COMMA: A new LoRaWAN communication optimisation mechanism for mobility adaptation of IoT

Alexis Picard1 | Jean-Christophe Lapayre2 | Fouad Hanna3 | Ronnie Muthada Pottayya1

1Maincare Corporate, Canéjan, France
2CNRS FEMTO-ST Institute—UBFC, Besançon, France
3WorldPlas Company, Besançon, France

Abstract
Communication protocols for the Internet of Things (IoT) have been optimised to meet the specific requirements of this fast-growing domain. They have been adapted mainly to optimise power consumption, maximise coverage area and facilitate their implementation. One of these protocols is LoRaWAN, which uses the ADR algorithm (Adaptive Data Rate) and aims to reduce power consumption by reducing data rate and transmission power. The ADR algorithm has been specifically optimised for static systems, that is, systems where devices do not move or have limited mobility. In this article, we introduce COMMA (Communication Optimisation Mechanism for Mobility Adaptation), an adaptive data rate mechanism optimised for everyday mobility of devices and people. We use COMMA in telemedicine, our area of expertise, to develop a medical remote-monitoring platform for mobile patients suffering from chronic diseases.

1 | INTRODUCTION

The Internet of Things (IoT) is a worldwide infrastructure of interoperable dynamic networks that use standard communication protocols [1]. Physical and virtual objects in the IoT have the following characteristics:

- Objects have identities, physical attributes and virtual personalities
- Objects use intelligent interfaces
- Objects integrate seamlessly into the information network

The IoT can be applied to many fields: transport [2, 3], banks [4], environment [5], public services [3, 6], health [7], smart cities [8], surveillance [9] and agriculture [10]. Many applications are based on wireless sensor networks where electric power (device autonomy and battery lifetime) is the most important element. Thus, reducing device power consumption is a key challenge for IoT networks. Communication protocols currently used for the classic Internet are not suitable for the Internet of Things (IoT) networks due to several reasons: excessive power consumption, significant network resource requirements, size of headers, memory requirements and implementation constraints. Therefore, new communication protocols have been introduced for IoT networks. The characteristics of these protocols have been adapted for IoT requirements with regard to power consumption, coverage and implementation (set-up) constraints. The introduction of Low-Power-Wide-Area-Network (LPWAN) was a major event in IoT evolution. These networks provide a wide coverage area and low power consumption.

We find SigFox, Ingenu, DASH7 and LoRaWAN among these protocols. To minimise power consumption and interferences, LoRaWAN uses an adaptive data rate algorithm. However, the adaptive rate algorithm used by LoRaWAN is not suitable for mobile devices. In this study, we propose a new adaptive data rate algorithm that is optimised for everyday mobility of devices and people.

For this study, our work focuses on monitoring mobile patients. In telemedicine, this is called medical remote monitoring. It allows health professionals to receive and interpret remotely the data necessary for patient monitoring [11, 12]. In this context, we worked with Maincare Solutions, which is a French leading company specialised in the field of eHealth and telemedicine.

This type of remote medical monitoring is needed particularly for the elderly and patients suffering from chronic diseases. Recently, we noticed that it can also be used in the event of pandemics, such as the one seen with COVID-19. The use of remote medical monitoring can offer solutions to important public-health and healthcare organisation issues, such as bed
availability in hospitals and the lack of medical centres in a given region. Remote medical monitoring can improve medical monitoring, provide medical care at home, prepare hospitalisation of patients, reduce patient travel to medical centres and improve and facilitate coordination between medical staff.

ADR is an algorithm (Adaptive Data Rate) that aims to optimize power consumption by reducing data rate and transmission power. However, the algorithm is optimized for static systems, that is, systems where devices do not move or have limited mobility. This is not adapted, as medical tele-monitoring is concerned on active and mobile patients. In this article, we aim to enhance the LoRaWAN ADR algorithm with our novel COMMA algorithm. The mechanism is adapted to daily mobility, such as active patient monitoring, to reduce energy consumption and extend battery lifetime.

2 | OVERVIEW OF LoRaWAN AND ADR ALGORITHM

2.1 | IoT networks

LPWAN (Low-Power Wide-Area-Network) [13] offers new wireless technologies (radio waves) designed specifically for IoT low-bandwidth applications. Three major characteristics of these new networks are: wide-range communication (coverage of several tens of kilometres, whereas a GSM network has a coverage of 1–2 km), low network cost and very low power consumption (when using hibernation, power consumption can be divided by 10, compared to Wi-Fi or 3G). All LPWAN technologies use the same set of network components when exchanging messages:

- End devices, sensors and actuators. These are our connected devices, for instance.
- One or more remote gateways (also called base stations). These are the antennas set up to cover a zone or area.
- A back-end server to process messages coming from the antennas. It manages the devices (the connected objects), security, duplicated data (when several antennas receive the same message) and the transmission of downlink messages to the devices. For example, downlinks are used to tell a device that a measurement has been received (acknowledgement), to send reconfiguration information, or to transmit a command when the device is an actuator.
- Data servers specific to applications, such as the telemedicine servers that store medical files for patients who are monitored and that offer remote monitoring services. Usually, the data travels through the core network to the application servers through HTTP requests.

