Adaptive Data Rate Techniques for Energy Constrained Ad Hoc LoRa Networks

Derek Heeger
Sandia National Labs
Albuquerque, NM

Maeve Garigan
Roper Solutions, Inc.
Las Cruces, NM

Jim Plusquellic
University of New Mexico
Albuquerque, NM

***Abstract*— Long Range (LoRa) is an emerging low-power wide-area network technology. LoRa messages can be transmitted with a variety of parameters including transmit power, spreading factor, bandwidth, and error coding rates. While adaptive data rate (ADR) capabilities exist in the LoRa wide-area network (LoRaWAN) specification, this work is motivated by a cattle monitoring application where LoRaWAN is not feasible. In this scenario, the mobility of the animal changes the optimal parameter selections, which are the settings that transmit the data with the lowest energy consumption. This work analyzes ADR techniques to most efficiently find the optimal data rate for a firmware update, although the techniques are still valid for any large data exchange. It extends the ADR to use frequency shift keying (FSK) when there is enough signal strength since Semtech LoRa integrated circuits support FSK mode. The work uses dynamic acknowledgements and timeout values to improve the convergence time. The paper experimentally validates an analytical transmit time model and then describes three different methods for accomplishing the adaptive data rate. The methods are modeled analytically for the different convergence settings and two are demonstrated using the Microchip SAMR34 Explained boards.**

***Index Terms*—LoRa, Adaptive Data Rate, FSK**

I. Introduction

IoT devices are becoming more prevalent in our everyday lives. One of the challenging aspects of IoT device design is the energy consumption associated with the communication mechanism which has caused a demand for low-power wide area network (LPWAN) technologies. A few key technologies have emerged to meet this need including LoRaWAN, Sigfox, Narrow Band IoT, and LTE-M [1].

LoRaWAN is a wide area network technology that is optimized for low data rate communications in the unlicensed spectrum bands. It is designed to connect directly to a LoRaWAN gateway that can route relevant data to the internet. There are three implementations of LoRaWAN that optimize for various power consumption scenarios. LoRaWAN uses LoRa for the physical layer; a closed source protocol that uses chirped spread spectrum modulation to achieve greater noise immunity at the expense of slower data rates. The LoRa communication protocol can be configured using a variety of settings including transmit power, spreading factor (SF), error coding rates, use of CRC, header types, and bandwidth.

This work is motivated by a cattle monitoring application which uses a battery-powered sensor to collect health information that is transmitted using LoRa. None of the LoRaWAN operating modes are sufficiently optimized for this application so the application currently uses the LoRa physical layer with a custom network. The choice to exclude LoRaWAN is driven by the need to achieve extremely low power operation, to communicate synchronously over a mesh topology, and to conduct firmware updates.

The techniques proposed in this paper are designed to optimize the remote firmware update time but they are relevant for any large data exchange. From Table 1, firmware updates using LoRa vary from 16 minutes to 8.36 hours depending on the SF selected. The times reported are based on the following assumptions: 1) no packet transmission errors occur, 2) the bandwidth is 125 kHz, 3) a 12-symbol preamble is used, 4) CRC is enabled, 5) an implicit header is used, and 6) the error coding rate is 4/5. Note that update times are shorter as SF is reduced, making them more attractive, but the communication range is also reduced. Therefore, the ADR optimization process should target the lowest SF that provides an acceptable communication bit error rate. LoRa ICs from Semtech can be configured to use frequency shift keying (FSK) which has higher data rates at the expense of less range. We investigate extending ADR to use FSK mode when there is appropriate signal strength, which, in turn, improves data rates as shown along the bottom of Table 1.

TABLE 1. *THE NOMINAL TIME IT TAKES TO UPDATE A 128kB FIRMWARE IMAGE.*

	Time to Update Firmware for a 128kB image			
LORA SF	***6***	***8***	***10***	***12***
Time	16 minutes	46 minutes	2.49 hours	8.36 hours
FSK (bps)	**19,200**	**57,600**	**115,200**	**300,000**
Time	53.3 s	17.7 s	8.88 s	3.41 s

This work makes the following contributions:

- We extend ADR to use FSK in addition to optimizing LoRa settings.
- We propose an error recovery process that is appropriate for this ad hoc scenario which addresses the possibility that two devices can reside on different settings if there is a communication error. LoRaWAN gateways avoid this issue because they receive packets from all LoRa SF settings simultaneously.

