GNU Radio implementation of a MAC level protocol for an avionic demonstrator

Guy Durrieu GUY.DURRIEU@ONERA.FR
ONERA/DTIS, Centre de Toulouse, 2, avenue Édouard Belin BP 74025 31055 TOULOUSE CEDEX 4, France

Abstract

The present paper describes some aspects of a project conducted at [our lab] the goal of which is to investigate the feasibility of wireless communication within avionic embedded critical system architectures, the expected benefits being, among others, a simplified design of the whole aircraft, an easier maintenance and a reduced weight of the aircraft.

During the project, which involved people from physics (electromagnetism, optics) as well as people from computer science, a demonstrator was developed, including a communication protocol, the physical layer and logical layer of which were implemented with GNU Radio.

1. Introduction

Wireless technologies are increasingly used in various domains, in particular in the aeronautical domain (aircraft/environment communication, passenger services and entertainment) but encounter severe difficulties in the domain of critical avionic embedded systems, main problems being environmental disturbance immunity, required temporal guarantees and security risks.

However, an increased use of wireless technologies in aeronautics would have significant advantages: modern aircraft include hundreds of kilometers of cable, resulting in extra weight of several tons and extra cost of millions of dollars.

ONERA (*Office National d'Études et de Recherches Aérospatiales*), French Aerospace Lab, conducted a project aimed at feasibility of wireless communication within avionic embedded critical system architectures, firstly focused on redundant or backup links, the expected benefits being a simplified design of the whole aircraft, gains in maneuverability, reduced fuel consumption, increased payload and autonomy, easier maintenance, etc.

This project involved people from physics (electromagnetism, optics) as well as people from computer science, in order to investigate different wireless communication techniques according to each specific case (e.g. optical communication between engines and flight computers, wireless communication within the cabin between sensors and flight computers, etc.). Models were developed (mainly for optical communications) as well as a demonstrator dedicated to wireless communication, briefly presented in the next section.

In this paper we obviously only consider communication protocols on wireless electromagnetic channels. GNU Radio was used to implement a physical communication layer between elements of the demonstrator. However the specific domain investigated (avionic embedded critical systems) imply to put in the loop the logical communication layer, since issues such as security, reliability or guaranteed bounded latencies cannot be analyzed, solved or mitigated only at the physical level. This logical layer must be tightly coupled to the physical one, and thus rather developed using GNU Radio as well. It also will be able to exchange data with the above application layer.

The main goal of this paper is to show that such an implementation can be achieved, as well as the problems encountered. The project is still in progress and the implemented MAC protocol quite simple; it will be enhanced as the experiment goes on and solutions

Proceedings of the 10^{th} GNU Radio Conference, Copyright 2020 by the author(s).

are successfully tested.

Section 2 summarizes the context of the study (the principles of the avionic demonstrator) and section 3 describes the proposed protocol and its GNU Radio implementation. Section 4 gives some hints about the future of the project.

2. Context

The Department of ONERA in charge of Electromagnetism and Radar studies developed an approach of channel propagation evaluation for wireless communication on aircraft based on a *Mode Stirrer Reverberating Chamber* (MSRC); This apparatus is described in (Quenum et al., 2016), (Quenum et al., 2017), (Quenum & Junqua, 2018), (Quenum & L'Hour, 2019).

Furthermore experiments were performed in a Falcon 20 aircraft in order to acquire representative and typical complex propagation path features in a typical aircraft. These experiments consisted in measuring the broadband transfer function, in magnitude and phase, between transmitting and a receiving antennas. Antennas were placed in various main bays of the aircraft, at the front (cockpit), middle (cabin) and rear of the aircraft. Additionally, a mechanical stirrer was introduced in aircraft compartments, inducing fluctuations of the propagation channel (DEMR, 2018).

The goal is then to simulate a similar aircraft confined environment. A metallic parallelepiped structure, made of three connected elementary cavities, possibly representing the front, middle and rear parts of the aircraft (figure 1).

The main advantage of the MSRC is the ability to generate a reproducible and controllable statistical environment (Holloway et al., 2006); it can be configured to obtain results similar to those observed in the real aircraft. An example is given at figure 2, which shows the result obtained in the MSRC compared to the measures done in the aircraft.

