Genetic Algorithm For LoRa Transmission Parameter Selection

Aghiles Djoudi[12], Rafik Zitouni[2], Nawel Zangar[1] and Laurent George[1]

[1]LIGM/ESIEE Paris, 5 boulevard Descartes, Champs-sur-Marne, France
[2]ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France
Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

***Abstract*—The exponential grow of IoT applications in both industry and academic research raises many questions in wireless sensor network. Heterogeneous networks of IoT devices strongly depend on the ability of IoT devices to adapt their data transmission parameters to the application requirement. One of the most important problem of the emerging IoT networks is the limitation in terms of energy consumption and computation capability. This limitation could be addressed by using edge computing to unload IoT devices from additional computation tasks. Our work is motivated by the idea of matching each transmission configuration with a reward and cost values to satisfy applications constraints. Our goal is to make IoT devices able to select the optimal configuration and send their data to the gateway with the QoS required by IoT applications. Determining the best configuration among 6720 settings is challenging. The difficulty is mainly due to the lack of tools that could take all applications requirements into account to select the best settings. To address this problem, we use a genetic algorithm in an edge computing to select the transmission parameters needed by the application. Genetic algorithms are used in feature selection and ranking problems. In our case, each configuration represents a feature that needs to be selected to match better the QoS criteria. Particularly, we analyze the impact of selecting one configuration in 3 kinds of applications: text, voice and image transmission by modeling a new adaptive data rate selection process. We validate our approach by using both simulation and a real environment testbed.**

I. Introduction

The need of Low power wide area networks (LPWAN) increased significantly these five last years. The main factor is that IoT devices require low power consumption to transmit data in a wide area. LoRa, Sigfox and NB-IoT are the most known technologies that satisfy these requirements. Applications like smart building and smart environment are one of hundreds use cases that need to be deployed with such technologies. Unlike Sigfox and NB-IoT, LoRa is more open for academic research because the specification that governs how the network is managed is relatively open. LoRa is a wireless modulation technique that uses Chirp Spread Spectrum (Proprietary) (CSS) in combination with Pulse-Position Modulation (PPM). The transmission could be configured with 4 parameters: Spreading Factor (SF), Transmission Power (Tx), Coding Rate (CR) and Bandwidth (BW), to achieve better performance.

The main LPWAN research directions are about link optimization, adaptability and large scale networks to support massive number of devices. The selection of an appropriate transmission parameter for IoT networks typically depends on the nature of the application. Thus heterogeneous transmission configuration and SF allocation strategies need to be studied. In this paper, we investigate the performance of heterogeneous networks (i.e. when each IoT device selects its LoRa transmission parameters according to its link budget and the application requirements). For that purpose we have developed a LoRa transmission adaptation mechanism, both ns-3 simulator and the Low cost Lora Gateway [1] are used to validate our approach. The computation tasks of the selection process will run on the Gateway device (Raspberry-pi) and the required settings will be sent to nodes for the next transmission.

This paper is organized as follows. Section II elucidates summary of related works. In section III, we propose our approach to solve LoRa parameter selection problem. Our experimentation in a real environment is presented in section IV. Section V concludes this paper.

II. Related work

Transmission parameter configuration mechanisms such as Adaptive Data Rate (ADR) scheme need to be developed to fit each application requirement in terms of power consumption, delay and packet delivery ratio. Solutions running on LoRa node should be as simple as possible as required in LoRaWAN specification. However, LoRa network server could run complex management mechanism, which can be developed to improve network performance. In this paper, we focus on the server-side mechanisms.

The basic ADR scheme [2] provided by LoRaWAN predicts channel conditions using the maximum received Signal Noise Rate (SNR) in the last 20 packets. The basic ADR scheme is sufficient when the variance of the channel is low, it reduces the interference compared with the static data rate [3][4]. However, their simplicity causes many potential drawbacks. First, the diversity of LoRa Gateway models that measures SNR make the measurement inaccurate as a result of hardware calibration and interfering transmissions. Second, selecting the maximum SNR each 20 packets received could be a very long period in many IoT applications that require less uplink transmission. Third, transmission parameters adjustment considers only the link of a single node. If many LoRa nodes are connected to the near gateway, all nodes connected to this gateway will use the fastest data rate. In this case, the number of LoRa nodes using the same data rate will increase and the probability of collisions also increases dramatically.

The most common metrics used in the literature are *SNR* and Received Signal Strength Indication (*RSSI*) to control *Tx* and *SF*. For example, the authors in [4] slightly modify the basic *ADR* scheme by replacing the maximum *SNR* with the average function. In this paper, we focus on building a framework that help IoT devices to adapt their transmission parameters to the application requirements in a server side.