- We incorporate dynamic timeouts and acknowledgements that depend on the communication settings which have not been investigated in previous LoRa ADR work.
- The ADR techniques are simulated and verified experimentally, going beyond most previous work that provides only simulation results.

Section 2 provides a detailed background of LoRa, LoRaWAN, and the related research. Section 3 experimentally validates the LoRa transmit time model. Section 4 describes the rate adaptation techniques in detail and shows modeling results. Section 5 describes the experimental validation of the techniques and Section 6 concludes the paper.

II. Background

The LoRa physical layer has many configurable parameters including SF, error coding rates, header types, preamble length, and bandwidth. SF is the number of chirps per meaningful symbol of information. The number of chirps improves the noise tolerance of the system but decreases the communication rate and can be configured to a value in the range of 64 to 4096. The error coding rate is the number of redundant bits transferred to allow for error correction and is configurable to include between 25% to 100% extra symbols. There is also an optional header that can be used to transmit the packet size and error coding rate prior to the data payload. For applications where this value is fixed, the header can be omitted to improve efficiency. The preamble length is used for synchronization between the transmitter and receiver and can be set to a value between 6 and 65535 symbols. The bandwidth can be decreased to increase noise immunity at the expense of data rate.

LoRaWAN is the network layer that uses LoRa to transmit data. It is defined as a standard for interfacing to a LoRa gateway. There are three methods of operation using LoRaWAN and they trade off performance for power consumption. Class A is the lowest power operation and all communication is initiated by the end device. There are two short windows within which the end device can receive data after it transmits if the gateway needs to respond. Class B is like class A except it periodically opens receive windows for data reception. Class C is the highest power consumption mode because the end device is always able to receive data. None of these LoRaWAN options are optimal for the cattle monitoring application, so we use a custom network with LoRa as the physical layer instead.

There have been a significant number of recent papers on LoRa and LoRaWAN due to its promising range and power performance. For example, the authors of [2] provide a broad analysis on LoRa parameter selection, while the authors of [3] investigate the scalability of LoRaWAN. An energy consumption model for different transmit parameters is developed and analyzed in [4]. The energy consumption of LoRaWAN is modeled for battery powered applications in [5]. There have been a variety of publications on applications that demonstrate performance improvements enabled by LoRa [6-8].

Most of the previous work on LoRa rate adaptation focuses on minimizing bit errors for a congested IoT environment using LoRaWAN. The authors of [9] develop a simulation framework and propose an optimized ADR technique with LoraWAN gateways. In [10,11], the emphasis is on implementing ADR to minimize collision probability and on maximizing data rates in congested networks. A probing algorithm to negotiate parameters that minimize bit errors and energy consumption for LoRaWAN gateways is proposed in [2]. The authors of [12] investigate LoRaWAN's ADR convergence time and propose optimization techniques for some of the tuning parameters. It has been shown that LoRaWAN's ADR performs well for stationary objects but can be improved for mobile devices [13]. In [14], the authors show that adding hysteresis to the current LoRaWAN ADR protocol can improve it.

III. LoRa Transmit Time Model

This work focuses on optimizing the total transmit time so it is important to utilize accurate packet transmission time models in order to properly evaluate the adaptive rate techniques. This section uses the model described in [15] and compares the modeling results to experimentally measured values.

