LoRaWAN Experiments - Adaptive Data Rate in a Gateway-Centric Network context

Édouard Lumet[1] for N7[2]

***Abstract*— LoRaWAN - for Long Range Wide Area Network - is know as a LPWAN (Low Power Wide Area Network) protocol as it is suited for wide area networks of sensors. Those sensors have to consume little energy but they can be few kilometers spaced each other. That is why we must define new types of protocols such as LPWAN. Recently, we have observed an important growth in terms of numbers of sensors and connected devices (smart cities, smart home, smart cars, self-monitoring, air or water quality monitoring, etc.) on the Internet we call now Internet of Things (IoT). Cisco defined IoT as "simply the point in time when more 'things or objects' were connected to the Internet than people".**

Therefore, IoT has a key challenge: dealing with massive networks. LoRaWAN has been thought to address this challenge and brings some solutions such as a modulation allowing several communications at the same time on the same channel or an adaptive data rate algorithm which take density into consideration.

I. INTRODUCTION

By considering the two solutions introduced above, the Adaptive Date Rate (ADR) algorithm is a good starting point if we want to address the density challenge of IoT networks. Indeed, the LoRa modulation is not open and can be considered as a robust and optimised one. ADR has a little specification and everybody is free to adapt it and to implement it in order to fit specific needs.

A. *LoRaWAN networks*

LoRaWAN networks are composed of one or more gateways and several nodes, usually sensors with two AA batteries. The gateway of such a network has different functions: (1) establishing the physical layer, (2) providing network services and configuring it through different parameters then (3) providing application handling. That is why a LoRaWAN network is known as a gateway-centric network.

B. *ADR*

Adaptive Data Rate mechanism is based on few parameters: (1) the *Spreading Factor* (SF), from 7 to 12, (2) the *Signal-Noise Ratio* (SNR), (3) the *Transmission Power* (TP), or (4) the *Received Signal Strength Indicator* (RSSI). The SF allows simultaneous communication over the same channel. The higher the SF, the higher the energy consumption and the lower the data rate but the wider the range. By changing these parameters, either the gateways or the nodes are able to adapt the configuration of a device or the network.

[1]É. Lumet is with ENSEEIHT engineering school, Toulouse, FRANCE `edouard.lumet@etu.enseeiht.fr`

[2]ENSEEIHT, abbreviated N7, is a french engineering school at Toulouse, `http://http://www.enseeiht.fr/en/home.html`

II. RELATED WORK

The key challenge of large scale networks is already being considered, such as in [3], with a machine learning approach, which is similar to our approach as far as we both consider the gateway as a central element. But with machine learning, we do not seek to implement a new adaptive data rate algorithm, we try to guess the best parameters depending on past parameters based on environment.

III. EXPERIMENTATION BASED ON A "HOME MADE" ALGORITHM

ADR algorithm specification as given in the LoRaWAN LoRa Alliance specification has few prerequisites such as: (1) being able to control the ADR bit in LoRaWAN frames, (2) being able to send MAC commands over the network or (3) being able to run code on the nodes. The two first requirements are performed on the gateway, at the network server layer. However, for security reasons, all the network servers do not allow to perform such actions as far as it could let malicious code or person send bad configuration all over the network and block it. Indeed, MAC command is a mechanism introduced in LoRaWAN which allow to send configuration parameters to a node, *see ADR subsection*.

A. *Implementation*

First, we had to *hack* a Network Server to be able to perform some of the actions described above. Our version of Chirpstack (`https://github.com/brocaar/chirpstack-network-server`) is not able to control the ADR bit in order to force the use of ADR over the network or not but it is able to send MAC commands to the nodes. Thus our implementation of ADR does not take ADR bit into consideration and is only based on a number of received frames or packet loss.

Our ADR implementation makes the communication more robust all across the network thank to the way of facing environment degradation. Indeed, when the SNR decreases due to interference for instance, we chose to first increase the transmission power then to increase the spreading factor to avoid packet loss or even communication break. We also chose to run the main ADR algorithm on the gateway only as far as we need to address the massive network key challenge, the gateway has a complete overview of the network.

B. *Results*

To experiment our ADR implementation, we used a node and a gateway. The gateway runs with the network server and the ADR algorithm. All communication were outdoor only but in urban environment, in the centre of Toulouse. We put the node at different places in a kilometer range and we observed the evolution of the parameters given by the ADR updates.

We could see that the algorithm had most of the time an expected behavior. When the Signal-Noise Ratio was to high - the power of the signal is much higher than the power of the noise - ADR output a better SF in terms of data rate but worst in terms of SNR. Hence we could observe a SNR decrease but with a same Packet Delivery Ratio. Indeed, LoRa is built to be able to decode even with a poor SNR. We could also observe the opposite behavior when the node were further.

IV. CONCLUSIONS AND FUTURE WORKS

A gateway-centric approach is not an easy one as it requires to know all the network and to "think" for every node but at a gateway level. However, we cannot run a complete ADR algorithm on a node due to energy consumption consideration. Yet we need to run a light algorithm on the node with the aim of facing packet loss. As a result of our experimentation, we could observe that the node could not rejoin after a communication break with new permanent degraded environment. Thus, we can develop as a future work a packet loss detection mechanism on the node to prevent a communication break by restarting with high parameters (SF12).

Scaling is a common challenge in networks and we could experiment it with our implementation and its evolution as another future work. We may need to add machine learning features which is possible with LoRaWAN as computing can be done in a remote calculator for example. Indeed, when a node send a data to the gateway, in the LoRaWAN A-class principle, the gateway has two one-second slots to reply just after so it could be enough amount of time to perform remote computing.

ACKNOWLEDGMENT

We would like to thank our supervisors, André-Luc Beylot and Rahim Kacimi, for their guidance through the project, along with Mohamed Hamnache, who let us using the nodes and gateways he needs for his PhD. We would give a special thank to Riadh Dhaou for lending us devices. Finally, we would like to thank our English teachers, who prepared us well for this project.

REFERENCES

[1] Dave Evans, *"The Internet of Things: How the Next Evolution of the Internet Is Changing Everything"*, CISCO white paper, April 2011.

[2] N. Sornin (Semtech), M. Luis (Semtech), T. Eirich (IBM), T. Kramp (IBM), O.Hersent (Actility), *"LoRaWAN Specification v1.0.1"*, LoRa Alliance, February 2016.

[3] R. M. Sandoval, A-J. Garcia-Sanchez, J. Garcia-Haro, *"Optimizing and updating LoRa communication parameters: a Machine Learning approach"*, IEEE Transactions on Network and Service Management, July 2019.