LPWANs can be divided into two categories:

- The networks using licensed bands correspond to the cellular networks for the IoT, such as LTE-M, NB-IoT and EC-GSM-IoT.
- The networks that use ISM bands, known as licence-free.

The latter has allowed implementation at a reduced cost and the arrival of operators on the market like Sigfox. These frequency bands are ISM (reserved for the Industrial, Scientific and Medical domains) and can be used free of charge provided they respect the maximum channel occupation time (duty cycle) and the maximum transmission power.

For example, for the 868 MHz frequency band, it is possible to communicate at 36 s per hour at most with a transmission power of 25 mW. However, some channels have more or fewer duty cycles and more or less maximum transmission power to improve the reliability of messages or to communicate only locally, for instance.

In this study, we focus on LoRaWAN, which is a telecommunication protocol that allows low-speed radio communication of low-power objects that communicate using LoRa technology.

2.2 | LoRaWAN

LoRaWAN is an LPWAN [14]. It is a cost-effective and less-expensive alternative to cellular networks for transmitting small chunks of data.

The protocol connects low-power devices and remote gateways connected to the Internet.

Various tests have shown that it can work in a mobility context [15]. The LoRaWAN protocol uses the proprietary chrip spread spectrum modulation called LoRa. LoRa, for long range, is the name of the radio frequency physical layer. The chirp protocol uses the fixed-amplitude frequency modulation on the 868 MHz ISM radio bands in Europe and 915 MHz in North America and allows a distance between remote devices and gateways of up to 5 km in urban areas and 15 km in rural ones.

By default, the operation is based on the ALOHA type. The equipment sends data without checking if the channel it uses is available. It retransmits the message after a random time if it has been lost. Although several LoRa transmissions sent simultaneously can be processed by a single remote gateway, the ALOHA access inevitably leads to the presence of transmission collisions and, therefore, loss of messages. But a Listen Before Talk (LBT) procedure can be used to guarantee a measured RSSI (Received Signal Strength Indication) at less than ~90 dBm before any data transmission.

LoRaWAN also relies on the LoRaMAC protocol, which defines the interaction between devices and remote gateways. This protocol proposes several interesting mechanisms, such as time synchronisation of the devices and management of identification keys of the devices, as well as adaptation management of transmission parameters.

Different classes (A, B and C) are defined to qualify the equipment used for sending/receiving. Class A is the equipment that can send data at will, but the network can only send data to it during a short window after an uplink. Class C is the most power-consuming equipment that is always listening. Class B is the equipment that sends messages periodically to the server and then opens a reception window.
Spectrum spreading is a technique that consists of transmitting the message many times over different frequencies to be noise resistant. LoRa uses a very large bandwidth to operate with a very small SNR (signal-to-noise ratio). LoRa uses three bandwidths: 125, 250 and 500 kHz. The spreading factor (SF), defined between 7 and 12, represents the duration of the chirp in a simplified way. The use of a higher spreading factor for LoRa increases time on air (ToA). It takes a longer time to send the same amount of data. In fact, the use of a larger spreading factor increases the modulation period, which increases the range, to the detriment of the data rate (bandwidth), which will be reduced. Therefore, longer communication implies higher power consumption, thus reduced autonomy of devices.

### 2.3 Optimization of data rate in LoRaWAN

In this context, the ADR algorithm is a mechanism for optimising the bit rate or the power of transmission, which makes it possible to save power, save transmission time and reduce interference, among other things [16, 17]. The protocol is based on the fact that LoRaWAN allows devices to modify transmission parameters individually: spreading factor (SF), bandwidth (BW), code rate (CR), transmission power (TX) and data rate (DR). Thus, the algorithm makes it possible to determine the minimum communication parameters between the device and the gateway through a history. The ADR can be activated when a terminal has sufficiently stable RF radio conditions, and it can, therefore, be activated generally for static devices. ADR must not be activated if the network is saturated, because the devices will modify their SF and saturate the network even more (bottleneck effect) [18].

When the adaptive data rate mode of the device is enabled, the data rate used is dictated by the server through a dedicated MAC ADR command, documented in the LoRaWAN specification. The 868 MHz frequency band available for the European Union is defined from 863 to 870 MHz. On this band, eight data rates (DR) are defined by the LoRaWAN locale specification [19]. The data-rate values from 0 to 55 correspond to the LoRa modulation with SF 12 to 7, respectively, using a bandwidth of 125 kHz. The other DRs are not used in the ADR algorithm. They use a wider bandwidth, such as DR6, or simply cannot be supported in ADR, such as DR 7. The first table of DRs is defined below. The larger the DR, the smaller the spreading factor used, which will result in a smaller range but an increase in physical throughput. Therefore, the communication time will be reduced, and the power consumption will be reduced as well. In addition, if devices send data with less range and power in a dense network, there will inevitably be less interference between them. As a result, the quality of service will be better.