A. Equations Used in Time Modeling

The transmit time can be partitioned into preamble and payload transmission times. The preamble time is defined by Eq. 1 where $N_{preamble}$ is the number of preamble symbols.

$$T_{preamble} = (N_{preamble} + 4.25)T_s \tag{1}$$

The payload time is defined by multiplying the number of payload bytes by the symbol time as shown in Eq. 2.

$$T_{payload} = n_{payload}\, T_{s}. \tag{2}$$

The symbol time is found by using the spreading factor and bandwidth as shown in Eq. 3.

$$T_s = \frac{2^{SF}}{BW}. \tag{3}$$

The number of bytes in a payload is found using Eq. 4 (a full definition of this expression can be found in [15]). Note that $n_{payload}$ cannot be less than 8.

$$n_{payload} = 8 + \mathrm{ceil}\left[\frac{8PL-4SF+28+16CRC-20IH}{4(SF-2DE)}\right](CR+4) \tag{4}$$

B. Experimental Test

The accuracy of these equations is important for understanding the error bounds of our modeling, so the timing was experimentally measured using the SARAR34 Explained board. An automated test was generated to loop through a variety of packet transmission parameters. A timer in the microcontroller was started before each packet transmission and was then terminated with an interrupt generated upon a successful completion of the packet transmission. The results of the analytical and measured times are shown in Figure 1. The x-axis varies the payload size in bytes while the y-axis shows the transmit time in seconds. The solid curve indicates the modeled time and the circles indicate the measured time. The error for all packet transmissions remains under 3% for all messages with a payload over 50 bytes but increases to 13% for 4-byte packet transmissions. This demonstrates that the payload length parameter is accurately modeled by Eq. 4 but the error is larger for the packet header, preamble, and CRC. We acknowledge that our measurement scheme is not ideal and includes two sources of error; namely in the oscillator tolerance and in the additional clock cycles required for starting and stopping the timer. The oscillator tolerance is 3% and the timer processing adds less than 10 clock cycles (~40 us).

This experiment was repeated using several LoRa configuration options and the results were found to be similar. In one case, we included the message header and in a second experiment, we modified the preamble length. Although the error is undesirable, the experimental results provide sufficient accuracy to define the error bounds for the ADR negotiation modeling described in the next section.

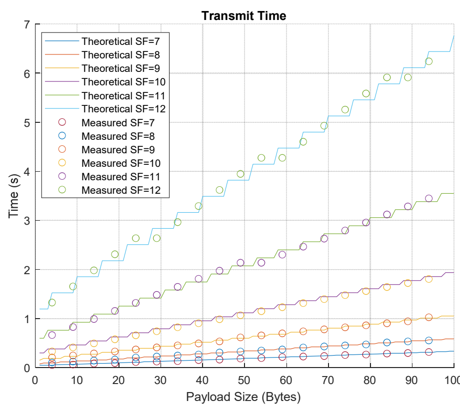

Fig 1. The experimental validation of the transmit time calculations.

IV. Adaptive Data Rate Techniques

A. Technique Overview

We build an ADR LoRa network with two devices and investigate a dedicated exchange scenario to determine the optimal rate. The device that already has the firmware update in memory will be referred to as the master and the device receiving the update is referred to as the end device (ED). Note that this is an ad hoc network so both devices are energy constrained and the devices are limited to receiving on a single channel unlike a LoRaWAN gateway.

We focus on optimizing the parameters that have the largest impact, namely the LoRa spreading factors and FSK bit rates. Further optimization can be achieved by varying the bandwidth and coding rates. The goal of this work is to define an algorithm that minimizes the amount of time required to converge to the optimal rate given a set of constraints on the possible settings. However, the investigated techniques are valid when the number of settings is expanded for cases in which the application can benefit from the increased resolution. The settings used for this experiment are given in Table 2.

We implement our communication testing protocol with the following 6 commands:

1. Go to ADR mode
2. Decrease data rate one setting
3. Go to a specific setting
4. Exit ADR
5. Ping
6. Acknowledge receipt of a packet.

Commands are transmitted with the format given in Figure 2, which consists of a preamble, a command, 3 arguments, and a CRC.

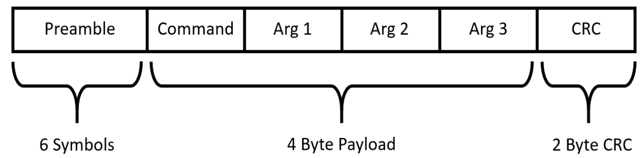

Fig 2. The communication packet structure.

