are possible, but this is compensated by the reduction in the network's packet error rate. The algorithm was also built to support multi-gateway scenarios and it was designed to take advantage of three LoRa characteristics as follows: i) the orthogonality of SFs; ii) LoRa channel capture and; iii) the ability of LoRa signals from a single end device to be received by multiple gateways.

B. *How do currently proposed approaches in RQ1 improve ADR performance and to what extent were they validated?*

Following the preliminary analysis performed on the current literary work done to improve the ADR, we realized that the proposed approaches and techniques in the literature can be generally categorized into two approaches: i) the network-aware approaches, and ii) the resource-constrained approaches. In this section, the implementations of the work proposed in the literature are discussed in detail following these two categories.

1) Network-aware approach: In the works implemented following this approach, the proposed algorithms make data rate optimization decisions based on the general knowledge of a specific characteristic about the entire network. This is evident in the following works:

Slabicki *et al.* [9] work involving the development of FLoRa and the performance evaluation of the ADR algorithm shows that, though ADR optimizes data rates and power efficiency in stable network conditions, it struggles severely in varying network conditions. To address this challenge, the authors firstly replaced the maximum SNR value in the ADR with the average SNR value. The motivation for this was that taking the maximum SNR value to estimate link quality may prove to be rather inaccurate in varying network conditions and therefore, a more conservative approach (i.e. the average) to link quality estimation should be taken. Secondly, the authors modified the ADR to take a network-aware approach to data rate optimization. They called the resulting algorithm ADR+ which sorts the end devices according to their distances from the gateway. Moreover, by using the optimal SF distributions derived in [15], the algorithm assigns to each end device an appropriate SF. This ensures that collision probabilities in the network are fairly distributed thereby making the entire networks fairer. Thus, the main goal of the ADR+ is to reduce the packet error rate and improve the energy efficiency of LoRaWAN communications over noisy channels irrespective of their sizes. In terms of the performance evaluation, FLoRa was used to study LoRa networks with and without the ADR. The simulations were conducted considering two deployment areas: urban and suburban areas involving a network setup of a single gateway with 100 to 700 (varied in 100 steps) end devices randomly distributed around the gateway. Each end device was transmitting a 20-byte packet every 1000 seconds. The end devices respected the 1% duty cycle as required in the EU868 ISM band. The performance of the ADR+ was compared to that of the native ADR using packet delivery ratio and energy consumption as evaluation metrics. The simulation results show a matched performance in terms of delivery ratio and energy consumption for both the ADR and the ADR+ in both urban and sub-urban areas when channel variance is zero or significantly low. Accordingly, ADR performance starts to deteriorate significantly with an increase in channel variation, and thus, the ADR+ performance is much superior in this instance. It is important to note though, that an inferior performance of the ADR in varying channel conditions is to be expected as the algorithm was not built to handle such conditions.

The dependence of the ADR on the reception of acknowledgments to optimize data rates can lead to transmission inefficiencies. For example, in a case where an end device does not receive an acknowledgment from the network server after switching to an optimized data rate, the end device assumes that connectivity is lost and attempts to regain it by automatically lowering its data rate. This behavior can lead to the use of an inefficient data rate even when higher data rates are possible. To solve this problem, Kima *et al*. [10] proposed a network-status-aware ADR algorithm that uses logistic regression to detect the congestion status of the network and then applies the knowledge to determine the best data rate for the end device. That is, when congestion is detected, the algorithm notifies the end device of an acknowledgment request waiting period, helping it to know when the network server may not be able to respond to ADR acknowledgment requests. The performance analysis of the proposed algorithm was carried out with a single gateway and a single end device network using Continuous Time Markov Chain. The network was assumed to be operating in the US902-928 MHz ISM band with only the 125KHz bandwidth being exploited while the end device sent a total of 150 packets (each of 100 bytes in size) to the network server with the data rate being dropped every 50 packets being sent. Transmission delay was the only metric measured while comparing the proposed algorithm with the LoRa-based ADR algorithm. The results show that the delay was significantly lowered than the LoRa ADR especially as the number of transmissions increases.