In addition, the Regional Parameters specification (RPS) defines six levels of output power (TX): 0 to 5 corresponding, respectively, to 20, 14, 11, 8, 5 and 2 dBm. To comply with the regulation [20], ADR is limited to the transmission power levels 1 to 5. Table 1 summarises what is meant by a DR change in scope, rate and energy consumption (always using the same frequency band here: 125 kHz—and using fixed TX).

Thus, the concept of the ADR algorithm is the adjustment of the DR and the TX to optimise saving power and to minimise interference [16, 17]. To adjust these parameters, the algorithm stores the most recent 20 uplinks (messages from device to gateway). These measurements contain the frame counter, the SNR and the number of gateways received by each uplink. For each of these measurements, the SNR of the best gateway is retained and the ‘margin’ is calculated. This is the difference between the best SNR in the history and the SNR required to retrieve a message based on the data rate. This margin is used to determine how much the data rate can be increased and whether the transmission power can be increased or decreased. However, the algorithm will never decrease the DR (on the server), because it could have a domino effect on the network. In addition, devices must implement an automatic decrease in the data rate. If the network is disconnected, they will increase their TX power level and reduce their DR data rate step by step until they are reconnected.

However, the ADR algorithm cannot be used in continuously moving systems. The faster the mobility, the lower the efficiency of ADR [21]. It can even be a factor of excessive power consumption while degrading the reliability of transmissions. The reason is that an unstable environment does not make it possible to calculate the optimal communication parameters (DR and TX) because the conditions are highly variable and the ADR algorithm does not manage travel predictions. It does not allow consideration in all cases of the degradation of the signal due to the mobility of the devices or the presence of mobile obstacles. In addition, LoRaWAN technology is not suitable for continuous data flow due to its design and duty-cycle limitation on its 868 MHz ISM frequency band (between approximately six to 20 messages/hour depending on the Time on Air (TOA)). The ADR algorithm does not allow quick modification of the optimal parameters, because it is done incrementally until communication is re-established by using a history of the last messages for each change. If the new environmental conditions are very different from the previous, it can take a long time. For example, after

| DR | Type | SF | Bandwidth | Real data flow |
|----|------|----|-----------|----------------|
| 0  | LoRa| SF12 | 125 kHz  | 250 bit/s      |
| 1  | LoRa| SF11 | 125 kHz  | 440 bit/s      |
| 2  | LoRa| SF10 | 125 kHz  | 980 bit/s      |
| 3  | LoRa| SF9  | 125 kHz  | 1 760 bit/s    |
| 4  | LoRa| SF8  | 125 kHz  | 3 125 bit/s    |
| 5  | LoRa| SF7  | 125 kHz  | 5 470 bit/s    |
| 6  | LoRa| SF7  | 250 kHz  | 11 000 bit/s   |
| 7  | FSCK|      | 500 kHz  | 50 000 bit/s   |

Abbreviations: DR, data rate; SF, spreading factor.
an environmental change and with a message every 10 min, the parameters would be optimal again after 20 new messages, or more than 3 h. Therefore, the adaptation of the communication parameters is too slow, and the ADR algorithm cannot be used during movement.

Benkahla et al. have proposed the E-ADR algorithm for this issue [22]. It is based on the estimation of the trajectory for a mobile device and its position at the next packet transmission. The goal is to reduce power consumption, the time the channel is busy and the loss rate due to the ADR adaptation time and the limits of the duty cycle. E-ADR uses the trilateration method with a LoRa packet to find the geolocation of the device. The server estimates the predicted position of the device and calculates the RSSI (Received Signal Strength Indication) to assign a set of communication parameters to use according to a correspondence table. The estimation of the position in this context must be very precise and requires the use of a personal local network, the size of a farm, for instance.

2.4 | Energy consumption model

To calculate the energy consumed when sending a packet [23], we must first calculate the ToA (in ms), \( T_{\text{Packet}} \). This is the time before a receiver receives a signal sent by a transmitter. It is represented as the total of the preamble time and the payload time: the amount of time the radio should send preamble bits (for synchronization) and payload bits (for data).

\[
T_{\text{Packet}} = T_{\text{Preamble}} + T_{\text{Payload}}
\]

the preamble duration \( T_{\text{Preamble}} \) (in ms) is calculated as

\[
T_{\text{Preamble}} = (4.25 + NP) \times T_{\text{Symbol}}
\]

with \( NP \) is the programmed preamble symbol number and \( T_{\text{Symbol}} \) the symbol period. The payload duration \( T_{\text{Payload}} \) (in ms) is

\[
T_{\text{Payload}} = N_{\text{Payload}} \times T_{\text{Symbol}}
\]

with \( N_{\text{Payload}} \) the number of payload symbols and \( T_{\text{Symbol}} \) the symbol period. The symbol period \( T_{\text{Symbol}} \) is

\[
T_{\text{Symbol}} = \frac{2^{SF}}{BW}
\]

with \( BW \) the bandwidth (125 or 250kHz in Europe) and \( SF \) the spreading factor. The payload symbol is the number of symbols in payload and packet header. It is calculated as