All tested algorithms carry out the following sequence of steps:

1. Initiate an exchange on the lowest data rate setting to establish communication.
2. The master sends the command to the ED to go to an improved data rate setting. The value of this setting depends on the algorithm being used.
3. Upon receiving the command, the ED will automatically reconfigure itself to that setting and acknowledge at the higher data rate.
4. If the devices are unable to communicate at a lower setting, the master will initiate the error recover process.
5. If both devices time out, they return to the setting they last successfully communicated on.

An important feature of these techniques is the use of dynamic timeout values to minimize time spent waiting for a response during an unsuccessful packet transmission. The timeout values are given for each setting in Table 2. The timeout values account for the transmit, execution, and waiting times for the SX1276 RF module to change modes. If an acknowledgement is not received by the specified timeout, the master will make three additional attempts before giving up. It will then ping the device at the faster setting to see if an acknowledgement message was missed. If all four of these attempts fail, it will try to communicate on a setting with a slower data rate (higher setting).

TABLE 2. *The Settings and Timeout Values*

ADR Settings and Timeouts				
Setting Number	**Modulation Type**	**Setting Config**	**Master Timeout**	**ED Timeout**
0	FSK	300 kbps	0.1 s	0.4 s
1	FSK	200 kbps	0.1 s	0.4 s
2	FSK	115.2 kbps	0.1 s	0.4 s
3	FSK	57.6 kbps	0.1 s	0.4 s
4	FSK	19.2 kbps	0.1 s	0.4 s
5	FSK	9.6 kbps	0.1 s	0.4 s
6	LoRa	SF=6	0.1 s	0.4 s
7	LoRa	SF=7	0.2 s	0.8 s
8	LoRa	SF=8	0.25 s	1 s
9	LoRa	SF=9	0.4 s	1.6 s
10	LoRa	SF=10	0.7 s	2.8 s
11	LoRa	SF=11	1 s	4 s

12	LoRa	SF=12	3 s	12 s

B. Incremental Search

In this section, we investigate an incremental search algorithm to find the lowest setting that enables communication. Incremental search is the simplest method to implement but can cause longer convergence times. This method starts with setting 12 with ADR mode enabled, and then it incrementally decreases the setting until the communications are lost. Upon failure, the master will try the error recovery process. The algorithm terminates when the setting reaches 0 or the error recovery process fails.

The estimated convergence time for each setting is shown in Figure 3. The x-axis gives the setting that the device is converging to while the y-axis gives the time. For example, the total convergence time to setting 4 (FSK, 19.2kbps) is just over 8 seconds. The plot is initially counterintuitive because converging to the higher settings has the slowest convergence times despite having the fewest number of packets exchanged. This is true because the error recovery process executes at the slowest communication rates. Note that the sum of the master and ED transmit times does not equal the total time because there are timeouts and processing delays involved with the packet transmissions. Moreover, the transmit and convergence times do not account for any unexpected packet communication errors so these results represent the best-case convergence times.

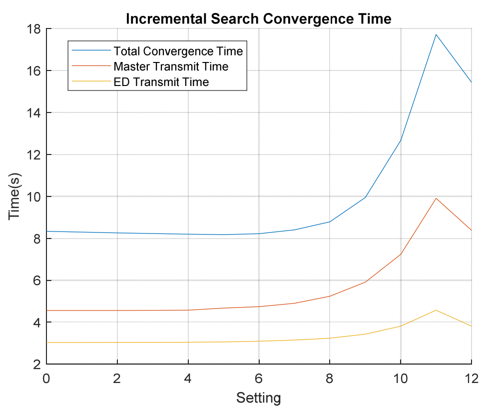

Fig 3. The convergence time for the incremental search.

Some applications do not require the range that setting 12 provides, so it could be advantageous to start at a lower setting. Figure 4 depicts the convergence time assuming the algorithm starts at a lower setting. For example, the purple plot shows the convergence times to switch to any of the lower settings if the algorithm begins with setting 10. The overall time savings given by the difference between the curves for setting 11 and 12 is substantial; however, decreasing the starting setting below 11 has diminishing returns. The danger associated with starting with a low setting is that successful communication may never be achieved.

These results show that incremental search is effective at quickly finding the minimum rate for higher settings but is time-consuming for the lower FSK settings. This is true because every setting involving LoRa is used before it reaches the FSK modes. Sections C and D will investigate techniques that pass through the higher settings more efficiently.