When the ADR is employed in large networks, a large number of end devices end up using the same SFs. Although different SFs are orthogonal, using the same SFs increases the probabilities of transmission interferences and therefore decreases the overall network throughput. To solve this problem Kim and Yoo [11] proposed a contention-aware approach to the ADR algorithm where the number of SFs that an end device can use is limited by communication range. To this end, the algorithm uses the gradient projection method to determine the optimal SF distribution that maximizes the total throughput per SF while the end device data rate is computed based on the optimized SF distribution. Consequently, the network contention is mitigated, and therefore the throughput is maximized. To assess its effectiveness, the proposed algorithm was compared to two other ADR approaches - the naïve approach and the uniform approach. The naïve approach represented the current ADR standard wherein the end devices all use the lowest SF possible while its uniform counterpart represented the ADR version wherein the number of end devices using specific SFs is made equal. Moreover, they assumed a LoRaWAN operating under EU863-870 ISM band with a 1% duty cycle limitation. SF values considered were SF7, SF8, and SF9. The analysis network consisted of 0 to 10 000 end devices sending 50-100 bytes of data and the parameter of interest was throughput. The overall analysis results showed superior performance of the proposed algorithm compared to the other two ADR algorithms. However, although the algorithm maximizes throughput, it decreases the transmission success rate of the end devices. Therefore, for applications where reliability is a priority, the proposed algorithm needs to be supplemented.

In another study, Khaled *et al.* [18] stated that LoRaWAN suffers from two forms of unfairness because of the near-far problem. One form of unfairness is in collision probabilities where different data rates experience different airtimes and thus, experience different collision probabilities. The other form of unfairness is in data extraction where (weaker) traffic from distant end devices is usually not successfully demodulated when it collides with (stronger) traffic from end devices closer to the gateway, a condition known as "capture effect" [14][18]. To mitigate these challenges, Abdelfadeel *et al.* [14] proposed a Fair Adaptive Data Rate (FADR) algorithm with two main functionalities: data rate and TP allocations. The data rate functionality applies mathematical modeling to derive the fairest data rate deployment ratios possible in the network while the TP allocation functionality is responsible for balancing the TPs of all the end devices in the network to avoid the capture effect. The algorithm's main goal is to achieve the fairest data rate allocations irrespective of the end devices' distance from the gateway while maintaining low power usage. The algorithm proposed in [14] was evaluated in a modified version of LoRaSim (an open-source LoRa simulation tool developed by authors in [5]). The simulation was implemented considering the EU868 ISM band with a single transmission channel and network setups involving a single gateway with a maximum of 4000 end devices transmitting 80-byte packets every 60 seconds. FADR performance was compared to the approaches proposed in [5] and [15] based on the parameters: fairness index, data extraction rate (DER), and energy consumption. The results obtained show an overall superior performance of the FADR in terms of fairness index and DER while being outperformed by the approach in [5] in terms of the overall energy consumption.

Furthermore, Como *et al.*[13] also proposed two ADR approaches of incremental complexity: EXPLoRa-SF and EXPLoRa-AT. EXPLoRa-SF was developed to prove that using higher SF can actually improve LoRaWAN performance. To achieve this, the algorithm divides the network into groups of end devices according to SF use and based on their uplink transmissions' RSSI values. Though this approach can lead to higher SF use and therefore higher airtimes, its drawback is compensated by the reduction in transmission interferences. By using the knowledge gained from EXPLoRa-SF implementation, they developed EXPLoRa-AT which uses an ordered waterfalling approach to assign SFs to end devices. The algorithm then uses some heuristics to the try and equalize the airtimes of all the SF groups which in turn ensures that radio channel usage by the end devices is equalized (when possible) and that all the end devices' traffic experiences the same interference probabilities. Thus, the ultimate goals of these algorithms were to reduce collisions, improve DER, and consequently increase network throughput. The performances of the EXPLoRa algorithms were evaluated using LoRaSim and by considering a single gateway network with a varying number of end devices randomly distributed around the gateway in a 2-dimensional space. The ISM frequency band for the simulations was the EU860 with only the 125 KHz bandwidth being utilized and the end devices were made to send 160-byte messages every 5 to 3600 seconds. DER and throughput were the performance metrics of interest and the