\[
N_{\text{Payload}} = 8 + \max \left( \text{ceil} \left( \frac{\Theta(PL, SF)}{\Gamma(SF)} \right) \times \frac{1}{CR} \right)
\]

with ceil representing the ceiling function (rounded upper); \( PL \) the payload size (in bytes); \( \Theta(PL, SF) = 8PL - 4SF + 16 + 28 - 20H \) symbols, with \( H = 1 \) if the header is present, and \( H = 0 \) otherwise; \( \Gamma(SF) = SF - 2DE \) symbols, with \( DE = 1 \) if the low data-rate optimisation is active and \( DE = 0 \) otherwise. The coding rate is the proportion of transmitted bits that actually carry information. It is modelled with the equation \( CR = \frac{n}{8} \) with \( n \) between 1 and 4, which represents the 5, 6, 7 or 8 bits used for coding, respectively.

We can now calculate the energy consumed per transmission (in \( \mu J \)) using the following equation:

\[
E_{\text{Cons}} = P_{\text{Cons}}(P_T)^n T_{\text{Packet}}
\]

with \( P_{\text{Cons}}(P_T) \) the total energy consumed depending on the transmission energy, and \( T_{\text{Packet}} \) the ToA. The total energy consumed is calculated using the following equation:

\[
P_{\text{Cons}}(P_T) = T_T x \times SV
\]

3 | CONTRIBUTION: THE NEW PROTOCOL COMMA

As we have seen, the ADR algorithm, as defined in the literature, allows a significant saving of power in static systems with a relatively stable environment.

However, as part of the daily use by an active person, we noticed that mobility is relative; individuals move regularly but spend most of their time in static and known environments. For example, the average person will spend most of their time at home and then at work. Therefore, the use of the ADR algorithm in daily mobility is possible. Nevertheless, an adaptation at each living space every day leads to a loss of energy when we transmit packets in over-optimised conditions and a risk of losing the packet and, therefore, of retransmission and overconsumption, the time that the communication parameters are increased to reach the remote gateway. In this case, if the device is expecting to receive an ACK, it can implement a retransmission procedure all the same, with a decrease of the Spreading Factor to find connectivity once again.

The work on E-ADR [22] allows improvement of the adaptation of the communication parameters within the framework of continuous mobility based on the predictions. But this does not correspond precisely to our requirements for daily mobility.
In this article, we propose a quick adaptation mechanism for communication parameters in the case of daily mobility, with the sudden changes in environments caused by travel and the stabilisation times of the parameters in known or unknown locations. It will operate along with the ADR to stabilise the parameters at each location. The objective will be to further reduce energy consumption and extend the life of the batteries.

### 3.1 The concept of the algorithm

The solution we present is based on an important condition, which is daily mobility. As a patient travels throughout their day, the following minimal conditions must be found in each place:

- A static environment, that is, conditions must be relatively stable to use the ADR algorithm normally.
- A known environment, that is, the individual has already used the algorithm in that place, or they have remained static long enough to have a communication history (20 records are recommended). Each known environment is defined by an area, represented by an average position in space (a geolocation) and a limited radius.
- The device must be listening to downlink messages after each transmission, which corresponds to Class A, B and C. It is therefore preferable to use the least power-consuming class as much as possible, which is Class A.

The originality of the proposed solution is based on the use of different communication histories of areas to find a faster adaptation of optimal communication parameters. To achieve this fast adaptation according to the environment where the device is located, the following steps are performed:

1. Preservation of communication histories for the different environments frequented by the device.
2. Detection of change of environment or travel of the device, which would require the fast adaptation of communication parameters.
3. Association of an environment with one of the saved communication histories, or definition of a new area and a new history.

We detail these issues in the following sections. Figure 1 represents the schema of our protocol.

### 3.2 Detection of an environment change

The detection of a change of environment occurs at both the server level and the device level. When the device does not receive the confirmation message from the server, this means that the upload message or the confirmation message (ACK) has never arrived. This can occur in the case of interference or if the device is in suboptimal sending conditions. For example, when moving the device away from the nearest remote gateway, it is a change of environment. In this case, the device can then modify its transmission parameters step by step (TX and then DR) until finding connectivity. It is important for the device not to be offline for too long a time only in the case of a connected health device and a potential emergency. Retransmissions are done with a latency and can only be executed every minute or so. This means it is important for us to use a mechanism to quickly find connectivity with the server. Therefore, we will use the maximum power and minimum DR to maximise chances for a message to reach a remote gateway. Sending a message with suboptimal parameters will probably make it possible for it to reach several remote gateways and, therefore, possible for the server to localise the device and associate an environment.

Then again, we have also detected changes to the environment when the environment improves for the device. An example is when it comes close to the gateway. This time, detection is not done on the side of the device, which is still able to send messages and receive the ACKs, but on the side of the server instead. If the difference between the SNRs of new and old messages is too high, we can deduce that there has been a change in the environment. We will take $SNR_{diff}$ as the difference between the minimum SNR and the maximum SNR over the last 30 min. If $SNR_{diff} > 5$, we consider this to be a broken environment. In this case, the server will send a geolocation request in the next confirmation message. This will then result in an attempt to associate the environment.