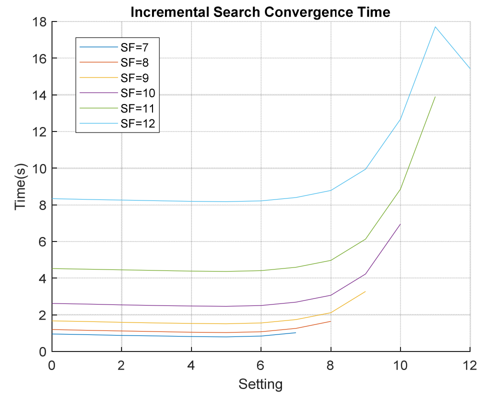

Fig 4. The effect on convergence time starting at different LoRa settings.

C. Binary Search

In this section, we investigate a binary search algorithm as an alternative strategy to reduce the convergence time for the lower settings. The binary search process chooses the next setting after a successful iteration using Eq 5. The $S_{current}$ variable is the current communication setting and $S_{highest,failed}$ is highest setting the algorithm has failed at starting with a value of 0.

$$S_{next} = S_{current} - ceil\left(\frac{S_{current} - S_{highest,failed}}{2}\right) \quad (5)$$

The binary search convergence time is depicted in Figure 5. The x-axis gives the setting number and the y-axis shows the convergence time. The total time assumes there are no communication errors until the attempted setting is lower than the optimal setting. Converging to settings 0 through 6 is faster with this technique because there is only a single exchange at the highest setting. The primary issue with the binary search algorithm is that it can go through the communication failure process multiple times. For example, converging to setting 12 causes four failures to occur with significant timeout periods resulting in the largest convergence time as shown in the figure.

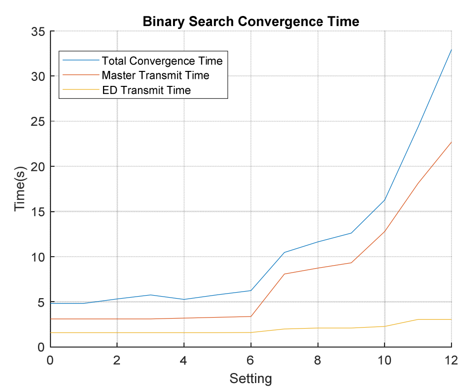

Fig 5. The convergence time for the binary search.

We then investigated the impact of the starting search setting on the convergence time, and the results are shown in Figure 6. The starting setting has a greater impact on binary search than was true for incremental search.

Binary search is superior to incremental search in cases where the devices are in close proximity to each other causing the search to converge to lower settings. However, the performance is poor for high settings, making the incremental search superior in cases involving long-range communication.

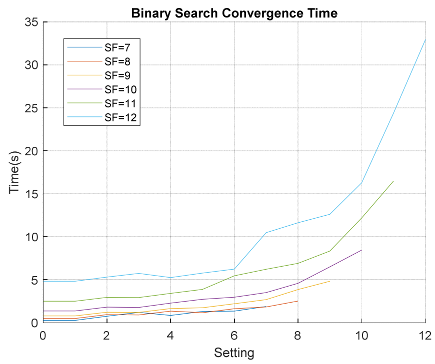

Fig 6. The effect on convergence time starting at a different LoRa setting.

D. Greedy Search Using RSSI and SNR

In this section, we investigate a greedy search algorithm to improve the search performance. Greedy search uses RF receiver data to intelligently judge the next appropriate setting. This can be the received signal strength indicator (RSSI) or the signal-to-noise ratio (SNR). In theory, this is the optimal technique because it converges after a few short exchanges.

The first step is to establish communications with the ED. The master will use the information gained from the received packet to estimate the RSSI and SNR. With this data, it will intelligently map the next communication setting. If the mapping is correct, the device will acknowledge the receipt of the packet at the next setting and the ADR process will terminate.