TABLE I. SUMMARY OF LORAWAN ADR APPROACHES

Ref	Challenges Addressed	Proposed Solutions	Critical Analysis	Tools Used
[9]	Poor ADR performance under varying channel conditions.	A network-aware approach to data rate adaptation.	FLoRa was developed and used to evaluate the performance of the ADR. The results showed a poor performance of the ADR in terms of data rate optimization and energy consumption. Hence two solutions were proposed as follows; i) the SNR_{max} was replaced with $SNR_{average}$ for a more conservative link margin estimation and; ii) a network-aware approach using optimal SF distributions was introduced into the ADR algorithm.	FLoRa.
[10]	Unnecessary switching of end devices to inefficient data rates due lost acknowledgements.	A network-status-aware approach to data rate adaption - network congestion status determines the best time for end devices to make acknowledgement requests.	Due to ADR dependence on the reception of acknowledgements, data rate adaptation decisions can lead to performance inefficiencies. Thus, a network-status-aware ADR approach which uses logistic regression to determine the network's congestion station was proposed. The Congestion status is then used to notify the end devices of the best time to make acknowledgement request to avoid unnecessary data rate decay due to lost acknowledgements.	Mathematical modeling using Continuous Time Markov Chain.
[11]	ADR leads to an increase in collisions probabilities and a decrease in throughput in dense networks due to more devices using the same SFs.	Contention-aware approach to data rate adaptation with restricted total number of end devices that can use a particular SF.	ADR approach in dense networks leads to collisions and reduced throughput due to more end devices using the same SFs. The proposed contention-aware approach therefore, applies gradient projection method to find the SF distributions that maximize the throughput for each SF and compute the end devices' data rates based on these distributions. This ensures that network contentions are mitigated and network throughput maximized.	Not specified.
[12]	ADR performance is inefficient in mobile end devices.	An enhanced E-ADR that uses the end device's position and trajectory.	E-ADR was developed for mobile end devices. It applies trilateration method to determine the end device's location and then combines it with the end device's predetermined mobility model to compute the optimized data rate for the end device at each location.	Waspmote end devices and gateways.
[13]	Increase in transmission interferences caused by simultaneous channel occupiers in ADR-enabled networks.	Two algorithms (EXPLoRa-SF and EXPLoRa-AT) – techniques for optimizing SF distributions.	EXPLoRa algorithms reduce transmission interferences and increase throughput by taking advantage of SF orthogonality and end devices' communication range. EXPLoRa-SF creates orthogonal channels by equally distributing SFs amongst groups of end devices, while EXPLoRa-AT equalizes airtimes for each SF group via ordered waterfalling to promote a balance in channel usage, and to ensure uniformity and fairness in collision probabilities.	LoRaSim
[14]	Near-far problem where distant end devices experience unfairness in terms of collision probabilities and capture effect.	FADR algorithm which enforces data extraction fairness in LoRaWANs.	FADR is designed to mitigate the unfairness in collision probabilities experienced by traffic from distant end devices using two functionalities: i) derive the fairest distributions of SFs and; ii) balance TPs for all end devices to prevent capture effect occurrence.	LoRaSim
[15]	Inability of nodes from distant end devices to successfully transmit due capture effect.	TP and SF control algorithm that uses channel constraining to optimize network fairness.	A TP and SF control algorithm is developed to solve the capture effect unfairness in LoRaWANs by sorting end devices by their distances from the gateway, grouping them by channel use, and assigning each channel group a SF factor based on the derived optimal SF distributions. This reduces collisions since near and distant end devices each have their own channels to transmit in and thus reduces the possibilities of channel capture effect.	NS-3
[16]	Possibility of end device to indefinitely lose connectivity to the network server with the newly optimized data rate.	Three minor modifications to the ADR algorithm implementation: data rate adjustment, SNR determination, and link margin estimation.	The performance of ADR algorithm is improved by i) decreasing the data rate before increasing TP when the NStep < 0 to avoid indefinite loss of connectivity to the network server; ii) replacing the SNR_{max} with the SNR_{avg} for a more conservative SNR estimation and; iii) introducing a hysteresis into the link margin estimate calculation to avoid oscillations when the estimate lies between two decision levels.	MATLAB
[17]	Increase in collisions due to multiple end devices using the same SFs in ADR-enabled networks.	ADR algorithm that forces some end devices to use higher than recommended SFs to reduce collision probabilities.	EXPLoRa-C extends EXPLoRa algorithms to support multi-gateway scenarios. It uses an ordered waterfalling approach to assign SFs to end devices and has the following design goals: i) to equalize airtimes for all SFs; ii) to balance SFs amongst gateways and; iii) to consider channel capture when determining data rates.	LoRaSim

