Framework for PHY-MAC layers Prototyping in Dense IoT Networks using FIT/CorteXlab Testbed

Othmane Oubejja, Diane Duchemin, Matthieu Imbert, Leonardo Cardoso,
Jean-Marie Gorce

► To cite this version:

Othmane Oubejja, Diane Duchemin, Matthieu Imbert, Leonardo Cardoso, Jean-Marie Gorce. Framework for PHY-MAC layers Prototyping in Dense IoT Networks using FIT/CorteXlab Testbed. CN-ERT 2019 - International Workshop on Computer and Networking Experimental Research using Testbeds, IEEE, Apr 2019, Paris, France. pp.1-2. hal-02150687

HAL Id: hal-02150687

<https://hal.inria.fr/hal-02150687>

Submitted on 7 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Framework for PHY-MAC layers Prototyping in Dense IoT Networks using FIT/CorteXlab Testbed

Othmane Oubejja, Diane Duchemin, Matthieu Imbert, Leonardo S. Cardoso, Jean-Marie Gorce

Univ Lyon, INSA Lyon

Inria, CITI

F-69621 Villeurbanne, France

othmane.oubajja@inria.fr

Abstract—In this paper we present an Internet-of-Things (IoT) network implementation developed as part of the project "Enhanced Physical Layer for Cellular IoT" (EPHYL), using FIT/CorteXlab radio testbed. The aim of our work is to provide a customizable and open source design for IoT networks prototyping in a massive multi-user, synchronized and reproducible environment thanks to the hardware and software capabilities of the testbed. The massive access feature is managed by emulating several sensors per radio nodes. Two categories of network components are used in our design: a base station unit and a multi-sensor emulator unit. The components are separately hosted in dedicated and remotely accessible radio nodes. Their design features can be illustrated through a live demo, which is also reproducible as it is available for any interested reader.

Index Terms—IoT, FIT/CorteXlab, NB-IoT, Reproducibility, LPWAN, Channel Emulation, Software Defined Radio

I. INTRODUCTION

With the growing needs of rigorous and transparent scientific experimentation across many research fields, the importance of reliable testbeds is only getting more crucial in order to validate scientific models and simulations. Among these fields is wireless communication, and more particularly Internet-of-Things (IoT) which is a rapidly growing and challenging domain. However, one of the challenges is robust joint hardware/software prototyping that is still lacking due to various reasons (e.g. synchronization, immutable implementation solutions, unrealistic testing environment) especially for large scale experiments as required for IoT.

FIT/CorteXlab [1] radio testbed has the key properties to develop an IoT network framework, remotely and freely available for the research community. It allows to fasten the execution of large scale emulated radio experiments, as illustrated here with the evaluation of custom PHY-MAC layers derived from 3GPP Narrowband Internet of Things (NB-IoT) standard [2]. The proposed framework offers to researchers the opportunity to plug their algorithms or designs in Software Defined Radio (SDR) nodes, thus making it possible to evaluate diverse communication scenarios and perform necessary physical measurements.

II. FIT/CORTEXLAB RADIO TESTBED

FIT/CorteXlab is a large scale radio testbed composed of an extensive set of SDR nodes. These latter provide substantial RF flexibility, enabling to tune the operating frequency, the

channel bandwidth, the emitted power and the waveform used to communicate. The testbed is located in a large shielded room which is partially covered with electromagnetic absorbing foams. Conclusively, the design of the room enables to ensure a high level of reproducibility of experimentations.

Last but not least, all SDR nodes are remotely accessible through a service interface allowing to deploy proprietary software [1].

III. DESIGN ARCHITECTURE

A. Overview

We started by investigating how we can deploy an IoT-like network considering the testbed capabilities for SDR applications, therefore the objective was to choose then implement key features from NB-IoT systems [2]. Our approach was to design two classes of nodes. The first class represents a Base Station (BS) and the second one a Multi-Sensor emulator (MS). In this work, a Sensor is the actual resource user. To ensure time and frequency synchronization, only USRP nodes are used, since they are connected to 8-Channel Clock Distribution modules. The wireless channels are the downlink Broadcast Channel (BCH) and the uplink, which is based on a custom and simplified version of NB-IoT PHY layer [2] [3]. Fig.1 depicts the overall network structure.

Fig. 1. EPHYL Network diagram.

B. Network Resources Scheduling

The BS follows a defined resource grid policy and a frame structure known by the sensors. At the present time, our design supports single carrier mode only. The BS starts by emitting a beacon signal (BCN) within the Broadcast Channel (BCH) which allows active Sensors to detect the BS and obtain

Fig.2 illustrates a functional example of the network flow during one frame, including one BS node and 3 emulated Sensors running with different parameters.

C. Nodes Description

2) **MS Node:** The MS node is composed of emulated *Sensor* blocks which are connected to a complex adder. After the "wake-up" state, each sensor generates its own packets and waits for the *Sync* block to trigger the local *Scheduler* as soon as a beacon is detected. The *Access Policy* block dictates to the *Scheduler* which shared slots to use based on the BS feedback and modifies PHY parameters if needed. Then, the generated waveform passes through a *Channel Emulator* which applies arbitrary attenuation as well as timing and frequency offsets. (See Fig.4)

The diagram illustrates the system architecture, divided into two main sections: **GNU Radio (Computer)** and **USRP**.

GNU Radio (Computer) Section:

- Sensors:** Four sensors are shown: **Sensor 1**, **Sensor 2**, **Sensor 3**, and **Sensor M**. They are connected via dashed blue lines to a central summation node (a circle with an upward arrow).
- Ethernet Downlink:** A dashed orange line labeled "(From BS)" enters from the left, connecting to a dashed orange box labeled "Ethernet Downlink". This box is connected via a dashed orange line to **Sensor 1**.
- Sensor 1 Internal Structure:**
 - Access Policy:** Receives input from above and sends a signal to **PHY TX (SC-FDM)**.
 - PHY TX (SC-FDM):** Sends a signal to **Local Scheduler**.
 - Local Scheduler:** Sends a signal to **Channel Emulator**.
 - Channel Emulator:** Sends a signal to the summation node.
 - Data bits:** Receives input from above and sends a signal to **PHY TX (SC-FDM)**.
- UHD (Universal Hardware Driver):** A large vertical block that receives signals from the summation node and the **Channel Emulator**. It also receives a **Sync** signal from the USRP section.

USRP Section:

- The **UHD** block outputs a signal to an antenna.
- A **Sync** signal is sent from the antenna back to the **UHD** block.

IV. DEMONSTRATION

V. ACKNOWLEDGMENT

REFERENCES

- [1] L. Cardoso, A. Massouri, B. Guillon, F. Hutu, G. Villemaud, T. Risset, J.M. Gorce, "CortexLab: A Cognitive Radio Testbed for Reproducible Experiments", in Proc. Wireless @ Virginia Tech Symposium, May 2014, Blacksburg, United States. 2014.
- [2] Y. -. E. Wang et al., "A Primer on 3GPP Narrowband Internet of Things," in IEEE Communications Magazine, vol. 55, no. 3, pp. 117-123, March 2017.
- [3] X. Lin, A. Adhikary and Y. -. Eric Wang, "Random Access Preamble Design and Detection for 3GPP Narrowband IoT Systems," in IEEE Wireless Communications Letters, vol. 5, no. 6, pp. 640-643, Dec. 2016.
- [4] "FIT/CortexLab Wiki". [Online]. Available: <https://wiki.cortexlab.fr/>. [Accessed: 17- Dec- 2018].