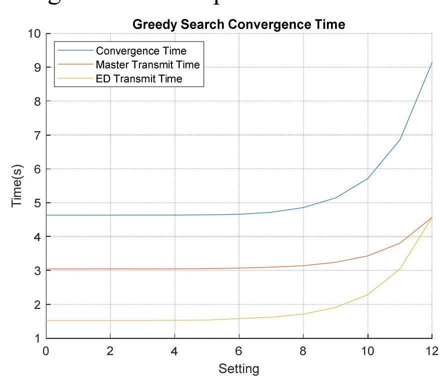

Fig 7. The convergence time for the greedy search.

Figure 7 shows the ideal convergence time for the greedy search This method is substantially better than the others. The worst-case convergence is 9 seconds but the majority of the cases converge in under 5 seconds. While this method is superior in theory, it is much harder to implement in practice due to measurement noise.

E. Effect of the Acknowledgement Setting

In this section, we investigate timing differences between the two methods of ED acknowledgement. This acknowledgement is used by the master to validate that the ED has properly changed communication settings and can be transmitted in one of two ways:

1. Receive the change setting command, send the acknowledgment, and then drop to the decreased rate state
2. Receive the change setting command, drop to the decreased rate state, and then send the acknowledgement

Option 2 is faster because the master spends significantly less time receiving the acknowledgement. Even if the ED fails to respond and the error recovery process is invoked, the timeout value is lower for the next setting so the convergence will remain shorter. The effect of faster acknowledgement on convergence time can be seen in Figure 8 for the incremental search and Figure 9 for the binary search. The curves shows the acknowledgement at the lower setting and the circles show the acknowledgement at the same setting. Both search methods are take substantially longer when the ED acknowledgement is made at the same setting (Option 1).

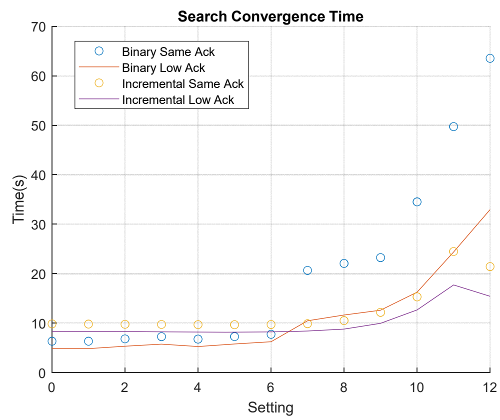

Fig 8. The timing difference using different acknowledgment rates.

V. Experimental Results

A. Experimental Setup

The algorithms were developed and tested on SAMR34 Explained development boards created by Microchip (Figure 9). The SAMR34 contains a Semtech SX1276 LoRa module and an ARM Cortex M0+ MCU integrated onto a 6mm x 6mm BGA package. It is tiny and easily configurable to achieve low power operation, making it well suited for IoT applications. The SAMR34 Explained board has a programming interface, an antenna, and the RF circuitry needed to evaluate the SAMR34.

The SAMR34 includes a USB peripheral that is used for communicating with the device to control and view the status of the board while ADR is running. The firmware is developed using Atmel Studio and the data is logged using Tera Term.

The LoRa ICs are configured with the settings:

- Bandwidth of 62.5 kHz
- 6-symbol preamble
- CRC is enabled
- Implicit header is being used
- Error coding rate of 4/5

The algorithms implement both a master and receive mode. The master binary and incremental search modes are initiated by typing a command through Tera Term. The ED mode is initiated by receiving a 'Go to ADR command' from a master. The ED receive mode function requires two timeouts:

1. A short-term timeout to return to the state that was last successfully communicated on
2. A long-term fail-safe timer to return to the starting setting if the ADR series is not properly terminated

B. Results

The algorithms were developed and tested. A timer was used in the master processor to measure the total convergence time during operations. In order to simulate long range transmissions, the antenna was removed from the board. The boards were still able to communicate but at a significantly reduced range that allowed for easy collection of data in a confined space.