3. GNU Radio and MAC protocol

3.1. Motivation

The demonstrator briefly described at section 2 and with more details in (Quenum et al., 2017) and

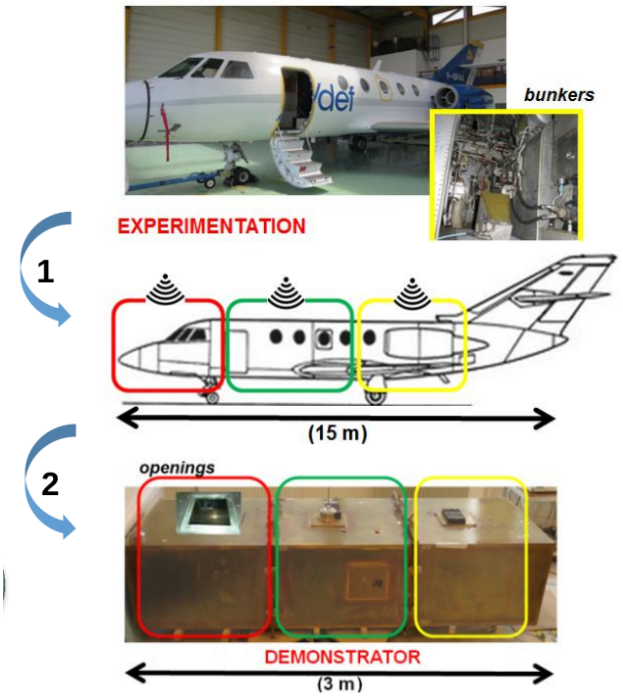

Figure 1. The demonstrator

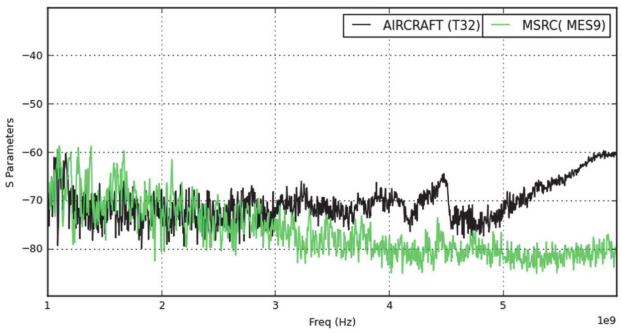

Figure 2. Aircraft cabin-rear vs. loaded MSCR structure path loss

(Quenum & Junqua, 2018) was first used for simulating and approximating a realistic avionic wireless channel.

When planning to use it in order to assess the reliability and security of a wireless communication, it becomes necessary to include several levels of protocols between the source and receiving antennas.

A physical layer was developed with GNU Radio by the Electomagnetism ad Radars Department team, mainly based on a burst chain and a QPSK modulation.

The Computer Science Department team was in charge of the design of a MAC (*Medium Access Control*) protocol logical layer, to be fully integrated in the demonstrator, therefore also implemented as a GNU Radio component. This section is devoted to the description of this protocol and its implementation.

3.2. Requirements for critical embedded systems

The key points restraining the use of wireless communication in critical embedded systems are the following:

- security of communications
- reliability of communications
- ability to guarantee a worst case bound to the end to end data transmission delay.

Communication security is especially crucial for wireless networks, compared to wired ones, as they are more vulnerable to malicious attacks.

Communication reliability mainly depends on the transmission channel characteristics. It theoretically can be analyzed through classic techniques (FHA: *Functional Hazard Assessment* and SSA: *System Safety Assessment*) and made compatible with the specified requirements through appropriate architectural redundancies (duplicated networks).

Concerning transmission latencies, trouble comes from access conflicts to the shared channel; additionally, for wireless networks, other specific troublemaking situations can occur, such so called *hidden terminal* or *exposed terminal* phenomena. However, bounded end to end latencies are required for networks used in critical avionic systems, in order to ensure some level of determinism for the system behavior.

