Optimization of LoRa Devices Communication for Applications in Healthcare

Alexander Valach and Dominik Macko
Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
Email:xvalacha@stuba.sk, dominik.macko@stuba.sk

***Abstract*—Wireless devices in the Internet of Things (IoT) face a communication parameters selection problem to avoid collisions due to inability to listen before a transmission, limited power supply, or duty-cycle restrictions. Another problem arises in densely populated areas, where license-free band is jammed with many different technologies. Therefore, a network controller does not have exact information about channel congestion and has to determine it for each node from observations. Among the most promising solutions for long-distance and low-power IoT networks is LoRa. In this paper, we propose a modified LoRa architecture using energy-wise LoRa@FIIT protocol to optimize communication parameters selection and ensure QoS for application in healthcare devices, where critical information must be properly delivered and acknowledged.**

***Keywords*—healthcare; Internet of Things; LoRa; low power**

I. Introduction

IoT end devices are usually small embedded computers that can be wearable. They have less computational power, less memory capacity, limited power supply and less bandwidth than conventional IP and Ethernet network devices. Those factors should be considered during protocol or network architecture design. Another constraint of communication in license-free bands is a duty-cycle. It is defined as a maximum percentage of time during which an end-device can transmit on a selected channel [1] (1% in Europe). To meet the industrial requirements, modern IoT networks should be scalable and automatically respond to network changes, e.g. congestion, duty-cycle limitations, interference from other technologies and interference within the same technology when two or more devices transmit at the same time using the same communication parameters (CP), e.g. spreading factor (SF) and power.

IoT devices have potential utilization in healthcare. The typical scenario assumes a patient wearing a battery-supplied device that would measure a blood pressure, a heart rate, or an oxygen saturation. Based on collected data, recommendations would be given, or early disease detection might be possible, e.g. detection of heart arrhythmia using heart rate long-term measurements.

This work was partly supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic (Incentives for Research and Development, Grant No.: 2018/14427:1-26C0), and the Eset Research Centre.

One of the toughest challenges in LoRa networks is a collision-free communication in densely populated areas [2], [3]. A lot of work has been done in the field of computing an optimal channel selection in the network server or end nodes [4]. Specifically, in the problem of end nodes energy optimization [5], [6] a channel selection [7], a collision-free communication in densely populated areas [8], or an optimization of communication in general [3]. In [9], a re-inforcement learning (RL) proved to be a huge advantage even if non-stationary settings are present in the environment. However, only the research in [10] has taken into account a dissemination, i.e. a process of distributing gained knowledge from the network server to the end nodes. All gateways are bound to duty-cycle limitations, since it is the key constraint in LoRa networks. For a real-world use case with hundreds of nodes, a proper way and time to disseminate a network-wise model to end nodes is an important requirement. Authors of [2] proposed a usage of Markov Decision Process to find optimal communication parameters from the end node perspective.

To the authors' best knowledge, most works used computer simulation only and did not really consider an overhead needed for LoRa gateways to update CP on end devices. Except for [10], none of the above mentioned have been dealing with duty-cycle limitations and an efficient way to transfer a control plane data from a network server to an end device or a mathematical formula to find a proper timing when to update end node (EN) configuration. This paper is focused on a proposal of a reliable LoRa network architecture using LoRa@FIIT and STIoT protocol [11] with ensured quality of service (QoS) using an efficient distribution of adaptive communication parameters.