simulations were run in three different scenarios: 1). DER was measured against an increase in the message rate in a network consisting of 500 end devices distributed in a circular manner around the gateway over a 50-meter radius, 2). DER was also measured with the end devices now increased to 1000 and each sending a message every 60 seconds, and 3). Throughput was measured against the message rate in a network consisting of 2000 end devices distributed around the gateway with a 200-meter communication range. Both algorithms' performances were compared against the native ADR's performance and the overall results favored the EXPLoRa algorithms. However, the native ADR outperformed the EXPLoRa algorithms in terms of throughput in low message rate scenarios.

2) Resource-constrained approach: In the works implemented following this approach, the proposed algorithms made data rate optimization decisions following resource constraints (such as limitation by channel use) imposed on the end devices as evident in the following:

In [15], Reynders *et al.* discussed the packet error rate unfairness in LoRa networks caused by the near-far effect problem. In this challenge, LoRa network traffic from end devices furthest away from the gateway is likely not to be successfully received by the gateway as it may collide and be absorbed by traffic from end devices closer to the gateway. To address this challenge, TP and SF control algorithm was proposed to have the capability to categorize end devices by their distance from the gateway and by channel use. It then assigns each group an SF based on the optimized SF distribution values derived using random access formulae. In this case, since the end devices are constrained by channel use, traffic from end devices furthest away from the gateway no longer shares channels with traffic from end devices closer to the gateway. This ensures that transmissions from each end device are only affected by traffic from the same transmission group. Thus, this keeps collision probabilities uniform per group and ensures that each end device's traffic experiences fairness in packet error rate. The effectiveness of the proposed algorithm was validated in NS-3 [19] using a single gateway network with 1000 end devices randomly distributed around the gateway over a 1000-meter radius. Accordingly, the end devices where each transmitting 85-byte packets to the gateway every 600 seconds using one of the three available transmission channels. The algorithm's performance was compared to the algorithm proposed in [20] where TPs and SFs are always the lowest and channel selection is random. In this study packet error rate was the only performance metric considered. The results obtained show that the algorithm achieved a 50% reduction in packet error rate for end devices furthest away from the gateway when compared to the algorithm in [20]. In essence, the proposed algorithm reduced the overall network packet error rate by 42%. Though the algorithm reduces the packet error rate for end devices further away from the gateway, it also increases it for end devices closer to the gateway.