None of these points can be guaranteed by the sole physical communication layer. Adequate features have to be implemented in the above logical (MAC) layer in order to obtain the desired level of guarantee about security, reliability and determinism of wireless communications:

- For the specific domain of avionic critical embedded systems, security can be at least partially taken into account through an exhaustive knowledge by the MAC protocol of the authorized sources on the network.
- Communication reliability for a single network can be enhanced by addition of CRC (Cyclic Redundancy Checks) and by exchange of MAC control frames between sources and receivers.
- From the performance point of view, the challenge, for a MAC layer protocol in the context of critical embedded systems, is to reduce or even eliminate occurrences of medium access contentions, in order to guarantee bounded end to end transmission latencies.

In the specific case of the avionic demonstrator presented at section 2, an additional requirement comes from the choice of GNU Radio as implementation tool for the physical layer of the protocol stack. This choice strongly suggests that GNU Radio should also be used to implement the logical layer: from performance point of view, the standard message passing of GNU Radio would allow a tighter interface between the physical layer and the logical layer than stream sockets used in (Gutierrez-Agullo et al., 2010). On the other hand, data exchanges with the application layer would be supported by sockets.

Section 3.5 shows how this implementation of a logical layer with GNU Radio can be achieved.

3.3. MAC wireless protocols

Lots of MAC level wireless protocol have been proposed, the goal of which being to eliminate or to prevent as far as possible interference between sources on a network. They can be classified into two main groups:

- distributed protocols
- centralized protocols

3.3.1. Distributed protocols

CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) is the basic distributed protocol for wireless networks. It is derived from the the

CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) used for wired networks such as Ethernet, but instead of detecting the occurrence of conflicts, it tries to avoid them. A source looks at the channel before transmitting anything. If the channel is busy, it waits during a random delay. Otherwise it initiates a negotiation process during which the other sources keep waiting, followed by the actual data transmission. Since conflicts cannot be entirely avoided, due in particular to hidden terminals, an acknowledge from the destination in necessary.

CSMA/CA is used in particular by the IEEE 802.11 Distributed Coordination Function (DCF) of the WiFi standard, and in the Contention Access Period (CAP) defined by the IEEE 802.15.4 (*ZigBee*) standard.

Other protocols may use additional features in order to improve contention avoidance:

- time slot reservation mechanisms (Tang & Garcia-Luna-Aceves, 1999), (Lin & Gerla, 1997),
- scheduling mechanism (Kanodia et al., 2002),
- specific control channel to help solving the hidden terminal problem (Tobagi & Kleinrock, 1975),(Tobagi, 1987).

3.3.2. Centralized protocols

Centralized protocol specialize a component as a *coordinator* defining a temporal slicing.

The coordinator periodically sends a "beacon" defining a new communication period. Inside this period, two sub-periods are defined:

- a contention free sub-period: CFP (Contention Free Period) for ZigBee, PCF (Point Coordination Function) for IEEE 802.11.
- a contention period, within which sources can send data on a random basis CAP (Contention Access Period) for ZigBee, DCF (Distributed Coordination Function) for IEEE 802.11 (see 3.3.1).

Both sub-periods are again divided into time slots, within wich communications occur.

3.4. Proposed MAC protocol

Given what was said in the previous sections, the chosen protocol is a centralized protocol similar to those of the section 3.3.2, since this kind of protocol is better able to meet the requirements listed at section 3.2, particularly the need of bounded transmission latencies. However it is kept simpler than the protocols mentioned at section 3.3.2, for two reasons:

- it is an experimental tool aimed at testing different mechanisms able to provide answers to the requirements listed at section 3.2; these mechanisms will be added to (or removed from) the protocol logical level as the experiment progresses.
- a certification process preparing the integration of a wireless communication protocol in an avionic critical embedded system would prohibit unnecessary features in the protocol.

Furthermore, the chosen protocol takes into account the geometrical specific features of an aircraft, which can make difficult a direct communication between a concentrator and the elementary sources.

As several other centralized protocols, the chosen protocol mainly addresses networks of sensors.

A coordinator defines a period, but instead of sending a global beacon in relation to which a timing is specified, the coordinator periodically sends an individual permission to emit (beacon) to each known source (figure 3), and waits for the answer before proceeding. A watchdog armed each time a beacon is emitted prevents from delaying the interrogation of next source when a given source doesn't respond within a given time interval.