II. A Design of Scalable and Reliable IoT Network

A typical LoRa architecture consists of several end nodes (EN), wireless access points (AP), also called gateways, and a central point of management called a network server (NS). End nodes are usually embedded devices with ability to measure or evaluate certain characteristics and send them via LoRa technology. LoRa frames are received by single or multiple APs in the node's surrounding. A NS controls network traffic and is responsible for a duty-cycle computation, communica-

This is an accepted version of the paper:
A. Valach and D. Macko, "Optimization of LoRa devices communication for applications in healthcare," in 2020 43rd International Conference on Telecommunications and Signal Processing (TSP), 2020, pp. 511-514.
doi: 10.1109/TSP49548.2020.9163432
URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9163432

tion parameters selection, and a most adequate AP selection in case a message requires an acknowledgement and had been received by multiple APs. For tasks like device addressing, encryption, and a message acknowledgment, a link layer protocol is required. The most popular and widely spread protocol is LoRaWAN [12]. LoRaWAN encrypts messages using AES-128 algorithm, which was not designed to be used in IoT devices and can be difficult to implement on memory and energy constrained nodes. To support a number of features, a LoRaWAN header is quite complex. For 1 B of useful information a device sends 29 B overhead [1], [12]. The protocol was also not designed with QoS in mind and is impractical to send an emergency message. LoRaWAN networks use Adaptive Data Rate (ADR) to update communication parameters of end nodes if required [12]. The ADR algorithm simply compares the average RSSI value (computed from N latest measurements) with a predefined threshold [11], [12]. If the value is below or above this threshold, a downlink message is scheduled and sent to the end node during the next uplink message.

A. *Overcoming Drawbacks of LoRaWAN Protocol Design*

LoRa@FIIT is a link layer protocol created to overcome drawbacks of LoRaWAN protocol stack. It was not designed to support roaming, so we need to make sure that owner of the network is also owner of the devices, which can be considered a disadvantage when compared to LoRaWAN. However, not all use cases require use of roaming and LoRa@FIIT was designed to provide more efficient communication for such scenarios [11].

On the other hand, LoRa@FIIT has some advantages over LoRaWAN:

1) It uses optional acknowledgments for sent messages; thus, it is not mandatory to open a receiving window for an EN after message sending.
2) It uses XXTEA algorithm for encryption, which needs smaller blocks (64-bit alignment) and was designed for IoT devices with memory constraints in mind.
3) It offers more energy-efficient communication using shorter headers compared to LoRaWAN, since for 1 B of payload there is only 12 B of overhead. This leads to approximately 42% less battery and duty-cycle usage.
4) It uses sequence numbers to achieve reliable delivery. Also the NS can utilize this numbers to evaluate a pseudo-link quality for a specific device.
5) It has a built-in mechanism for QoS support, called emergency message, which must be acknowledged and is transmitted using the maximum power.

LoRa@FIIT differentiates several message types. There are register messages sent during initial EN registration process, hello messages which serve as a connection keepalive and health check mechanism, already mentioned emergency messages used for critical information transfer, and data messages used for regular data transfer. Register and emergency messages are sent using full transmission power to ensure a successful delivery.

LoRa@FIIT networks do not use Message Queuing Telemetry Transport (MQTT) message broker for communication between APs and NS in comparison to known open-source LoRaWAN implementations [13]. It uses STIoT (Secured TCP for IoT) protocol [11]. Rather than using subscriber-publisher model, it focuses on reliable and secure information exchange between APs and the NS. This architecture is simple to deploy as it is client-server based.

Both LoRa@FIIT and LoRaWAN are ALOHA-based protocols and do not have any collision detection or avoidance mechanism. This responsibility is transferred from end nodes to network server or even APs [10].

When updating a selection algorithm on an end node, the energy efficiency must be considered. Many IoT devices have very simple firmware implemented and adding additional overhead, when proper SF and power must be selected and statistical model updated, could lead to higher power consumption.

B. *Effective Selection of Communication Parameters*

Current implementations of communication parameters selection (CPS) are mostly based on ADR algorithm [13]. As a research have proven, it performs not very well when number of devices is increasing [10], [14]. It also strongly depends on threshold values that has to be precalculated or updated according to results from observations. These values are also globally set in the current implementation of the LoRa@FIIT network server [15].

We propose a more modern approach suitable for a dynamic environment, which is exactly the case for LoRa devices supporting a mobility. The network server algorithm is based on Thompson Sampling with Switching Environments. The algorithm was primarily developed as a RL technique to solve a Multi-Armed Bandit Problem (MABP), well-known from a recommendation process. It performs well in case of CP selection according to performed computer simulations [16] along with Upper Confidence Bound (UCB) [9].