In a similar study, Bianchiy *et al.* [17] argued that a data rate adaptation algorithm that makes its data rate computation decisions solely based on the link budget cannot fully take advantage of the orthogonality of the LoRa SFs. For instance, if a LoRa network consists of end devices that are closer to the gateway, the traditional ADR algorithm will force all the end devices to use the lowest data rate possible. This data rate is most likely to be the same for most, if not all, end devices and this may lead to increased collisions even though these collisions can be avoided by simply switching to lower unused data rates. Thus, this challenge is mitigated in [17] by developing EXPLoRa-C, a data rate adaptation algorithm that assigns SFs to end devices following a sequential waterfalling technique. EXPLoRa-C was designed as an extension to the EXPLoRa algorithms proposed in [13] to support multi-gateway scenarios. Its development process is governed by three design goals as follows: i) to equalize the Time-on-Air (ToA) for every SF group; ii) to ensure a balanced distribution of SFs amongst gateways, and iii) to take into account the channel capture when determining the data rates. EXPLoRa-C was implemented and evaluated using LoRaSim based on two scenarios: a single gateway scenario and a multi-gateway scenario. In a single gateway scenario, a varying number of end devices (from 100 to 4000) randomly distributed around the gateway in a 12km and a 34km radius were considered. In the multi-gateway scenario, two topologies of up to 8000 end devices were considered as follows: i) 25 gateways were placed within regular distances from each other and no border effects were considered and; ii) a partitioned network with 3 gateways and border effects was considered. The simulations were done considering the EU860 ISM band with only the 125 KHz bandwidth being utilized and the end devices were transmitting 20-byte packets at a frequency of 1 packet every 90 seconds while DER was the parameter of interest. Accordingly, the EXPLoRa-C algorithm was compared against the EXPLoRa-SF, EXPLoRa-AT, and an algorithm that randomly distributes SFs to end devices. The results obtained showed the superiority of EXPLoRa-C against other algorithms, however, with the claim that EXPLoRa-C outperforms the native ADR algorithm, there were no results that support this claim.

3) Other approaches: Works presented in this section are generally based on tweaking the ADR control knobs or modifying minor functionalities of the ADR. Such approaches are discussed.

The current implementation of the ADR can only be used in static devices and devices that spend most of their time stationary. For mobile devices, the ADR cannot be used because it was not designed to handle the changes in link states that come with end devices' mobility. Therefore, Benkahla *et al.* [12] addressed such challenges via designing and implementing an algorithm called Enhanced Adaptive Data Rate (E-ADR) which is purely network server controlled. Unlike in current ADR, the network server can increase or decrease the transmission parameter values accordingly. The rationale for this algorithm is to estimate the end device's current location based on its predefined mobility trajectory and the trilateration method. A position estimation technique that uses three gateways positioned in a triangular manner to estimate a device's location using the collected RSSI values of its transmissions. Once the device's current location has been determined, the network server uses it to determine the RSSI interval which can be used to best determine the device's recommended optimized transmission parameters. To assess E-ADR performance, a practical experiment using Waspmote SX1272 end devices was carried out. For location purposes, the join-request packets were modified to include the end devices' predefined trajectory. The algorithm performance was tested on the following use cases: i) cleaning robot; ii) parcel inspection drone; ii) fruit and vegetable growth monitoring robot and; iv) feeding system and temperature sensor. The network topologies employed had three gateways and five end devices and the data rate adaptation was configured to occur every 3 transmitted packets. In addition, the network performances were measured based on ToA and energy consumption and were compared to that of the ADR algorithm. The results obtained for all the tested use cases showed superior performance by the E-ADR. However, the native ADR would be expected to perform poorly for mobile end devices as it was only designed for static end devices.