The permission, or beacon, for a given source as well as the corresponding response may pass through a gateway when the source is not directly reachable by the coordinator, due to a geometrical issue. the MAC protocol specification includes this gateway function, which may be carried out either by a specific device or by a source.

This experimental protocol *a priori* meets the requirement related to the ability of binding the transmission delay of a message, since at most one source is able

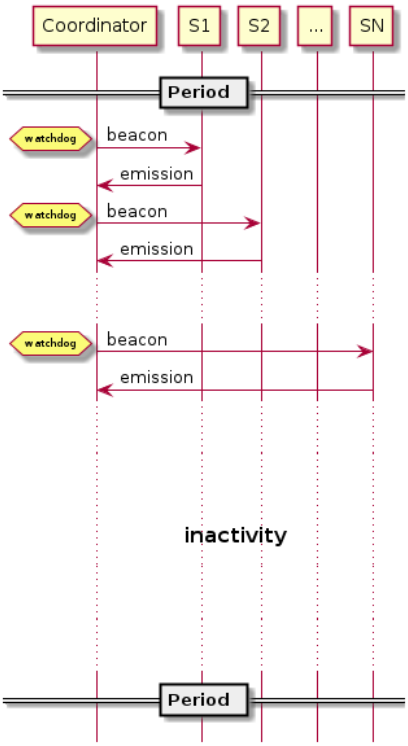

Figure 3. Schematic operation of the protocol

to emit on the channel at a given time; this is to be experimentally verified on the avionic demonstrator. Furthermore, in order to allow a comparison of the performance of this centralized protocol with a distributed one, a basic CSMA/CA protocol is currently integrated, the choice between the two being fixed by a parameter when initializing the experimental system.

3.5. Implementation

3.5.1. Frame

La figure 4 gives the format of a frame routed through the network.

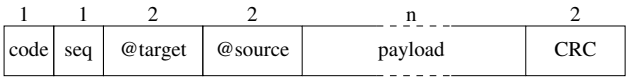

Figure 4. Frame format

The code field differentiates the messages (beacon, data,...). The sequence field is related to a given source, and incremented each time the source emits a new message; it allows the coordinator to verify that id did not missed a message and that a message has not been received several times, due to transient conditions.

In addition to the variable size payload, the frame also includes the target and source addresses, and a Cyclic Redundancy Check field allowing to detect a transmission incident.

3.5.2. GNU Radio component

The proposed MAC protocol is implemented as a standard GNU Radio module (figure 4). This component is intended to be used either in an operational context within the demonstrator or in simulations for experimenting network topologies which cannot be easily set up in the demonstrator.

When starting this work we had a knowledge of previous attempts (Bloessl, 2012), (Gutierrez-Agullo et al., 2010), (Gutierrez-Agullo et al., 2013).

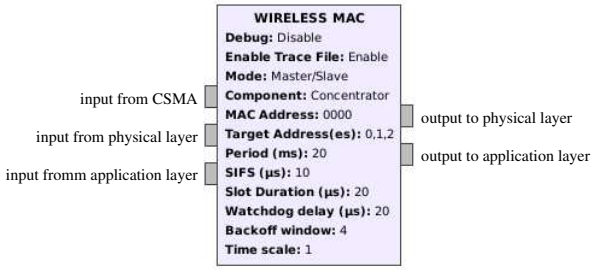

Figure 5. GNU Radio component implementing the MAC protocol

Ports

The CSMA input port allows the MAC component to be connected to another component detecting activity on the channel, still under test.

Application ports allow the MAC component to exchange information with the upper layer, through sockets.

Physical ports allow the MAC component to send/receive messages to/from the physical layer, also implemented using GNU Radio.

Parameters

The MAC component is parameterized as follows:

Debug This parameter allows to obtain additional data during debug phases.

Enable Trace File Each MAC component optionally may produce a log file, the name of which is related to the instance or the component, containing essential data about communication done during the session.

Mode This parameter defines the operation mode of the component:

- Master/Slave (centralized mode),
- CSMA/CA.

Component This parameter defines the role of the module, either:

- Concentrator (coordinator), or
- Device (source and/or gateway).

MAC Address This parameter defines the MAC address of the component.