III. Proposed Framework

The scheme selection process can be described following these five steps:

1) According to the Semtech SX1276 specification[5], there is 6720 possible settings (s_1, ... ,s_{6720}) and the framework has to select the most optimal one or to rank them according to their relevance.
2) The first step of the selection process depends on multiple criteria up to i (c_1, ... , c_i). Different type of criteria can be measured from different sources to cover the maximum point of views, as example, the network server requirements, the applications requirements and the devices conditions.
3) The Fuzzy Logic (FL) based subsystem gives an initial score for each configuration that reflects its relevance. The different sets of scores (d_1, ... ,d_i) are sent to the Multi-Criteria Decision Making (*MCDM*) in the 5^{th} step.
4) At the same time, the genetic algorithm (GA) [6] assigns a suitable weight (w_1, ... ,w_i) for each initial selection decision according to the objective function that is required by the application.
5) Using the initial scores coming from the 3^{rd} step and the weights that are assigned using the 4^{th} step, the multi criteria decision making *MCDM* will select the most relevant settings and rank them according to their reward.

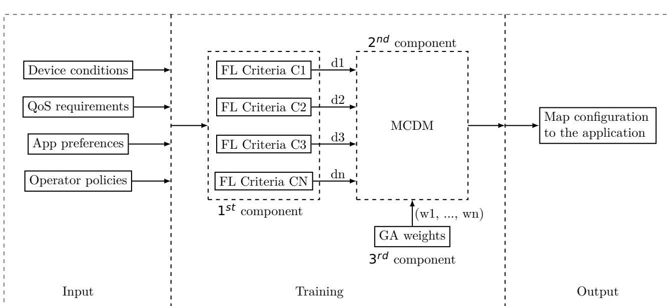
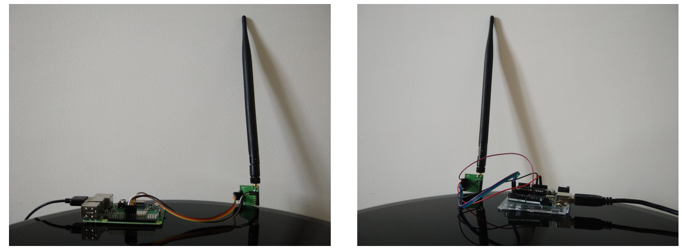


Fig. 1: The proposed scheme for LoRa transmission parameters selection based on *GA*, *FL* and *MCDM*.

IV. Experimentation

For our experimentation we use both real environment (Figure **??**) and ns-3 simulator with SX1276 Lora module. However, to test the scalability of genetic algorithm with numerous IoT devices in a real environment, we use FIT IoT-LAB platform among other platforms presented in [7]. This choice is motivated by the number of devices supported by this platform (up to 2000 nodes).



(a) Gateway (Raspberry-pi). (b) Sensor node (Arduino).

Fig. 2: Gateway & Sensor node.

Figure 2a presents the LoRa gateway that we build using a low cost LoRa gateway [1] on a Raspberry-pi. Figure 2b presents one of the two Arduino boards equipped with an antenna that cover the 868 MHz band and with a SX1276 LoRa Transceiver.

V. Discussion

The main challenge of this work was to explore the application of genetic algorithm in LoRa transmission parameter selection. The efficiency of such algorithms is measured by the ability to satisfy each application requirement. Our main contribution was to build 3 applications that requires 3 different levels of QoS, such as text, sound and image transmission. We used a low cost LoRa gateway on a Raspberry-pi with 2 Arduino boards equipped with 2 LoRa Transceivers based on the Semtech SX1276 specification. To measure the accuracy of applying genetic algorithm in an edge computing we expect to compare our approach with other adaptive data rate solutions.

A. Lora

Preamble		Sync msg	PHY Header	PHDR-CRC	PHY Payload															CRC	
Modulation	length	Sync msg	PHY Header	PHDR-CRC	MAC Header			MAC Payload											MIC	CRC Type	Polynomial
Modulation	length	Sync msg	PHY Header	PHDR-CRC	MType	RFU	Major	Frame Header									FPort	**Frame Payload**	MIC	CRC Type	Polynomial
Modulation	length	Sync msg	PHY Header	PHDR-CRC	MType	RFU	Major	Dev Address		FCtrl					FCnt	FOpts	FPort	**Frame Payload**	MIC	CRC Type	Polynomial
Modulation	length	Sync msg	PHY Header	PHDR-CRC	MType	RFU	Major	NwkID	NwkAddr	ADR	ADRACKReq	ACK	FPending /RFU	FOptsLen	FCnt	FOpts	FPort	**Frame Payload**	MIC	CRC Type	Polynomial
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