### 3.3 Environmental association

The association function of known environments can be implemented in different ways. We chose to use geolocation as the main criterion.

Currently, GPS receivers associated with a GSM network are used the most to geolocate a device. There are currently
affordable GPS receivers on the market, but the main problem is power consumption and battery life. The power consumption of a GPS receiver is approximately 30 to 50 mA, which represents a considerable energy requirement for most low-consumption IoT devices. For example, a LoRaWAN module operating in an 868 MHz frequency band consumes 2.8 mA in an activated state, 38.9 mA for outgoing data and 14.2 mA for incoming data [25].

A solution without GPS using an accelerometer and a compass also exists [26]. However, the error cumulates over time, and we cannot distinguish the same location visited at two different moments.

There are several techniques to estimate the geographic position of a device. Each of these techniques has different characteristics and requirements. The three most commonly used methods for geolocation are triangulation, trilateration and multilateration. In the absence of time synchronisation between gateways and devices, the only method that can be used with LoRaWAN is multilateration. The location is calculated with the difference in TDoA (Time Difference of Arrival) arrival time [27, 28] on the remote gateways. Therefore, a minimum of three remote gateways synchronised temporally between them and attainable by the device is required.

The environment association task for the device will be implemented on the server, which keeps the communication history on different encountered areas. Each area will be defined on the server by a single dot \( \text{AREA}_{\text{POS}} \), which will be the average of the last 20 positions of the messages associated with it. To associate a geolocation with an area, we define \( \text{RADIUS} \), which is the maximum distance between the newly calculated point and the average point of the area to make an association.

If the distance between the two points is greater than \( \text{RADIUS} \), the new point will form a new area. It has been shown [27] that the LoRaWAN geolocation is not really accurate. For stationary devices, the average reported accuracy was about 100 m reaching four remote gateways. Therefore, geolocalisation with LoRa is not very precise, but it is sufficient to distinguish the different locations (home, work, etc.). Consequently, it will be necessary to take a large enough \( \text{RADIUS} \).

When an already existing area is found after an environment change detection, the best DR and TX pair on the last 20 communications will be sent to the device as the new setting to use. Without further environmental change, the ADR algorithm will be used again to vary the communication parameters in the static environment.

In the case where it is not possible to reach three remote gateways, geolocation cannot be done. We then apply the classic ADR algorithm to stabilise the communication parameters step by step.

Algorithm 2 shows the procedures to be implemented on the server with the detection of changes in the environment, the association of an existing environment and the request for geolocation. Algorithm 1 shows the client-side management.

The function \( \text{ReceiveRequest} \) is used when the server receives a request from an object. It saves the SNR in the history and detects if there is a break in the environment. In this case, it asks the object to send a geolocation request. Otherwise, it launches the classic ADR algorithm which will adjust the parameters \( TX \) and \( DR \).

The \( \text{ReceiveGeolocRequest} \) function is used when the object sends a geolocation request to the server. It will associate (with \( \text{FindArea} \) function) the position of the object with an already known area if possible, or create a new one otherwise. The best TX and DR parameters found for the area will then be sent to the object, which will use them during the next sending round.

\[
\text{Algorithm 1 Device Algorithm}
\]

1: \( TX_{MAX} \leftarrow 20 \)
2: \( DR_{MIN} \leftarrow 0 \)
3:
4: \( \text{procedure} \ \text{receiveGeolocRequest}(\text{message}) \)
5: \( \quad TX \leftarrow TX_{MAX} \)
6: \( \quad DR \leftarrow DR_{MIN} \)
7: \( \quad \text{send empty request} \)
8:
9: \( \text{procedure} \ \text{adjustParameters}(TX, DR) \)
10: \( \quad TX \leftarrow TX \)
11: \( \quad DR \leftarrow DR \)
12:
13: \( \text{procedure} \ \text{loopDataRequest} \)
14: \( \quad \text{send data request} \)

\[
\text{Algorithm 2 Server Algorithm}
\]