Target Adresses(es) This parameter defines a set of addresses:

- for the concentrator these addresses are the addresses of the known components; when operating in centralized mode, these components have to be periodically polled.
- for a device, the first address is the concentrator one; if there are others, the gateway function is active, and the additional addresses are the addresses of components the messages of which have to be resent.

Period This parameter defines:

- for the concentrator in centralized mode the duration in milliseconds of the polling period.
- for a basic device in distributed mode the duration in milliseconds of the emission period.

SIFS This parameter defines the duration in microseconds of the *Short Inter Frame Space* of the CSMA/CA protocol.

Watchdog delay This parameter defines the duration in microseconds of the maximal time delay during which the concentrator waits for a response of a device.

Backoff window This parameter defines the size in microseconds of the window from which, in CSMA/CA mode, a random delay is taken when an activity is detected on the channel by a component while attempting an emission.

Time scale This parameter is discussed in the following section

Simulation vs. integration

An attractive feature of GNU Radio is the ability to simulate a component before integrating it into a "real" system. This is especially interesting for a logical component, the functional behavior of which can be independently verified.

As said previously, the MAC component needs a variety of real-time delays obtained from the operating system. Its right behavior depends on an adequate value for these real-time delays.

However, the physical components potentially involved in the simulation are considerably slowed down by the simulation process, due to the *Channel Model* and *Throttle* components introduced for the simulation. It is thus necessary to scale up as well the different real-time parameters defined for the MAC component. The *Time scale* parameter defines the multiplicative factor to apply to each of the time parameters in order to cope with the global rate of the simulation.

As an example, the simulation displayed at figure 6, essentially including a concentrator and a device communicating through the physical layer (*Psk Burst Tx* and *Psk Burst Rx*) requires a time factor of 300 for each MAC component in order to work properly. If other components are added to the simulation, this value will be likely to increase.

The scale factor is normally 1 when the MAC component is integrated into the "real" system.

Figure 7 shows an extract of the trace file produced by the concentrator in the simulation of figure 6 when the *Enable trace file* parameter is on (which is not the case

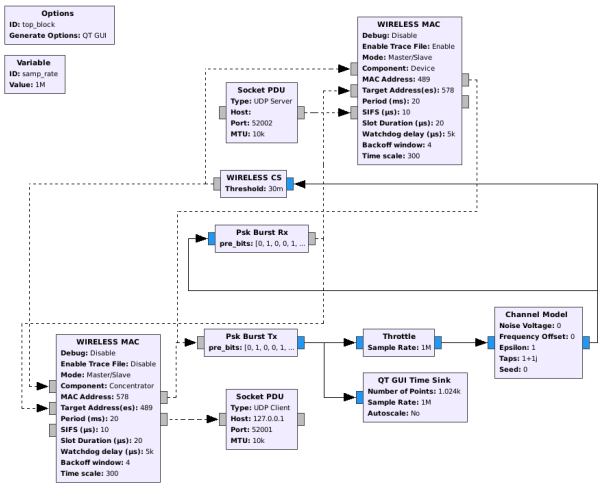

Figure 6. Simple simulation of a source and target units using the MAC protocol

```
578 OK MASTER/SLAVE 578 and CONCENTRATOR
578 : NEW PERIOD AFTER 0 milliseconds
578 : REQUEST SENT TO 489 SEQ = 0 AFTER 0 milliseconds
578 : TIMEOUT AFTER 1500 milliseconds
578 : NEW PERIOD AFTER 6000 milliseconds
578 : REQUEST SENT TO 489 SEQ = 1 AFTER 6000 milliseconds
578 : TIMEOUT AFTER 7500 milliseconds
578 : NEW PERIOD AFTER 12000 milliseconds
578 : REQUEST SENT TO 489 SEQ = 2 AFTER 12000 milliseconds
578 : TIMEOUT AFTER 13500 milliseconds
578 : NEW PERIOD AFTER 18000 milliseconds
578 : REQUEST SENT TO 489 SEQ = 3 AFTER 18000 milliseconds
578 : MESSAGE RECEIVED FROM 489 AFTER 19488 milliseconds
578 : NO TIMEOUT AFTER 19488 milliseconds
578 : NEW PERIOD AFTER 24000 milliseconds
578 : REQUEST SENT TO 489 SEQ = 4 AFTER 24006 milliseconds
578 : MESSAGE RECEIVED FROM 489 AFTER 25429 milliseconds
578 : NO TIMEOUT AFTER 25429 milliseconds
578 : NEW PERIOD AFTER 30000 milliseconds
```

Figure 7. Extract of the trace produced by the concentrator in the simulation of figure 6

in figure 6). A new line is produced on occurrence of an event related to the communication process.The first line summarizes the configuration of the component.