These algorithms are suitable for a selection of CPs (combination of SF and transmission power, where lower values are preferable) when a success of a message delivery or signal strength are not predefined. However, we cannot let the end nodes simply select these parameters. There are two main reasons for this. A statistical model, which both algorithms are based on, has to be constantly updated using RSSI, SNR or Sequence Number values, which are not known to end nodes. Also, energy consumption comes to mind when considering such a solution. All the IoT devices have less memory, computational power and have a limited power supply, and are not ready to select parameters with limited knowledge of a network state.

Another option is to let a NS maintain a statistical model for EN and AP and send it only when the environment changes or a link quality has been degraded. ENs are responsible only for CP selection based on the updated statistical model. The main problem of the proposed solution is maintaining this model on ENs. The model is easily updated on NS using data from APs

(SNR, RSSI and Sequence Number). SNR and RSSI are used to determine a channel quality and Sequence Number serves as a mechanism for early detection of link congestion (missing values) or signal loss for certain ENs.

MABP algorithm calculations are based on provided reward when a certain arm (CS selection in our case) is pressed (chosen) by a bandit (device). Due to duty-cycle limitations, it is not possible to provide immediate feedback to ENs after an uplink message has been sent. This is a reason, why we need to modify and maintain statistical model for each node and determine a moment in which it needs to be updated.

The proposed statistical model consists of spreading factor, power and a probability of successful packet delivery when using combination of both parameters. It is stored in the JSON format and could be encoded to base64 string or even BSON (Binary JSON) to minimize the size of downlink messages. When a significant change occurs in a network, a model intended for certain EN or AP has to be updated, a NS schedules a downlink message with the updated model and sends it during the next opportunity (immediately for AP, after the next uplink message for EN). During an initial device registration, a statistical model of AP that received the message is sent to the device.

C. Ensuring Quality of Service

LoRa@FIIT protocol has a built-in mechanism to ensure QoS and optional acknowledgement. There is no need for a device to open a receiving window when no acknowledgment from network server is expected. However, there are some messages that require acknowledgement. Message acknowledgement can vary depending of application needs. This is also an advantage, because non-critical systems do not need their hello messages to be acknowledged by the NS.

One example of critical communication is an emergency message. It is sent using maximum transmission power to make sure it is received by one or more nearby APs. This special type of message also ensures QoS, but only on ENs [11].

To the authors' best knowledge, there is no implementation of QoS on other network components, such as APs or NS. If the message is not prioritized on each component, the QoS is not ensured at all. Therefore, we propose a priority queue to be implemented in the wireless access point. Its aim is to handle messages from the priority queue prior to any other message type, as presented in Fig. 1.

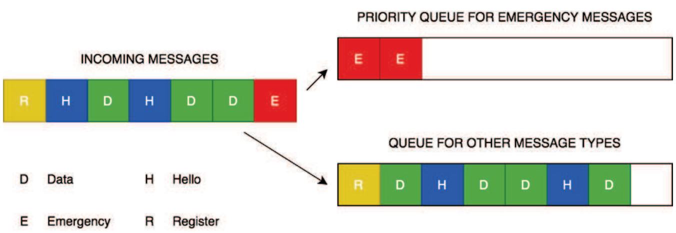

Fig. 1. Separated queues dependent on incoming messages types.

Another technique which might be considered a bad practice for QoS is the fact that a network server waits predefined time in seconds for another LoRa packets to arrive. The reason behind this is a duty-cycle limitation on APs (also called gateways or concentrators). The network server also manages a duty-cycle of APs. It waits for replicas of the received message to select an AP with the highest duty-cycle available, even if it was received only by a single AP, which is unknown to network server until the waiting timer expires. This leads to a delay, which is not tolerated in QoS ensured environment.