1: \( TX_{MAX} \leftarrow 14 \)
2: \( DR_{MIN} \leftarrow 0 \)
3: \( \text{RADIUS} \leftarrow 100 \)
4: \( \text{SNR}_{\text{MAXDIFF}} \leftarrow 5 \)
5: \( \text{last}_{\text{SNR}}, \text{areasHistories}[], \text{currentArea} \)
6: 7: \( \text{procedure} \ \text{receiveRequest}(\text{msg}) \)
8: \( \quad \text{if last}_{\text{SNR}} \neq \text{UNDEFINED then} > \text{new area} \)
9: \( \quad \text{last}_{\text{SNR}} \leftarrow \text{msg}.\text{SNR} \)
10: \( \quad \text{else} \)
11: \( \quad \text{SNR}_{\text{diff}} \leftarrow |\text{msg}.\text{SNR} - \text{last}_{\text{SNR}}| \)
12: \( \quad \text{last}_{\text{SNR}} \leftarrow \text{msg}.\text{SNR} \)
13: \( \quad \text{environment change detection} \)
14: \( \quad \text{if SNR}_{\text{diff}} > \text{SNR}_{\text{MAXDIFF}} \text{then} \)
15: \( \quad \quad \text{send geolocation request to device} \)
16: \( \quad \text{else} \)
17: \( \quad \quad \text{currentArea}.\text{addHistory}(\text{msg}.\text{TX}, \text{msg}.\text{DR}) \)
18: \( \quad \quad \text{apply ADR algorithm with currentArea history} \)
19: \( \quad \quad \text{send new parameters TX and DR to device} \)
20: 21: \( \text{procedure} \ \text{ReceiveGeolocRequest}(\text{msg}) \)
22: \( \quad \text{currentArea} \leftarrow \text{findArea}(\text{msg}) \)
23: \( \quad \text{currentArea}.\text{adjustPosition}(\text{msg}) \)
To compare data rate adaptation algorithms, we performed tests of the ADR algorithm described by Semtech [16, 17] with and without the use of the COMMA mechanism for daily mobility conditions (home and work) over 3 days. All stakeholders (device, network, and server) have been described in the IPHEN platform [29]: we used the LoRaWAN network of the operator Objenious, which covers the entire French territory with many remote gateways. Objenious would take care of geolocalisation TDoA with the help of machine learning to improve precision. We did not have any additional information because Objenious is a private operator and it does not want to communicate on this subject. The device was an Arduino UNO with a Dragino LoRa Shield module that had a Semtech SX1276 chip.

Messages were sent every 10 min to meet the duty cycle of the 868 MHz band. Table 4 gives the approximate distances between the device and the remote gateways at different times of the day. It must be taken into account that fast trips were made between the various locations of the device during the 3 test days.

Figure 2 shows variations of the SNR over time using the ADR algorithm without COMMA mechanism. The SNR was calculated by the back-end server (Objenious). The remote gateways measured the connection between the signal and the noise when receiving a message. The peaks of SNR correspond to breaks in the environment (sudden changes) at different times of the day. As the device gets closer to the nearest gateway, the signal it receives is more intense, and its SNR increases rapidly. By applying the ADR decreasing algorithm of transmission parameters (visible in Figure 3), the SNR tends to go back down for a certain time until a pseudo-stabilisation of the parameters occurs.

When the device moves away from the nearest gateway, the SNR decreases, and the peak is inverted after the application of the algorithm. With a message every 10 minutes, the ADR algorithm without the COMMA mechanism must wait for 20 messages, almost 3 h, to begin to adapt the transmission settings. For example, in this figure, we can see the change in environment at 8:00 over 3 days; the device moved closer to the remote gateway and the peak SNR was reached quickly at about 10:00. The transmission settings were adjusted incrementally on the DR rate first, and then on the TX transmission rate (when the DR is at maximum). This adjustment, which can be seen in Figure 3, took several hours.

We can also see that when a device moves away from the remote gateway (when the person leaves work in our example), the DR parameter decreases. This signifies that the device has lost its connectivity, and the device itself decreased this parameter to find it again. Therefore, several messages will be sent several times, requiring greater energy consumption.

In the end, when the device moves closer to the gateway, the algorithm spends a large part of the day performing adjustments to the communication settings and is, therefore, over-consuming power even though very little travel has happened. The COMMA mechanism allows faster adjustment of communication settings and quickly reduces the peak SNRs. As a result, this reduces the processing latency and saves power while maintaining a similar quality of service.

Figure 4 shows the variations of DR and TX using the COMMA mechanism. We can see the detection of breaks in the environment, visible in the figure at 8:05 and 18:30 on the first day and 8:25 and 18:30 on the second day. At 8:05 on the first day, the environment was unknown to the device: the DR was reset to 0 and the TX to 20 dBm. Conversely, at 18:00, the environment was known; the previous DR and TX was reused and the adaptation of the sending parameters was done quickly.

### 4.2 Power comparison

To compare the ADR algorithm with and without the use of the COMMA mechanism, we must evaluate the energy consumption that happens when sending.

For all of the transmissions we carried out, only the TX transmission power and the Data Rate DR varied ($SV, CR, PL, NP, BW, H$ and $DE$ are always the same in our tests). Here is
FIGURE 2 SNR variation without COMMA

FIGURE 3 Data rate and TX power variations without COMMA

FIGURE 4 Data rate and TX power variations with COMMA
an example with $DR = 0$ (either $SF = 12$ based on Table 1) and $TX = 11$ (or $TRx = 32 \text{ mA}$ based on Table 3):

We calculate the power $P_{\text{Cons}}(P_T)$ according to the transmit power $TX$:

$$P_{\text{Cons}}(P_T) = TRx \times SV \times 32 \times 3V = 105.6mW;$$

where $3, 3V$ is the consumption of the SX1276

$$T_{\text{Symbol}} = \frac{2^{SF}}{BW} = \frac{2^{12}}{125kHz} = 32,768 \text{ ms}$$