The different time delays appearing in the trace file, counted from the beginning of the simulation/session, take into account Te scale factor. For example, the specified period of 20 milliseconds results in a period of 3000 milliseconds during the simulation . Similarly, the specified watchdog delay of 5000 microseconds results in a timeout delay of 1500 milliseconds during the simulation. Timeouts occur at the beginning of the simulation since the targeted component starts sending data in response to the beacon after an initial delay.

The value of the scale factor is currently empirical, since it is difficult to automatically evaluate it. A similar solution was used in (Gutierrez-Agullo et al., 2010), (Gutierrez-Agullo et al., 2013). We briefly revisit this point in the section 4.

C++ implementation

The MAC component implementation makes use of the C++ BOOST library (BOOST), unique tool allowing to handle, in C++ code under GNU Radio, real-time objects and functions such as *delays*, *timers*, *semaphores*, *synchronizations*, *wait* and *threads*, necessary for implementing the protocol described at section 3.4. In particular involves several *threads*. Usage of threads is made necessary given the waiting periods implied by the protocol, which should not block the others functions of the protocol. A consequence of the usage or threads is the need of mutual exclusion semaphores, especially at the channel access level, for example when the periodic emission function and the gateway function of a basic device have to transmit something at the same time.

Concentrator: The polling function of the concentrator in centralized mode is implemented as a thread. This thread enters waiting periods:

- after polling a given device, for a time delay limited by the watchdog delay parameter,
- when all devices have been polled, until the next polling period.

The polling thread leaves its waiting state:

- on arrival of a device response; the callback associated to the input port from physical layer then emits a notification which reactivates the polling thread,
- on watchdog delay expiration,
- at the beginning of a new polling period.

Basic device: The protocol for a basic device implies two FIFOs:

- a FIFO (1) receiving the incoming data from application layer,
- a FIFO (2) receiving the incoming data from the physical layer which needs to be forwarded to the concentrator (gateway function).

The basic function executed by a device in distributed mode is implemented as a thread. this thread is periodically activated and then sends to the concentrator the first data present in the FIFO 1, if any, according to the CSMA-CA policy.

The gateway function of a device is also implemented as a thread. It is awakened as soon as there is a data in the FIFO 2. And as long as there still are data in the FIFO 2, it send them to the concentrator according to the CSMA-CA policy.

As previously said, a mutual exclusion semaphore guarantees that the two above threads cannot access the medium at the same time.

4. Conclusion and perspectives

In this paper we presented an experimental apparatus aimed at investigating the feasibility of wireless communication within avionic embedded critical system. this demonstrator reproduces a typical aircraft wireless channel. We enumerated some of the requirements which must be met by a wireless communication in the context of avionic critical systems and pointed out that assessing the extend to which these requirements can be satisfied requires a wireless protocol including at least a physical layer and a logical (MAC) layer.

GNU radio was chosen for implementing the physical layer and we shown that this choice strongly suggest to implement the logical layer with GNU Radio as well, We presented the first elementary version of the MAC protocol defined and a method for implementing it with GNU Radio, taking into account both simulation and integration aspects.

The simulations and the first experiments carried out shown that the logical and physical layers work as expected. The demonstrator enters now an exploitation phase; depending on results, additional features could be integrated to the protocol.

Concerning the use of this GNU Radio Component within simulations, it would be interesting to calculate an upper bound of the Time Scale parameter instead or giving it an empirical value. This is possibly tricky, since this factor depends on multiple features, among others: the processor computing power, the number of modules involved in the simulation and the sample rate of the Throttle component. However this would be useful for simulating systems involving both physical and logical functions.