We assume there will be no waiting timer for the emergency queue, as it is of a high importance. If an emergency message is received by the network server, it does not wait for other packets to come. It immediately processes the message and send it to the AP from which the message originated.

III. Network Evaluation Using Load Tests

To test our proposed solution, we have designed the network architecture presented in Fig. 2. Both physical and simulated devices are present in this architecture. The devices have been divided into three groups. Groups A and B consist of simulated end nodes and access points. However, main difference between these groups is a network delay. The fact that LoRa messages could be decoded by every single access point in the nodes' range is taken into account. As a result, the network server should wait for any other duplicate messages to properly select an access point with the highest duty-cycle available. A network delay is simulated by connecting to a remote VPN server to ensure the traffic from group B will come later than packets from Group A.

We also need physical devices to compare energy efficiency with the currently implemented LoRa@FIIT library on end nodes [17]. We use the nodes based on 8 MHz ATmega328P processor, powered by 3.3 V LiSOCl2 battery. The selected power supply is preferred for long-life (several years) ENs, because it has a very low discharging rate, less than 1% per year. A RFM95W LoRa communication module is also used. To simulate a heart-rate measurement a MAX3015 particle-sensing board is used rather than heart-rate monitoring

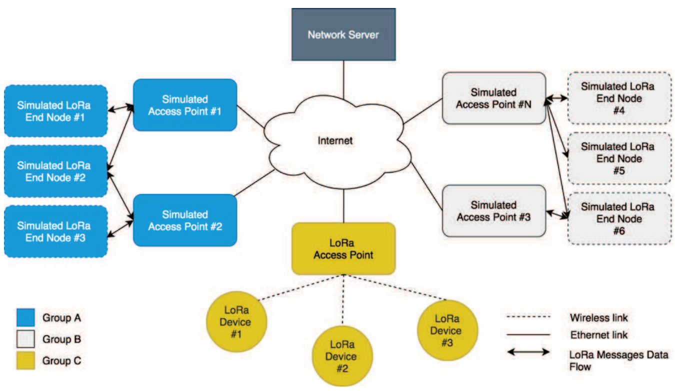

Fig. 2. Proposed architecture for evaluation of network reliability and scalability.

sensor MAX30101. However, it is sufficient to simulate the measurements and drain a battery power, for the scenario to be more realistic.

Our real-world LoRa AP in group C is based on Raspberry Pi 3 model B and iC880A concentrator and connected to the Internet. To test the scalability of the proposed solution, we use a single LoRa@FIIT NS implemented in Java. We plan to register thousands of simulated LoRa ENs and hundreds of APs to simulate the situation which might occur during this decade.

In a real-world scenario, there would be interference and packet collisions. In our simulated environment, a collision occurs on APs when two ENs transmit using the same frequency and the same SF at the same time. Interference and signal loss during a movement of patient wearing a sensor device are simulated by pseudo-randomly decreasing SNR values of received packets on APs.

At the time of writing this paper, we have successfully deployed our own LoRa@FIIT network with 10 ENs and 1 AP. We are now collecting data from stationary (not moving) nodes. One of them has a MAX30105 [18] sensor connected. Other nodes just simulate the process of heart-measuring by generating pseudo-random sensor-measured values and calculations based on the heart-rate calculation algorithm currently implemented in the library [18].

At this state, we are currently developing a console application written in Python to simulate a daily routine of ENs. The application is heavily inspired by the LoRa@FIIT library [17]. It simulates the functions of both ENs and APs.

IV. Conclusion

One of the most promising technology for IoT devices communicating over long distances with minimal power consumption is LoRa. The potential of this technology could be lost when not handling with care. An increasing number of connected devices, duty-cycle limitations, and unoptimized communication parameters selection threaten a real-world deployment with thousands of connected devices.

Based on the work of other researchers in this field, we used a network-wise statistical model for each network device maintained by a NS using information gathered from APs. ENs acquire this model during a registration process and use it as a knowledge base for optimal communication parameters selection. The model is further updated by NS when significant network changes occurs. As a channel selection based on statistical model can be energy-consuming, we intend to test the process of communication selection in a real-world scenario using ATmega328P based ENs.