We calculate $N_{\text{payload}}$ the number of symbols used to transmit the payload using $\Theta(PL, SF)$ and $\Gamma(SF)$:

$$\Theta(PL, SF) = 8PL - 4SF + 16 + 28 - 20H = 60 \text{ symbols}$$

$$\Gamma(SF) = SF - 2DE = 12 \text{ symbols}$$

$$N_{\text{payload}} = 8 + \max\left\{ \left\lceil \frac{\Theta(PL, SF)}{\Gamma(SF)} \right\rceil \times \frac{1}{CR} \right\} = 13 \text{ symbols}$$

with $CR = 1, H = 0$ (no header), $PL = 8$ bytes and $DE = 0$ (no low data rate optimization).

We calculate the payload duration $T_{\text{Payload}}$ and the preamble duration $T_{\text{Preamble}}$:

$$T_{\text{Payload}} = N_{\text{payload}} \times T_{\text{Symbol}} = 425,984 \text{ ms}$$

$$T_{\text{Preamble}} = (4.25 + NP) \times T_{\text{Symbol}} = 335,872 \text{ ms};$$

with $NP = 6$ Symbols

Finally, we calculate the ToA $T_{\text{Packet}}$ and the energy consumption $E_{\text{Cons}}$ (depending on the transmission power):

$$T_{\text{Packet}} = T_{\text{Preamble}} + T_{\text{Payload}} = 761,856 \text{ ms}$$

$$E_{\text{Cons}} = P_{\text{Cons}}(P_T) \times T_{\text{Packet}} = 80451,994 \mu J \approx 0,0805J$$

The energy consumed was approximately $0.0805J$ for $DR = 0$ and $TX = 11$. We reproduced this formula for all of the packets sent from the devices over the 3 days of tests to calculate the cumulated Time on Air and produce the graph 5 of the cumulated energy consumption during transmissions.

We observed that the total energy consumed during transmission, is lower over the last 2 days of tests with the mechanism for rapid adaptation in daily movement, COMMA. Over the last 2 days, the cumulative energy is $3.49J$ without the COMMA mechanism and $2.47J$ with the COMMA mechanism, a gain of approximately $41.3\%$. Per day, this gives a total charge of $1.058C$ without COMMA and $0.748C$ with COMMA (dividing by $3.3V$ voltage and the number of days).

### 4.3 Discussion

The COMMA mechanism could be an interesting solution in the case of daily mobility. However, it has certain technical limitations that do not allow it to be used all the time.

First, the number of remote gateways that can be reached by the device must be at least three, which is currently rare in rural areas. The detection of the environment is also not optimal when geolocation is used, because it is not precise and can be erroneous [27]. In this case, a server-side learning algorithm could improve the detection and association of the environment using patterns of travel of moving people. A prediction of travel could be done.

In addition, environmental conditions change in the same place over time. For example, a Doppler effect can happen

| Hour   | Remote gateway distance |
|--------|-------------------------|
| 8:00-12:00 | ≈1000 m         |
| 12:00-14:00 | ≈2000 m         |
| 14:00-18:00 | ≈1000 m         |
| 18:00-8:00  | ≈5000 m         |

**TABLE 4 Distance to the remote gateway during the test days**

**FIGURE 5 Cumulative energy with and without COMMA**
when a device is moving around the devices. This does not allow certainty of an uplink with previously used emission parameters. A margin of error can then be used to avoid suboptimal environmental conditions. However, if this happens, a degraded procedure could be used. The device could use its own send parameters and not temporarily follow the instructions of the algorithm sending by the server.

Finally, the algorithm only works with acknowledgements for each sent message. This will consume a significant amount of power and create additional interferences. The solution might be to make grouped acknowledgements. However, an instantaneous adaptation during a break in the environment would no longer be possible Figure 5.

5 CONCLUSION

One of the most important advantages of the LoRaWAN protocol, and several other IoT protocols, is its low power consumption. This advantage has led researchers and developers to explore many new-use cases. In this study, we showed that the ADR algorithm used with LoRaWAN can further reduce power consumption and interference by adapting its transmission power and data rate. The main challenge of this study was the use of the Adaptive Data Rate algorithm in cases when patients are mobile and continue their daily activities. This is particularly important for patients suffering from chronic diseases, such as children with asthma, congenital heart disease, diabetes, hyperactivity disorder or attention deficit. We showed that using an adaptive data rate algorithm is possible under some conditions and this has led us to the better understand and analysis of many issues in this context. Our first results are very encouraging and show that it is possible to design systems using this adaptive data rate algorithm to save power consumption. Further work will be done on the robustness of the algorithm. For instance, collisions between data packets or GPS unavailability.

ACKNOWLEDGEMENTS

The authors wish to thank the CNRS, the University of Bourgogne/Franche-Comté, Maincare Solutions Company and WorldPlas Company for their support.

They have contributed to the development of our work, which combines academic and industrial skills for the benefit of eHealth and telemedicine.