In the same vein, Hauser and Tomáš [16] carried out a performance analysis of the LoRaWAN ADR in terms of packet error rate, power efficiency, and scalability. The results obtained were used as a basis to propose three improvements to the ADR algorithm summarized as follows: i) decreasing the data rate before increasing the TP when the NStep < 0 to avoid failure of the end device to re-establish a connection to the network server in instances where ADR_ACK_DELAY cannot be reached; ii) using the average instead of the maximum of the 20 collected SNRs when optimizing the data rate to avoid the impact of insignificant outliers on the link margin estimation; iii) introducing hysteresis into the algorithm to help avoid the oscillation of the algorithm when the link margin estimate lies between two decision levels. The essence of these ADR improvements was to reduce the number of uplink and downlink transmission in LoRaWANs in order to create more transmission opportunities for other end devices and to reduce transmission collisions. To evaluate the performance of the LoRaWAN ADR algorithm based on the modifications, physical experiments, and simulations were performed. The experiments were performed in a college building in the evening using the IMST iC880A-SPI gateway and IMST iU880A-USB end device operating in the EU 863MHz to 870mHz ISM band. On the other hand, the simulations were performed using MATLAB where message exchange between an end device and a gateway were modeled. With the simulation, each end device was made to sequentially send 2000 link check requests with the largest payload possible and the gateway responded accordingly. Moreover, the modified ADR was compared against “the ADRs implemented in major open-source projects”. The performance metrics of interest for the experiments were output energy, packets, and bytes received by the gateway and packets sent by the end device, all measured against Gain. The results of the experiments showed an overall superior performance of the proposed ADR version in terms of error reduction and network scalability improvement. However, the implementations of the “major open-source ADR algorithms” considered in these experiments were neither specified nor referenced so their functionalities are not known and therefore, the results presented in this study may be questionable.

TABLE I provides a summary of the papers surveyed in this research. It summarizes the challenges addressed in each study, the approaches proposed as solutions, critical analysis of the proposed ADR algorithm, and the tool used to evaluate the performance and effectiveness.

IV. OPPORTUNITIES FOR FUTURE RESEARCH

The majority of the algorithms proposed in the papers we studied were mainly focused on data rate optimization in terms of spreading factor and transmission power control. However, the determination of LoRaWAN data rates also depends on the bandwidth and coding rate selection. Therefore, to fully study data rate optimization, the future implementations of the proposed ADR algorithms also need to consider these transmission parameters.

Another limitation in the current implementations of the ADR is that they were designed to only operate in the European ISM bands which means they are limited to the use of only six data rates and only the 125 kHz bandwidth. Therefore, a future challenge for ADR designers is to design algorithms that are able to operate in other ISM bands and take full advantage of all the available data rates and bandwidths. Another future challenge for researchers and developers in the development of an ADR algorithm that supports mobile end devices. As things stand, the current ADR algorithm only supports static end devices, leaving the mobile end devices to self-adapt their own data rates. In the papers we surveyed, only one paper was focused on an ADR algorithm for mobile end devices. Although this paper provides a good foundation for the development of a rate adaptation algorithm for mobile end devices, this topic is still understudied and needs more attention from researchers.

V. CONCLUSION

In order to be able to support a large fraction of the billions of end devices and sensors projected for the IoT, LoRa/LoRaWAN will need to have high capacity and be scalable, reliable, and energy-efficient. To meet some of these requirements, LoRaWAN employs the ADR algorithm which optimizes the end devices’ data rates and energy consumption for improved network performance. However, the ADR is still fairly new, its functionalities, and therefore, its performance is still limited. Multiple proposals have been made by researchers in academia as steps towards improving the ADR algorithm. Therefore, this paper performed and presented a survey of the currently proposed approaches and techniques for solving the ADR problems. The analysis findings show that a lot of progress has been made towards finding the ideal implementation of the ADR algorithm. However, most of the proposed approaches are limited to the European ISM band operation which means the data rates that they can achieve are limited to six options with only the 125 kHz bandwidth being available for utilization. Therefore, more research and development need to be done to enable the proposed algorithms to operate in other ISM bands and be able to take advantage of more available data rates and bandwidths. As part of our future work, we intend to improve the performance of the ADR by extending its data rate support and by enabling it to take full advantage of all the available LoRa bandwidths.

ACKNOWLEDGMENT

This research was supported by FRC, Department of Computer Science at the North-West University Mafikeng campus and the Council for Scientific and Industrial Research

(CSIR), through the Smart Networks collaboration initiative and IoT-Factory Program (Funded by the Department of Science and Innovation (DSI), South Africa).