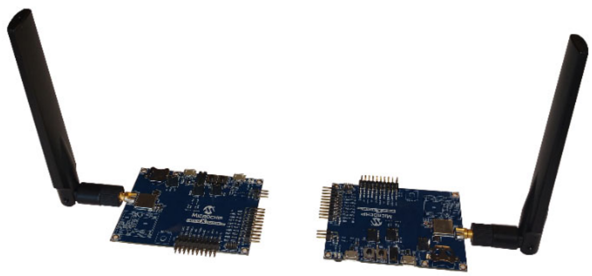

Fig 9. The SAMR34 explained boards

One important observation was that most communication errors came from the master missing the acknowledgement instead of the ED missing the command to change settings. This leaves the system in a state where the master is commanding the ED to go to a setting that it is already at. This motivated the final step of the error recovery process where the master drops to the next setting and pings the ED.

There were more packet transmission errors for communications than expected even if the settings were far above the optimal. This made the experimental data typically converge to a higher setting than was modeled. If the error occurred when the settings were 10-12, the experimental results were typically much higher than the model.

VI. Conclusion

A LoRa timing model was validated experimentally and then applied to analyze ADR techniques for ad hoc networks. The ADR technique extended LoRa to also use FSK when possible. Three different algorithms were investigated including an incremental, binary, and greedy search. The greedy search is superior but is the most challenging to implement due to noise in the RSSI and SNR of the system. Binary search is ideal when converging to high data rate settings but is slow to converge to the high SF settings. Incremental search outperforms binary search for high settings but is consistently worse at converging to the FSK settings. Both search techniques were implemented experimentally in an attempt to validate the models. Future work on improving the rate involves implementing the RSSI/SNR intelligent technique. Additional work involves implementing the bootloader and applying an ADR technique during the firmware update.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly-owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525

References

[1] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment", ICT Exp., vol. 5, no. 1, pp. 1-7, Mar. 2019.

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

[3] M. C. Bor, U. Roedig, T. Voigt, J. M. Alonso, "Do lora low-power wide-area networks scale?", MSWiM, 2016.

[4] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN," Sensors, vol. 18, no. 7, p. 2104, Jun. 2018

[5] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the Energy Performance of LoRaWAN," Sensors, vol. 17, no. 10, p. 2364, Oct. 2017

[6] V. Sharma, I. You, G. Pau, M. Collotta, J. Lim, and J. Kim, "LoRaWAN-Based Energy-Efficient Surveillance by Drones for Intelligent Transportation Systems," Energies, vol. 11, no. 3, p. 573, Mar. 2018

[7] L. Germani, V. Mecarelli, G. Baruffa, L. Rugini, and F. Frescura, "An IoT Architecture for Continuous Livestock Monitoring Using LoRa LPWAN," Electronics, vol. 8, no. 12, p. 1435, Dec. 2019

[8] R. P. Centelles, F. Freitag, R. Meseguer, L. Navarro, S. F. Ochoa, and R. M. Santos, "A LoRa-Based Communication System for Coordinated Response in an Earthquake Aftermath," Proceedings, vol. 31, no. 1, p. 73, Nov. 2019

[9] M. Slabicki, G. Premsankar, M. D. Francesco, "Adaptive configuration of lora networks for dense iot deployments", IEEE/IFIP Network Operations and Management Symposium (NOMS), pp. 1-9, April 2018

[10] S. Kim and Y. Yoo, "Contention-Aware Adaptive Data Rate for Throughput Optimization in LoRaWAN," Sensors, vol. 18, no. 6, p. 1716, May 2018.

[11] K. Abdelfadeel, V. Cionca, D. Pesch, "Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN", 2018 IEEE World of Wireless Mobile and Multimedia Networks (WoWMoM), June 2018

[12] S. Li, U. Raza, A. Khan, "How Agile is the Adaptive Data Rate Mechanism of LoRaWAN?", 2018 IEEE Global Communications Conference (GLOBECOM), pp. 206-212, 2018.

[13] K. Kousias, G. Caso, O. Alay, F. Lemic, "Empirical Analysis of LoRaWAN Adaptive Data Rate for Mobile Internet of Things Applications", 2019 MobiCom, October 2019

[14] V. Hauser, T. Hegr, "Proposal of Adaptive Data Rate Algorithm for LoRaWAN-based Infrastructure", 2017 IEEE 5th International Conference on Future Internet of Things and Cloud, 2017

[15] Semtech, "SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver", SX1276/77/78/79 datasheet, August 2016