0) **Modulation :**
   - Lora: 8 Symbols, 0x34 (Sync Word)
   - FSK: 5 Bytes, 0xC194C1 (Sync Word)
- **Length :**
- **Sync msg :**
- **PHY Header :** It contains:
  - The Payload length (Bytes)
  - **The Code rate**
  - Optional 16bit CRC for payload
- **Phy Header :** CRC It contains CRC of Physical Layer Header
- **MType :** is the message type (uplink or a downlink)
  - whether or not it is a confirmed message (reqst ack)
  - 000 Join Request
  - 001 Join Accept
  - 010 Unconfirmed Data Up
  - 011 Unconfirmed Data Down
  - 100 Confirmed Data Up
  - 101 Confirmed Data Down
  - 110 RFU
  - 111 Proprietary
- **RFU :** Reserved for Future Use
- **Major :** is the LoRaWAN version; currently, only a value of zero is valid
  - 00 LoRaWAN R1
  - 01-11 RFU
- **NwkID :** the short address of the device (Network ID): 31th to 25th
- **NwkAddr :** the short address of the device (Network Address): 24th to 0th
- **ADR :** Network server will change the data rate through appropriate MAC commands
  - 1 To change the data rate
  - 0 No change
- **ADRACKReq :** (Adaptive Data Rate ACK Request): if network doesn't respont in 'ADR-ACK-DELAY' time, end-device switch to next lower data rate.
  - 1 if (ADR-ACK-CNT) >= (ADR-ACK-Limit)
  - 0 otherwise
- **ACK :** (Message Acknowledgement): If end-device is the sender then gateway will send the ACK in next receive window else if gateway is the sender then end-device will send the ACK in next transmission.
  - 1 if confirmed data message
  - 0 otherwise
- **FPending↓ /RFU ↑ :** (Only in downlink), if gateway has more data pending to be send then it asks end-device to open another receive window ASAP
  - 1 to ask for more receive windows
  - 0 otherwise
- **FOptsLen :** is the length of the FOpts field in bytes ă 0000 to 1111
- **FCnt :** 2 type of frame counters
  - FCntUp: counter for uplink data frame, MAX-FCNT-GAP
  - FCntDown: counter for downlink data frame, MAX-FCNY-GAP
- **FOpts :** is used to piggyback MAC commands on a data message
- **FPort :** a multiplexing port field
  - 0 the payload contains only MAC commands
  - 1 to 223 Application Specific
  - 224 & 225 RFU
- **FRMPayload :** (Frame Payload) Encrypted (AES, 128 key length) Data
- **MIC :** is a cryptographic message integrity code
  - computed over the fields MHDR, FHDR, FPort and the encrypted FRMPayload.
- **CRC :** (only in uplink),
  - CCITT $x^{16} + x^{12} + x^5 + 1$
  - IBM $x^{16} + x^{15} + x^5 + 1$

References

[1] *CongducPham/LowCostLoRaGw*. [Online; accessed 13. Aug. 2019]. Aug. 2019.

[2] *LoRaWANő Specification v1.0.3 | LoRa Alliance^{TM}*. [Online; accessed 8. Aug. 2019]. Aug. 2019.

[3] Martin C. Bor et al. " Do LoRa Low-Power Wide-Area Networks Scale? " In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM '16*. The 19th ACM International Conference. 00000. Malta, Malta: ACM Press, 2016, pp. 59–67.

[4] Mariusz Slabicki, Gopika Premsankar, and Mario Di Francesco. " Adaptive Configuration of Lora Networks for Dense IoT Deployments ". In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium* (2018). 00025, pp. 1–9.

[5] *Semtech LoRa Technology Overview | Semtech*. [Online; accessed 12. Sep. 2019]. Sept. 2019.

[6] Mohammed Alkhawlani and Aladdin Ayesh. " Access Network Selection Based on Fuzzy Logic and Genetic Algorithms ". In: *Advances in Artificial Intelligence* 2008 (2008). 00089, pp. 1–12.

[7] Anne-Sophie Tonneau, Nathalie Mitton, and Julien Vandaele. " How to Choose an Experimentation Platform for Wireless Sensor Networks? A Survey on Static and Mobile Wireless Sensor Network Experimentation Facilities ". In: *Ad Hoc Networks* 30 (July 2015). 00046, pp. 115–127.