REFERENCES

[1] S. Li, U. Raza and A. Khan, "How Agile is the Adaptive Data Rate Mechanism of LoRaWAN?" *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, 2018, pp. 206-212, doi: 10.1109/GLOCOM.2018.8647469.

[2] LoRa Alliance, *LoRaWAN, What is it?: A Technical Overview of LoRa and LoRaWAN.* Camino Ramon, San Ramon, USA: Semtec Corporation, 2015.

[3] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui and T. Watteyne, "Understanding the Limits of LoRaWAN," in *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34-40, Sept. 2017, doi: 10.1109/MCOM.2017.1600613.

[4] Semtec Corporation, *Understanding the Lora Adaptive Data Rate*. Camarillo, CA, USA: Semtec Corporation, 2019.

[5] Bor, Martin & Roedig, Utz & Voigt, Thiemo & Alonso, Juan. (2016). Do LoRa Low-Power Wide-Area Networks Scale? 10.1145/2988287.2989163.

[6] Lora Alliance Technical Committee (2017) LoRaWAN 1.1 Specification. Version no. 1.1, Lora Alliance, Beaverton, CA, USA.

[7] J. de Carvalho Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic and A. L. L. Aquino, "LoRaWAN — A low power WAN protocol for Internet of Things: A review and opportunities," 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, 2017, pp. 1-6.

[8] Semtec Corporation, *LoRaWAN – Simple Rate Adaptation Recommended Algorithm: Common Class A/B/C Specification*. Camarillo, CA, USA: Semtec Corporation, 2016.

[9] M. Slabicki, G. Premsankar and M. Di Francesco, "Adaptive configuration of lora networks for dense IoT deployments," NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, 2018, pp. 1-9, doi: 10.1109/NOMS.2018.8406255.

[10] Kim, Dae-Young & Kim, Seokhoon & Hassan, Houcine & Park, Jong. (2017). Adaptive Data Rate Control in Low Power Wide Area Networks for Long Range IoT Services. Journal of Computational Science. 22. 10.1016/j.jocs.2017.04.014.

[11] Kim, Sungryul & Yoo, Younghwan. (2018). Contention-Aware Adaptive Data Rate for Throughput Optimization in LoRaWAN. Sensors. 18. 1716. 10.3390/s18061716.

[12] N. BENKAHLA, H. TOUNSI, Y. SONG and M. FRIKHA, "Enhanced ADR for LoRaWAN networks with mobility," 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 2019, pp. 1-6, doi: 10.1109/IWCMC.2019.8766738.

[13] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini and P. Pisani, "EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations," *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Rome, 2017, pp. 1-8, doi: 10.1109/WiMOB.2017.8115779.

[14] Abdelfadeel, Khaled & Cionca, Victor & Pesch, Dirk. (2018). Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN. 10.1109/WoWMoM.2018.8449737.

[15] B. Reynders, W. Meert and S. Pollin, "Power and spreading factor control in low power wide area networks," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6, doi: 10.1109/ICC.2017.7996380.

[16] V. Hauser and T. Hégr, "Proposal of Adaptive Data Rate Algorithm for LoRaWAN-Based Infrastructure," 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Prague, 2017, pp. 85-90, doi: 10.1109/FiCloud.2017.47.

[17] Bianchi, Giuseppe & Cuomo, F. & Garlisi, Domenico & Tinnirello, Ilenia. (2019). Sequential Waterfilling for Adaptive Data Rate allocation in LoraWAN.

[18] Abdelfadeel, Khaled & Cionca, Victor & Pesch, Dirk. (2018). A Fair Adaptive Data Rate Algorithm for LoRaWAN.

[19] S. Siraj, A. K. Gupta, and R. Badgujar, "Network Simulation Tool Survey," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 1, no. 4, June 2012.

[20] O. Georgiou and U. Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?" in IEEE Wireless Communications Letters, vol. 6, no. 2, pp. 162-165, April 2017, doi: 10.1109/LWC.2016.2647247.