The problem would be ideally solved if GNU Radio had available a "true" simulation tool such those found in the Ptolemy II modeling and simulation environment (PtolemyII), which allow among others both physical and logical simulations. In this regard it would be interesting to explore the possibility to establish a bridge between GNU Radio and the Ptolemy II environment, as it was done for other tools (Matlab/Simulink for example (Chenguang et al., 2014)). By the way some work has been done in this direction several years ago (Dreier, 2006), which deserves to be further explored.

References

Bloessl, B. IEEE802.15.4 O-QPSK for GNU Radio. https://github.com/bastibl/gr-ieee802-15-4, 2012.

BOOST. Boost C++ libraries. https://www.boost.org/, 2020.

Chenguang, Z., Hongman, Y., Dong, L., Hong, Z., Yun, W., and Yunhui, C. Co-simulation research and application for Active Distribution Network based on Ptolemy II and Simulink. In *2014 China International Conference on Electricity Distribution (CICED)*, pp. 1230–1235. IEEE, September 2014.

DEMR. Contribution techniques réalisées dans le PRF WIRELESS: Caractérisation du canal de propagation RF dans une structure de type avion. Rapport technique RT 2/24215 DEMR, ONERA, Novembre 2018.

Dreier, T. J. *Design Environment for Rapid Prototyping of Software Defined Radio*. Thesis submitted in partial satisfaction of the requirements for the

degree master of science in electrical engineering, University of California Los Angeles, 2006.

Gutierrez-Agullo, J.R., Coll-Perales, B., and Gozalvez, J. An IEEE 802.11 MAC Software Defined Radio Implementation for Experimental Wireless Communications and Networking Research. In *Proceedings of the 2010 IFIP/IEEE Wireless Days (WD'10)*, Venice (Italy), October 2010. IEEE.

Gutierrez-Agullo, J.R., Coll-Perales, B., and Gozalvez, J. Uwicore MAC Project. https://github.com/syifan/gr-ieee802-11/tree/master/uwicore_80211_MAC, 2013.

Holloway, C. L., Hill, D. A., Ladbury, J. M., Wilson, P. F., Koepke, G., and Coder, J. On the Use of Reverberation Chambers to Simulate a Rician Radio Environment for the Testing of Wireless Devices. *IEEE Transactions on Antennas and Propagation*, 54(11):3167–3177, November 2006.

Kanodia, V., Li, C., Sabharwal, A., Sadeghi, B., and Knightly, E. Ordered Packet Scheduling in Wireless Ad Hoc Networks: Mechanisms and Performance Analysis. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing MobiHoc'02*, pp. 50–70, Lausanne, Switzerland, June 2002. ACM.

Lin, C. R. and Gerla, M. Adaptive Clustering for Mobile Wireless Networks. *IEEE Journal on Selected Areas in Communication*, 15(7):1265–1275, September 1997.

PtolemyII. Ptolemy Project. https://ptolemy.berkeley.edu/, 2020.

Quenum, W. and Junqua, I. Use of Mode Stirred Reverberating Chambers for evaluating wireless communication performances. In *2018 IEEE Asia-Pacific Microwave Conference*, pp. 396–398. IEEE, November 2018.

Quenum, W. and L'Hour, C.-A. Wireless Systems Assessment in Confined Metallic Environment. In *2019 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)*, New Orleans, USA, July 2019. IEEE.

Quenum, W., Junqua, I., and Parmantier, J.-P. Experimental Power Transfer and Signal Assessment of Wireless Communications in Reverberating Cavities. In *2016 IEEE Wireless Power Transfer Conference (WPTC)*, pp. 1–4, Aveiro, Portugal, May 2016. IEEE.

Quenum, W., Jeannin, N., and Junqua, I. Channel Propagation Emulation in Mode Stirred Reverberating Chambers. In *2017 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)*, pp. 135–139, Washington, DC, USA, August 2017. IEEE.

Tang, Z. and Garcia-Luna-Aceves, J. J. Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks. In *Proceedings of IEEE INFOCOM'99, Conference on Computer Communications*, volume 1, pp. 194–201, New York, NY, USA, March 1999. IEEE.

Tobagi, F. A. Modeling and Performance Analysis of Multihop Packet Radio Networks. *Proceedings of the IEEE*, 75(1):135–155, January 1987.

Tobagi, F. A. and Kleinrock, L. Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *IEEE Transactions on Communications*, COM-23(12):1417–1433, December 1975.