ORCID

Alexis Picard © https://orcid.org/0000-0003-0253-1406

REFERENCES

1. Brito, J.M.C.: Trends in wireless communications towards 5g networks—the influence of e-health and IoT applications. In: 2016 International Multidisciplinary Conference on Computer and Energy Science (Split-Tech), pp. 1–7 (July 2016)
2. Kirk, R.: Cars of the future: the internet of things in the automotive industry. Netw. Secur. 5(5) 16–18 (2015)
3. Zhang, M., Yu, T., Zhai, G.F.: Smart transport system based on the internet of things. Appl. Mech. Mater. 48-49, 1073–1076 (2011)
4. Hao, Z., et al.: A secure fast payment method for edge-IoT platforms using blockchain. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC) (2018)
5. Bovenzi G., et al.: IoT-enabled distributed detection of a nuclear radioactive source via generalized score tests. In: Communications in Computer and Information Science, pp. 77–91. Springer Singapore (2019)
6. Joshi, R., Jyothiradana, A., Kumary, I.S.: Iot based school bus tracking and arrival time prediction. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (2017)
7. Swaroop, K.N., et al.: A health monitoring system for vital signs using IoT. Internet of Things. 5, 116–129 (2019)
8. Liu, Y., et al.: Intelligent edge computing for IoT-based energy management in smart cities. IEEE Netw. 33(2), 111–117 (2019)
9. Ciouzon, D., et al.: Decision fusion rules in ambient backscatter wireless sensor networks. 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (2019)
10. Khattab, A., Abdelgawad, A., Yelmarthi, K.: Design and implementation of a cloud-based IoT scheme for precision agriculture. 2016 28th International Conference on Microelectronics (ICM) (2016)
11. Freed, J. et al.: Telemedicine: is it really worth it? A perspective from evidence and experience. BMJ Health Care Infor. 25(1), 14–18 (2018)
12. Woottton, R.: Recent advances: Telemedicine. BMJ. 323(7311), 557–560 (2001)
13. Dali, I., Mahbubur, R., Abusayeed, S.: Low-power wide-area networks: Opportunities, challenges, and directions. ICDCN Workshops (2018)
14. Vangelista, L., Zanella, A., Zorzi, M.: Long-range IoT technologies: The dawn of LoRa™. In: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 51–58. Springer International Publishing (2015)
15. Alvear, O., et al.: Assessing the impact of mobility on LoRa communications. In: Interoperability, Safety and Security in IoT, pp. 75–81. Springer International Publishing (2018)
16. Corporation, S.: Lorawan—simple rate adaptation recommended algorithm. Lora Netw. Rate Adapta. (2016)
17. Hauser, V., Hegr, T.: Proposal of adaptive data rate algorithm for LoRaWAN-based infrastructure. 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud) (2017)
18. Haxhibeqiri, J., et al.: A survey of LoRaWAN for IoT: From technology to application. Sensors. 18(11), 3995 (2018)
19. Alliance, L.: Lorawan Regional Parameters, pp. 2020–21 (2016). https://lorawan-alliance.org/sites/default/files/2018-05/lorawanRegionalParameters_v1.0
20. ERC ERC Recommendation 70-03 Relating to the Use of short Range Devices, pp. 2020–27 (2017). https://docdb.cept.org/download/25e41779-ed6c/Rec7003ce.pdf
21. Kousias K., et al.: Empirical analysis of LoRaWAN adaptive data rate for mobile internet of things applications. In: Proceedings of the 2019 Wireless of the Students, by the Students, and for the Students Workshop - S3 19. ACM Press, (2019)
22. Benkahla N., et al.: Enhanced ADR for LoRaWAN networks with mobility. 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) (2019)
23. Bouguerra, T., et al.: Energy consumption model for sensor nodes based on LoRa and LoRaWAN. Sensors. 18(7), 2104 (2018)
24. Semtech: Sx1272 lora calculator, pp. 2020–27 (2015). https://sx1272-lora-calculator.software.informer.com/download/
25. Libelium, Wasmote Lorawan networking guide. Comunic. Distrib. S.L. 56 (2016), p.
26. Zhu, X., et al.: Accurate outdoor pedestrian tracking with smartphones. 2013 Proceedings IEEE INFOCOM. (2013)
27. Podevijn, N., et al.: TDoA-based outdoor positioning with tracking algorithm in a public LoRa network. Wirel. Commun. Mobile Comput. 2018, 1–9 (2018)

28. Bakkali, W., et al.: Kalman filter-based localization for internet of things LoRaWAN™ end point. 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC) (2017)

29. Picard, A., et al.: I2phen: A novel interoperable IoT platform for medical telemonitoring. Coop. Design Visual. Eng. 84–97 (2019)

How to cite this article: Picard A, Lapayre J-C, Hanna F, Muthada Pottayya R. COMMA: A new LoRaWAN communication optimisation mechanism for mobility adaptation of IoT. IET Wirel. Sens. Syst. 2021;11:120–130. https://doi.org/10.1049/wss2.12017