For evaluation purpose, we have proposed a network architecture consisting of three groups of devices. In group A, there are only simulated devices with direct Internet connection. Group B also consists of simulated devices; however, they use slower VPN connection to provide higher latency than group A. The last one, group C, consists of real ATmega328P-based nodes powered by batteries.

In the future, we plan to implement modified Thompson Sampling and Upper Confidence Bound on the network server to compare the results with the currently implemented Adaptive Data Rate algorithm. There is also a possibility to choose not only between Spreading Factor and transmission power, but also using different frequencies. This can lead to higher required maintenance for the statistical model in compensation for higher throughput or collision-free communication.

References

[1] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Communications magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[2] R. M. Sandoval, A.-J. Garcia-Sanchez, J. Garcia-Haro, and T. M. Chen, "Optimal policy derivation for transmission duty-cycle constrained LPWAN," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3114–3125, 2018.

[3] D. Zorbas, G. Z. Papadopoulos, P. Maille, N. Montavont, and C. Douligeris, "Improving LoRa network capacity using multiple spreading factor configurations," in *2018 25th International Conference on Telecommunications (ICT)*. IEEE, 2018, pp. 516–520.

[4] M. Parvin and M. R. Meybodi, "MABRP: A multi-armed bandit problem-based energy-aware routing protocol for Wireless Sensor Network," in *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*. IEEE, 2012, pp. 464–468.

[5] N. Michelusi and M. Levorato, "Energy-based adaptive multiple access in LPWAN IoT systems with energy harvesting," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 1112–1116.

[6] J. Zhang, H. Jiang, Z. Huang, C. Chen, and H. Jiang, "Study of multi-armed bandits for energy conservation in cognitive radio sensor networks," *Sensors*, vol. 15, no. 4, pp. 9360–9387, 2015.

[7] A. Hoeller, R. D. Souza, O. L. A. López, H. Alves, M. de Noronha Neto, and G. Brante, "Analysis and performance optimization of LoRa networks with time and antenna diversity," *IEEE Access*, vol. 6, pp. 32 820–32 829, 2018.

[8] J. Pullmann and D. Macko, "A new planning-based collision-prevention mechanism in long-range IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9439–9446, 2019.

[9] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, "Multi-armed bandit learning in IoT networks: Learning helps even in non-stationary settings," in *International Conference on Cognitive Radio Oriented Wireless Networks*. Springer, 2017, pp. 173–185.

[10] R. M. Sandoval, A. Garcia-Sanchez, and J. Garcia-Haro, "Optimizing and updating LoRa communication parameters: A machine learning approach," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 884–895, 2019.

[11] O. Perešíni and T. Krajčovič, "More efficient IoT communication through LoRa network with LoRa@FIIT and STIOT protocols," in *2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, 2017, pp. 1–6.

[12] LoRa Alliance, "Whitepaper: A solution for successful interoperability with DLMS/COSEM and LoRaWAN," 2019, https://lora-alliance.org/sites/default/files/2019-11/dlms-lorawan-whitepaper_v1.pdf.

[13] (2019) Chirpstack network server. [Online]. Available: https://github.com/brocaar/chirpstack-network-server

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

[15] (2019) Lora network server. [Online]. Available: https://github.com/alexandervalach/lora-network-server

[16] R. Kerkouche, R. Alami, R. Féraud, N. Varsier, and P. Maillé, "Node-based optimization of LoRa transmissions with multi-armed bandit algorithms," in *2018 25th International Conference on Telecommunications (ICT)*. IEEE, 2018, pp. 521–526.

[17] (2017) Lora@fiit library. [Online]. Available: https://github.com/HalfDeadPie/LoRa-FIIT

[18] (2018) Max30105 particle and pulse ox sensor breakout. [Online]. Available: https://github.com/sparkfun/MAX30105_Particle_Sensor_Breakout